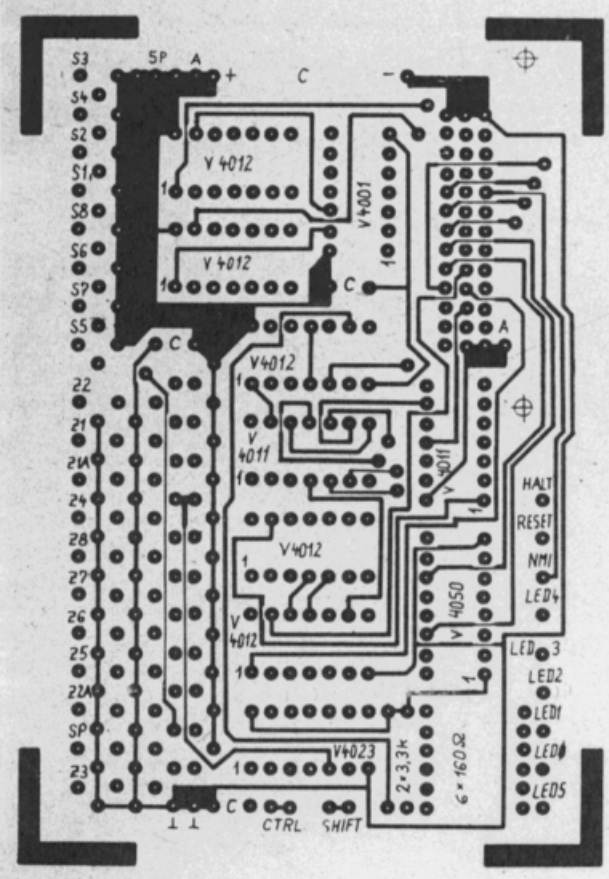

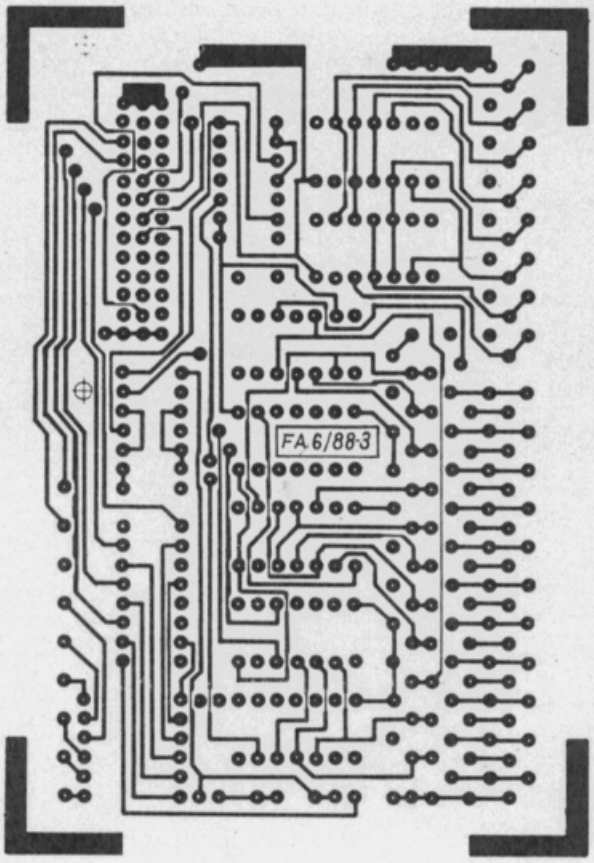

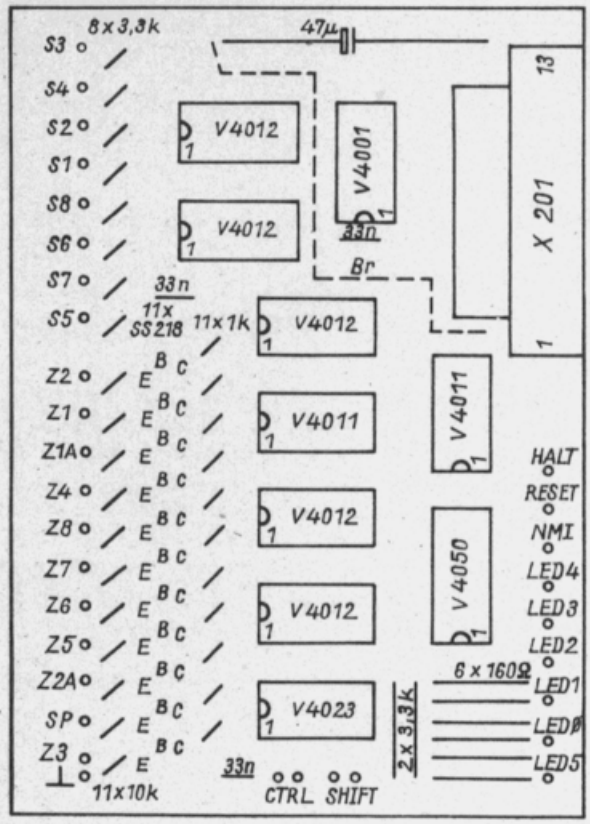

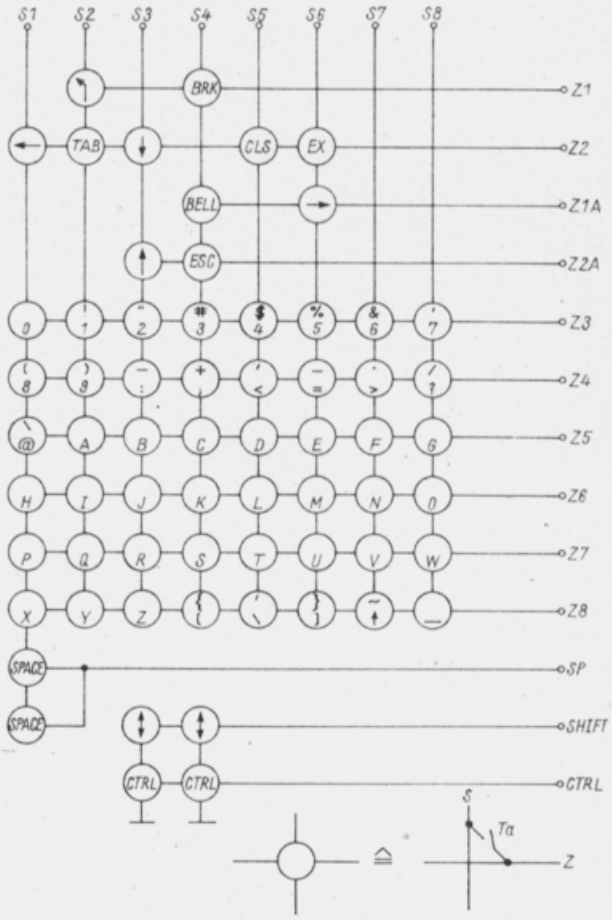

Bild 16a: Layout der Bestückungsseite der Platine für die Tastaturelektronik

Bild 16b: Layout der Leiterseite der Platine für die Tastaturelektronik

Bild 16c: Bestückungsplan der Platine für die Tastaturelektronik

Bild 17: Darstellung der in der Matrix angeordneten 65 Tasten

enthalten die erforderlichen Informationen. Die Ein- und Austrittspunkte der Module sind in Listen (Sprungvektoren) auf den ersten Adressen des jeweiligen Modules zusammengefaßt. Eine Erweiterung ist möglich, wenn man weitere Sprungvektoren anfügt. Die Reihenfolge der Sprünge und deren Bedeutung sollte aber unter allen Umständen erhalten bleiben, um die Kompatibilität zu anderen Computern des gleichen Typs zu erhalten.

Die Bestandteile des PC/M-Grundbetriebssystems sind:

- das BIOS (BASIC Input Output System);
- der Debugger (Eingeben, Ändern und Testen von Programmen und Dateien);
- das Kassettenmodul V-Tape (logischer Kern des Grundbetriebssystems und Kassettenarbeit).

Werden die Programmteile

- CCP (Control Command Processor) und
- BDOS (Basic Disc Operating System)

nachgeladen, erhält man ein CP/M-kompatibles Betriebssystem, für das es bereits eine Vielzahl von Programmen gibt.

5.2. Der Software-Kern

Kern des gesamten Betriebssystemes ist eine Sammlung von Programmen, die die Hardware des Rechners unmittelbar steuern (physische Treiber). Diese Routinen tragen den Charakter von Standardschnittstellen, die dem CP/M- bzw. SCP-BIOS entsprechen [5]. Viele Programme von Heimcomputern oder anderen Systemen verwenden diese oder sehr ähnliche Schnittstellen, wodurch sich der Aufwand bei der Programmanpassung sehr stark reduzieren läßt. Voraussetzung ist natürlich eine ähnliche Hardware und der gleiche Mikroprozessortyp.

Das BIOS beginnt auf der Adresse ØDEØØH des Computers. In lückenloser Folge stehen ab dieser Adresse Sprünge, die zu den einzelnen Programmteilen des BIOS verzweigen. Das BIOS ist in zwei Versionen vorgestellt; je nach Ausbaustufe des RAM-Bereiches. Wird nur eine RAM-Bank verwendet (64 KByte), so sind verschiedene Routinen im BIOS nicht verwendbar und das System ist nicht mehr als CP/M-kompatibler Rechner einsetzbar. Trotzdem ist eine Fülle von Programmen nutzbar. Ist der Rechner mit insgesamt 128-KByte-RAM ausgerüstet, kann bereits mit einem RAM-Floppy von 62 KByte gearbeitet werden. Dazu existiert ein BIOS, das zwei RAM-Floppy's zu je 62 KByte verwalten kann (Bild 5). Bei 192 KByte RAM ist dann ein BIOS mit einem RAM-Floppy zu 124-KByte-RAM einzusetzen, da viele Programme einen hohen Speicherbedarf haben und nicht auf mehrere Floppys (RAM-Bereich) verteilt werden dürfen (z. B. WordStar). Das BIOS-Listing ist in Bild 36 dargestellt. Die BIOS-Routinen sind im einzelnen:

- **BOOT (Kaltstart; ØDEØØH)**

Der Start des Betriebssystems erfolgt nach RESET oder nach dem Einschalten. Nach Vorbereitung der Hardware für den Betrieb und verschiedenen Grundeinstellungen wird das Betriebssystem vom ROM in den RAM geladen und der ROM-Bereich danach abgeschaltet. Dadurch ist der gesamte Arbeitsspeicher des Mikroprozessors als RAM verfügbar und beliebig beschreibbar, was wiederum die Anpassung fremder Software erleichtert.

Nach Löschen des Bildschirms erfolgt eine Kaltstartausschrift mit Anzeige der Systemversion. Die Systemports (Kassettengerät und SIO-Schnittstelle) werden initialisiert. Die Steuerwörter für die Initialisierung habe ich in einer Tabelle zusammengefaßt. Sie sind erweiterbar bzw. ergänzbar (Adresse ØE3B2H bis ØE3F1H, bei 124-KByte-RAM-Floppy). In dieser Tabelle ist zuerst der Port eingetragen, auf den das nachfolgende Steuerwort auszugeben ist. Schließlich erfolgt ein Sprung in das Kassettenmodul V-Tape, welches zusätzlich die Steuerung des Systems übernimmt.

- **WBOOT (Warmstart; ØDEØ3H)**

Die Routine versucht, in das RAM-Floppy-Betriebssystem (Bestandteil des CP/V-Systems) einzusteigen bzw. wieder dorthin zurückzukehren. Gleichzeitig lädt das Programm den Debugger und das V-Tape-Modul vom ROM nach. Nun wird geprüft, ob CCP und BDOS bereits geladen sind. Ist das der Fall, erscheint die Angabe des aktuellen Laufwerkes, und die Arbeit im CP/M-kompatiblen CP/V-System kann beginnen. Verschiedene CP/M-Programme zerstören während ihrer Arbeit den CCP. Um ein häufiges Nachladen zu vermeiden, ist der CCP im ROM abgelegt. Das BDOS ist grundsätzlich von Kassette nachzuladen, es wird aber im Normalfall während der Programmarbeit nicht zerstört. Stellt das Programm einen Fehler im BDOS bzw. CCP fest, erscheint die Ausschrift "CCP/BDOS Laden!" (CCP-ROM ok). Dann muß man das entsprechende Modul von Kassette nachladen. Die ROM-Belegung ist:

IS D14-BIOS,
IS D15-Debugger,
IS D16-V-Tape,
IS D17-CCP.

Der Rechner ist bereits mit einer Bestükkung von D14 und D15 arbeitsfähig.

- **CSTS (Control STatuS, Tastaturstatus; ØDEØ6H):**

Die Routine kehrt mit einer Information über den Tastaturstatus in das aufrufende Programm zurück. Wurde eine Taste gedrückt, enthält das Register A der ZVE den Wert 0FFH, ist keine Taste gedrückt, enthält es 00H.

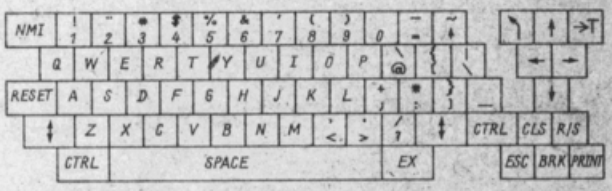

Bild 18: Ansicht der realisierten Tastenanordnung

- **CI (Consol Input, Tastaturabfrage; ØDEØ9H):**

Diese Routine kehrt mit dem Kode der gedrückten Taste ins Register A der ZVE zurück. Durch den Tastenkode 01EH wird der Tastenklick (kurzer Signalton) ein- bzw. ausgeschaltet. Ist der Signalton ausgeschaltet, zeigt LED 2 dies an. Der Kode 013H wird für die Run/Stop (Lauf/Halt)-Funktion verwendet und durch LED 1 angezeigt. Der Tastenkode 01FH ist zur Rückkehr aus dem RAM-Floppy-System in das CP/V-Grundbetriebssystem nutzbar.

- **CO (Consol Output, Zeichenausgabe; ØDEØCH):**

Ein im Register C der ZVE abgelegtes ASCII-Zeichen wird auf der Konsole ausgegeben. Das Beschreiben des Bildschirms erfolgt zeilenweise, links oben beginnend. Ist die Kursordarstellung zugelassen, erscheint die nächste zu beschreibende Bildschirmposition mit der inversen Zeichendarstellung (Bildpunkte hell statt dunkel und umgekehrt).

(wird fortgesetzt)

Literatur

[1] Kieser, H.; Meder, M.: Mikroprozessortechnik, 4. Auflage, VEB Verlag Technik, Berlin 1986
[2] Kühn, E.: Handbuch TTL- und CMOS-Schaltkreise, 2. Auflage, VEB Verlag Technik, Berlin 1986
[3] Classen, L.; Oefler, U.: Wissensspeicher Mikrorechnerprogrammierung, 1. Auflage, VEB Verlag Technik, Berlin 1986