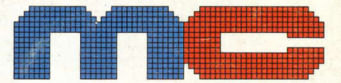


Sonderheft Nr. 80
Preis 17 DM,
129 öS, 17 sfr.



DAS EMUF- SONDERHEFT.

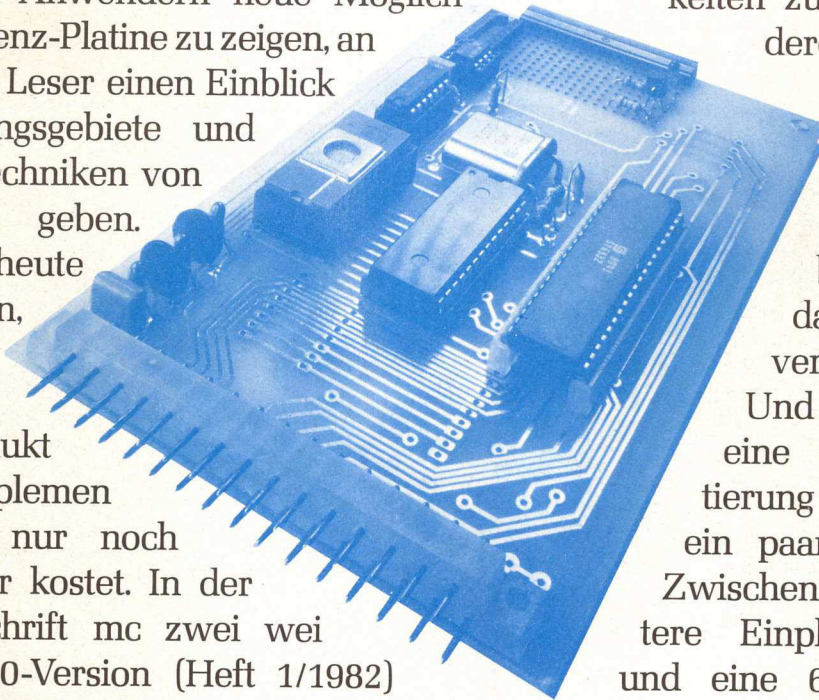
EINPLATINEN-MIKROCOMPUTER FÜR
UNIVERSELLE FESTPROGRAMM-ANWENDUNG.



Vorwort

W. Römer
Erdweg 7
4242 Rees 3

Seit mc in Heft 2/1981 den EMUF, den „Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung“, veröffentlicht hat, wurden viele interessante Anwendungen damit realisiert. Einige von ihnen sind im vorliegenden Sonderheft zusammengefaßt – einerseits, um EMUF-Anwendern neue Möglichkeiten zum Einsatz der Intelligenz-Platine zu zeigen, andererseits aber auch, um dem Leser einen Einblick in die vielfältigen Anwendungsgebiete und typischen Pro-grammiertechniken von Einplatinen-Com-putern zu geben. Einplatinen-Com-puter können heute bereits so preiswert gebaut werden, daß sich ihr Einsatz an verdrahteter Logik häufig lohnt. Und gleichzeitig wird das fertige Produkt eine Spur intelligenter – weil die Implementierung zusätzlicher Geräte- ein paar EPROM-Speicher- funktionen nur noch Zwischenzeit veröffentlichte ein paar EPROM-Speicher- zellen mehr kostet. In der tere Einplatinen-Computer – die Zeitschrift mc zwei wei- und eine 6502-Version mit eine Z80-Version (Heft 1/1982) – und eine 6502-Version mit zwei VIA-Bausteinen 6522 (Heft 2/1982). Der EMUF hat jedoch nicht zuletzt wegen seines extrem geringen Preises von kaum 100 DM nichts von seiner Aktualität eingebüßt. Viele der abgedruckten Programme lassen sich mit nur geringen Adressen-Änderungen für zahlreiche andere Computer auf 6502-Basis umschreiben. Dieses Heft enthält dafür auch die nötigen Adressentabellen.



Ihre
Redaktion

6502-Programmieren in Assembler

Von Lance A. Leventhal. 600 Seiten, kart. 59 DM. Te-Wi-Verlag, Theo-Prosel-Weg 1, 8000 München 40. ISBN 3-921803-10-1

Dieses Buch war bisher unter dem Titel „6502 Assembly Language Programming“ nur in englischer Sprache erhältlich. Da es zu den umfassendsten Einführungen in die 6502-Programmierung gehört, ist es um so erfreulicher, daß es nun auch in Deutsch zur Verfügung steht. Der Verfasser geht zunächst allgemein auf binäre, oktale und hexadezimale Zahlensysteme ein, beschreibt die Wirkungsweise von Assemblern und gibt eine Übersicht von 6502-Befehlen und Adressierungsarten. Da man aus Beispielen stets am meisten lernt, zeigt Leventhal die Wirkung unterschiedlicher Befehle an einer Reihe kurzer Beispielprogramme und stellt dem Leser nach jedem Kapitel auch kleine Programmier-Aufgaben. Außer dem CPU-Befehlssatz wird auch der Umgang mit Peripherie-Bausteinen wie 6520, 6522, 6530 und dem im EMUF enthaltenen Multifunktions-Baustein 6532 besprochen. Auch dazu dient wieder eine Reihe von Beispielen, wie LED-Ansteuerung, Tastenabfrage, Siebensegment-Codierung und serielle Ausgabe. Ein weiteres Kapitel ist der Interrupt-Programmierung gewidmet – eine nützliche Sache, an die sich leider viele nicht heranwagen.

Anwendungsbeispiele für den Mikroprozessor 6502

Von Herwig Feichtinger. 96 Seiten, 40 Abbildungen. RPB Nr. 173. Kart. 8.80 DM. Franzis-Verlag, Postfach 37 01 20, 8000 München 37. ISBN 3-7723-1731-6

Das Bändchen aus der Reihe „Radio-Praktiker-Bücherei“ ist nicht als Einführung in die 6502-Programmierung gedacht, sondern enthält eine Reihe von Beispielprogrammen, die weitgehend für die Adressenbelegung des bekannten Mikrocomputers KIM-1 ausgelegt sind. Ein Umschreiben auf andere Computertypen wird aber durch eine ausführliche Dokumentation der KIM-1-Adressenbelegung erleichtert. Zahlreiche darin enthaltene Programme lassen sich sehr leicht für den EMUF adaptieren, da der KIM-1 über zwei 6530-Bausteine verfügt, die exakt die gleiche Timer- und I/O-Struktur besitzen wie der EMUF-Baustein 6532: Man braucht nur die KIM-Adressen 1700...170F durch die entsprechenden EMUF-Adressen 0800...080F zu ersetzen. Der Autor beschreibt zunächst kurz die Systeme KIM-1, SYM-1, AIM-65 und PC-100 sowie die Programmierung des Timers im 6530/6532; es folgen eine Übersicht „inoffizieller“ 6502-Befehle, einige Hardware-Tips sowie Applikationen wie Funktionsgenerator, Speichervorsatz für Oszilloskope, Uhr, Druckerausgabe und anderes.

6502 Software Design

Von Leo. J. Scanlon. 270 Seiten, zahlreiche Tabellen und Listings, englisch, kart. 29 DM. Howard W. Sams & Co., Indianapolis, USA; in Deutschland beziehbar von R. Löhner, 2070 Ahrensburg, oder Rockwell. ISBN 0-672-21656-6

Der Autor Scanlon ist Documentation Manager bei Rockwell, Second-Source-Hersteller der verbreiteten 8-Bit-CPU 6502, die in solchen Computern wie PET, CBM, Apple, AIM-65 oder PC-100 eingebaut ist. Das (bisher leider nicht in Deutsch erhältliche) Buch ist hervorragend geeignet, um sich effizient in die Maschinensprache-Programmierung dieser CPU einzuarbeiten. Die darin verwendeten Programmierbeispiele nehmen meist bezug auf das in den Geräten AIM-65 und PC-100 vorhandene Monitor-Programm, was Ein- und Ausgaberroutinen usw. angeht, lassen sich aber dank der guten Beschreibung der Adressenbelegung des AIM leicht auf andere Systeme übertragen. Scanlon beginnt mit einer groben Beschreibung des 6502-Befehlssatzes, stellt Unterprogramm- und Interrupt-Techniken vor, zeigt, wie man die Grundrechenarten auf einer 8-Bit-CPU realisiert und geht schließlich auf die Programmierung des Schnittstellenbausteins 6522 (VIA) ein. Am Schluß des Buches finden sich übersichtliche Tabellen mit dem ASCII-Zeichensatz und den 6502-Befehlscodes, wobei auch die jeweilige Statusregister-Beeinflussung angegeben ist.

Assembler-Handbuch PC-100

Hrsg.: Siemens AG. 122 Seiten, zahlreiche Listings, kart. 12 DM. Siemens AG, Bauteile-Service, Postfach 146, 8510 Fürth.

Obwohl das PC-100-Assembler-Handbuch eigentlich speziell für den Tischcomputer PC-100 geschrieben wurde, der mit Rockwells AIM-65 intern baugleich und softwarekompatibel ist, handelt es sich dabei um eine der wenigen guten deutschsprachigen Darstellungen des 6502-Befehlssatzes (identisch mit dem des EMUF-Prozessors 6504) sowie des Umganges mit der 65XX-Assembler-Syntax. Das Buch ist daher auch für Besitzer anderer Computer auf 6502-Basis sehr nützlich, wenn man die Hürde ausgiebiger Kenntnisse der englischen Fachsprache umgehen möchte. Ausführlich werden die speziellen Adressierungsarten der 65XX-Prozessorfamilie beschrieben. Eine 31seitige Übersicht aller Befehle zeigt auf einen Blick die Quellencodierung (Mnemonics, alphabetisch geordnet), die zu den jeweils möglichen Adressierungsarten gehörigen Operationscodes, die Anzahl der Bytes, die Ausführungszeit sowie die Beeinflussung der Flag-Bits im Statusregister. Weitere Kapitel gehen auf die Befehle des PC-100-Monitorprogramms sowie auf den Umgang mit dem optionalen 4-KByte-Assembler-ROM ein.

Die hier aufgeführten Bücher wurden ausgewählt, weil sie auf für den Einplatinen-Computer EMUF typische Programmier-techniken besonders ausführlich eingehen. Mitte 1982 erscheint ferner im Franzis-Verlag das Buch „Mikrocomputer ohne Ballast“ (Immerzeel), das eine sehr brauchbare systemneutrale Einführung in die 6502-Programmierung darstellt.



Der Whisky-EMUF

Eine Aquarium- oder Kfz-Scheibenwasch-Pumpe, ein EMUF, eine Hexadezimal-Tastatur und ein paar Kleinteile genügen, um einen spaßigen Spielautomaten zur Belustigung Ihrer Party-Gäste zu bauen. Der Witz dabei: Jeder, der innerhalb einer gewissen Zeit eine Binär-Zufallszahl in eine Hex-Ziffer umrechnen kann, bekommt ein Gläschen Whisky geschenkt. Allerdings paßt der EMUF die zulässige Reaktionszeit dem durchschnittlichen Intelligenz-Quotienten der Benutzer an!

Seite 40

Interface für Typenrad-Schreibmaschine

Wenn Sie einen Computer mit serielltem Ausgang besitzen (TTY- oder V.24-Schnittstelle) und Briefe, Daten oder Programme in Briefqualität ausdrucken möchten, so brauchen Sie dafür nur noch rund 1200 DM investieren. So viel kostet nämlich die Typenrad-Schreibmaschine zusammen mit einem EMUF-Bausatz. Der EMUF dient hier als intelligentes Interface zur Codeumsetzung und wickelt das V.24-Handshake-Protokoll ab.

Seite 46

| | |
|---|----|
| Vorwort | 3 |
| mc-bücher | 4 |
| mc-grundlagen | |
| Glossarium | 6 |
| So entsteht eine EMUF-Applikation | 15 |
| EMUF-Programmiertips | 16 |
| mc-hard | |
| Mädchen für alles | 8 |
| Über die EMUF-Chips | 12 |
| AIM-65 emuliert EMUF | 14 |
| AIM schießt EPROM | 21 |
| Programmierte EPROMs für den EMUF | 29 |
| Wenn der EMUF streikt | 33 |
| EMUF-AIM-Adapterkabel | 45 |
| Ein Netzteil für den EMUF | 52 |
| EMUF mit erweiterter Adressierung | 66 |
| mc-applikation | |
| Standardschnittstellen für Schreibmaschinen-Drucker | 18 |
| EMUF bringt Strichcode zum IEC-Bus | 22 |
| Türklingel und Alarmanlage | 26 |
| Mini-Datenlogger | 30 |
| EMUF morst Rufzeichen | 36 |
| EMUF steuert Relaisfunkstelle | 38 |
| Der Whisky-EMUF | 40 |
| Der LCD-EMUF | 42 |
| Interface für Typenrad-Schreibmaschine | 46 |
| Funkfernseh-Empfänger | 53 |
| V.24-Schnittstellentester | 56 |
| EMUF als DCF-77-Decoder | 60 |
| V.24-Interface | 62 |
| mc-soft | |
| D/A-Wandlung per Software | 28 |
| AIM steuert Typenrad-EMUF | 55 |
| Formatierte Assembler-Listings mit dem AIM-65 | 61 |

Impressum: 1982, Franzis-Verlag GmbH, Karlstraße 37, D-8000 München 2. Produktion: ELVAG, Elektronik-Verlag Luzern AG, CH-6002 Luzern.

Bearbeitet von der Redaktion der Zeitschrift mc. Für den Text verantwortlich: Dipl.-Ing. (FH) Herwig Feichtinger.

© Sämtliche Rechte – besonders das Übersetzungsrecht – an Text und Bildern vorbehalten. Fotomechanische Vervielfältigung nur mit Genehmigung des Verlages.

Jeder Nachdruck, auch auszugsweise, und jede Wiedergabe der Abbildungen, auch in verändertem Zustand, sind verboten.

ISSN 0722-0022. Druck: Stämpfli, CH-Bern. ZV-Art.-Nr. 80041. St/ZV/282/568/7'

Herwig Feichtinger

Glossarium

Hier finden Sie die wichtigsten in diesem Sonderheft gebrauchten Fachausdrücke in alphabetischer Reihenfolge. Die Erläuterungen beziehen sich im wesentlichen auf die Hard- und Softwarekonfiguration des EMUF, um den es in diesem Heft ja geht, und auf geeignete Entwicklungssysteme.

ASCII

American Standard Code for Information Interchange; 7-Bit-Code zur Darstellung von Schriftzeichen (Ziffern, Satzzeichen, Klein- und Großbuchstaben, Steuerzeichen).

Assembler

Hilfsprogramm, oft als ROM lieferbar, zum automatischen Übersetzen eines in Mnemonics geschriebenen Quellen-Programms in den für die CPU verständlichen Objektcode. 2-Pass-Assembler tun dies in zwei Durchläufen und gestatten die Verwendung von Namen (Labels, Symbolen) statt absoluter Adressen.

Assembler-Listing

Von einem Assembler während des Übersetzungsvorganges ausgegebenes Kontroll-Listing mit Adressen, Symbolen, Operationscodes, Mnemonics und Kommentaren. Es zeigt nebeneinander Quellen- und Objektcode.

Baud

Einheit für die Übertragungsgeschwindigkeit: 1 Baud bedeutet einen Schritt pro Sekunde, bei digitaler Übertragung 1 Bit/s. Bei ASCII sind 110, 150, 300, 600, 1200, 2400, 4800 und 9600 Baud üblich, beim Baudot-Code dagegen 45,5, 50, 75 und 100 Baud.

Baudot-Code

Im Fernschreibverkehr verwendeter 5-Bit-Code zur Zeichendarstellung. Es gibt zwei „Zeichenebenen“, zwischen denen mit besonderen Steuerzeichen hin- und hergeschaltet werden kann, nämlich eine Buchstaben- und eine Ziffernebene. Es ist nur Groß- bzw. nur Kleinschreibung möglich.

CPU

Central Processing Unit, Zentraleinheit eines Computers; bei Mikrocomputern ist die CPU der Mikroprozessor.

D/A-Umsetzer

Digital/Analog-Umsetzer. Er erzeugt eine zu einem binären Zahlenwert (z. B. einem Byte) proportionale Ausgangsspannung.

Debugging

Fehlersuche in noch nicht ganz fertigen Programmen. Für das Debugging enthalten die Monitorprogramme von Entwicklungssystemen meist Hilfsmittel, z. B. Breakpoints, Trace-Einrichtungen und Single-Step-Betriebsarten.

Editor

Hilfsprogramm im Entwicklungssystem zur Eingabe und Änderung des Quelltextes.

EMUF

Einplatinen-Mikrocomputer für universelle Festprogrammanwendung; Minimalconfiguration eines Mikrocomputers.

Entwicklungssystem

Computer, dessen Hauptzweck die Entwicklung von Programmen ist. So lassen sich Geräte wie KIM-1, AIM-65, PC-100, CBM oder Apple als Entwicklungssystem für den EMUF einsetzen. Voraussetzung ist dazu das Vorhandensein eines Monitorprogramms sowie möglichst auch eines Assemblers.

EPROM

Erasable Programmable Read-Only Memory, mit ultravioletttem Licht löschbarer Festwertspeicher. Üblich sind heute die Typen 2758 (1 KByte), 2716 (2 KByte) und 2732 (4 KByte) mit 5 V Betriebsspannung und etwa 25 V Programmierspannung.

Europakarte

Platine mit der genormten Größe 100 × 160 mm² und einer Steckleiste an einer Schmalseite (oder an beiden Schmalseiten).

Hex-Dump

Darstellung eines Speicherbereichs in hexadezimaler Form. Am Beginn jeder Zeile steht eine vierstellige Anfangsadresse, gefolgt von den aus je zwei Hex-Zeichen bestehenden Bytes ab dieser Adresse. Hex-Dumps dienen u. a. der platzsparenden Veröffentlichung.

Hexadezimal-Darstellung

Bei Maschinensprache und Assemblern übliche Form der Zahlendarstellung mit den „Ziffern“ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, die den Dezimalzahlen 0...15 entsprechen. Hex 10 ist dezimal 16, hex FF ist dezimal 255.

I/O-Port

Input/Output-Port, 8-Bit-Parallelschnittstelle, bei der sich einzelne Bits (Leitungen) als Ein- oder Ausgang per Programm deklarieren lassen. So kann man z. B. mit vier Bits ein Tastenfeld abfragen und die übrigen vier Bits zur Ansteuerung einer Anzeige verwenden.

IEC-Bus

Auch IEEE-488, HP-IB, GPIB genannt; Normschnittstelle zur Verbindung eines Controllers (Computer) und bis zu 15 Peripheriegeräten miteinander. Ein Datenaustausch kann in beiden Richtungen erfolgen, und der Controller kann jedes Peripheriegerät einzeln adressieren.

Interface

Gerät oder Baugruppe zur Verbindung von zwei Geräten mit unterschiedlichen Schnittstellen-Normen. Einfache Interfaces, z. B. TTY auf RS-232, enthalten nur eine Pegelanpassung; in anderen Fällen, z. B. von einer parallelen ASCII-Schnittstelle eines Computers auf einen Baudot-Fernschreiber mit TTY-Eingang, kann auch ein kleiner Mikrocomputer wie der EMUF als Interface dienen.

Interrupt

Anhalten des gerade laufenden Programms durch einen Hardware-Impuls an einem CPU-Anschluß. Die CPU führt daraufhin ein Interrupt-Programm aus und kehrt dann zum nächsten Befehl des vorher verlassenen Hauptprogramms zurück. Beim 6502 gibt es NMI (non-maskable interrupt) und IRQ (interrupt request), beim 6504 nur IRQ.

Maschinensprache

Diejenige Programmiersprache, die die CPU direkt versteht, d. h. binäre bzw. hexadezimal geschriebene Befehle, bestehend aus den CPU-Operationscodes und ihren Argumenten.

Minimalkonfiguration

Mikrocomputer, bei dem alles weggelassen wurde, was für eine bestimmte Anwendung nicht nötig ist; typisches Beispiel ist der EMUF mit nur 128 Byte RAM und 1 KByte EPROM.

Mnemonics

Abkürzungen für Maschinensprache-Befehle, wie sie im Quelltext verwendet werden und die man sich leichter merken kann als die hexadezimalen Operationscodes. 65XX-Mnemonics bestehen stets aus drei Buchstaben, z. B. LDA, STA, TAX, JMP usw.

Monitorprogramm

Gewöhnlich fest in einem Entwicklungssystem gespeichertes Hilfsprogramm zum Eingeben, Ändern, Abspeichern, Laden und Starten eines Maschinensprache-Programms, meist für hexadezimale Eingabe über eine Tastatur.

Objektcode

Das von einem Assembler aus dem Quellcode übersetzte Maschinensprache-Programm, wie es auf dem jeweiligen Computer direkt ablauffähig ist.

Operationscode

Binär oder hexadezimal geschriebener Maschinensprache-CPU-Befehl, z. B. hex A9 für „Load Accu Immediate“. Außer dem Operationscode gehört zum vollständigen Befehl (bei Mehrbyte-Befehlen) noch das Argument, z. B. ein Wert oder eine Adresse.

Parity-Bit

Als achttes Bit an ein 7-Bit-ASCII-Zeichen angehängtes Prüfbit als auf ein Bit begrenzte Summe der sieben Datenbits, um Fehler bei Datenübertragungen erkennen zu können.

Peripheriebaustein

Baustein in einem Mikrocomputer, der die Verbindung zur Außenwelt ermöglicht und der z. B. I/O-Ports enthält (RIOT, VIA).

Pointer

Ein oder zwei aufeinanderfolgende Speicherzellen, die auf eine Adresse „hinweisen“, z. B. für die indirekte Adressierung mancher CPU-Befehle.

Pull-Up-Widerstand

Widerstand (z. B. 1...10 k Ω), der zwischen einem hochohmigen Eingang z. B. einem I/O-Port und der positiven Betriebsspannung (+5 V) liegt und dafür sorgt, daß dieser Eingang auf High-Pegel

(log. 1) liegt, wenn er sonst nicht beschaltet wird oder z. B. an ihm ein offener Schalter nach Masse (0 V) liegt.

Quelltext

In Mnemonics und mit symbolischen Adressen geschriebenes Programm, das vom Assembler in ein Maschinenprogramm (Objektcode) übersetzt werden kann.

RAM

Random Access Memory, Schreib/Lese-Speicher, der Arbeitsspeicher in einem Computer.

Relokatibel

Ohne Anpassung z. B. von Sprungbefehlen frei im Speicher verschieblich. Relokatable Programme dürfen intern ausschließlich relative Sprungbefehle verwenden.

Reset

Hardware-Impuls an einem CPU-Anschluß, der dafür sorgt, daß (bei 65XX-Prozessoren) das Programm an jener Stelle begonnen wird, deren Adresse in den Speicherzellen hex FFFD und FFFC steht. Ein Reset ist beim Einschalten des Computers erforderlich, um einen definierten Programmstart sicherzustellen.

RIOT

RAM-I/O-Timer, Multifunktionsbaustein (z. B. 6532), der mehrere Aufgaben in einem Chip vereint und so den Bauelemente-Aufwand eines Mikrocomputers verringert.

ROM

Read-Only Memory, Festwertspeicher, meist zur Speicherung des Betriebsprogramms eines Mikrocomputers, z. B. des Monitorprogramms eines Entwicklungssystems. Da die Programmierung schon beim IC-Hersteller erfolgt, sind ROMs im Gegensatz zu EPROMs nur bei größeren Stückzahlen eines Programms sinnvoll.

RS-232

Amerikanische Norm für serielle Schnittstelle mit einer Steuerleitung (Busy), die dem Datensender mitteilt, ob der Datenempfänger noch mit der Verarbeitung der letzten Information beschäftigt ist. Für log. 0 werden -3...-15 V, für log. 1 dagegen +3...+15 V übertragen.

Stack

Bei der 65XX-CPU-Familie im Adressenbereich 0100...01FF befindlicher Speicherbereich, der für die Speicherung von Unterprogramm- und Interrupt-Rücksprungadressen reserviert ist.

Symboltabelle

Optional vom Assembler nach dem Assemblerlisting ausgedruckte Tabelle mit den Symbolen (Label-Namen) und den dazugehörigen absoluten Hexadezimal-Adressen.

Timer

Baustein zur Steuerung von Zeitabläufen. Der EMUF enthält zwei Timer: einen (555) zur Erzeugung des Reset-Impulses, und einen per Software programmierbaren im RIOT 6532, der z. B. nach Ablauf einer bestimmten Zeit einen Interrupt auslösen kann.

TTL-Pegel

TTL heißt Transistor-Transistor-Logik, eine IC-Technologie mit den Pegeln 0...0,8 V für log. 0 und 2...5 V für log. 1. Ein Eingang eines TTL-IC stellt eine Belastung von etwa 5 k Ω dar. I/O-Ports sind meist TTL-Pegel-kompatibel aufgebaut.

TTY

Genormte Schnittstelle für die serielle Datenübermittlung (Teletype-Stromschleife): „0“ bedeutet, es fließt kein Strom, und „1“, es fließen etwa 20 mA. Da TTY-Schnittstellen meist nicht Masse als Bezugspotential verwenden, ist der Einsatz von Optokopplern zur Potentialtrennung üblich.

V.24

Europäische Äquivalent-Norm zu RS-232, allerdings mit geringen Abweichungen, z. B. bezüglich der Polarität der Busy-Leitung.

Vektor

Zwei aufeinanderfolgende Speicherzellen (16 Bit), die eine Adresse für einen indirekten Sprungbefehl enthalten. Z. B. ist bei den 65XX-CPUs die Adresse des Interrupt-Programms in den Speicherzellen FFFF und FFFE abzulegen; diese Zellen stellen also den IRQ-Vektor dar.

VIA

Versatile Interface Adapter; Bezeichnung für den Baustein 6522, der zwei Timer, ein Schieberegister und zwei I/O-Ports enthält und z. B. in den Computern AIM-65, PC-100 und CBM eingebaut ist.

Zero Page

Speicherbereich hex 0000...00FF, auf den die 65XX-CPU mit speicherplatzsparenden 2-Byte-Befehlen zugreifen können – gegenüber 3-Byte-Befehlen für den übrigen Speicherraum. Auch sind bestimmte Adressierungsarten nur auf Zero-Page-Zellen anwendbar.

Herwig Feichtinger

Mädchen für alles

Was hier im folgenden vorgestellt wird, ist ein fest zu programmierender, sehr preiswerter Mikrocomputer, der sich zum Beispiel als Drucker-Interface, intelligentes Bedienteil für Meßgeräte, Frequenzgenerator, Schaltuhr, Codeumsetzer oder für tausend andere Zwecke einsetzen läßt. Die Programme für ihn lassen sich mit preiswerten Tischcomputern auf 6502-Basis wie Apple, PET, CBM, AIM-65, PC-100 oder KIM-1 entwickeln; Beispiele dafür folgen in den nächsten Heften.

Wenn man von Computern spricht, meint man meist Geräte, die sich frei programmieren lassen, mit denen man eigene Programme entwickeln und testen kann und die über eine Tastatur sowie über einen Bildschirm oder we-

nigstens ein einfaches Display verfügen. Solche Computer bekommt man heute schon für weniger als 1000 DM. Hier wird aber etwas ganz anderes vorgestellt, nämlich ein Mikrocomputer, der nur einmal und vor allem fest pro-

grammiert und dann für einen ganz bestimmten Verwendungszweck eingesetzt wird (Bild 1). Er ist also in keiner Weise dafür konstruiert, Programme mit ihm zu entwickeln, als Lehr- und Lerncomputer zu dienen oder später mit zusätzlichem Speicherplatz, ja vielleicht sogar mit einem Basic-Interpreter erweitert zu werden.

Ein Computer für weniger als hundert Mark

Unser Computerchen ist also dafür gedacht, überall dort eingesetzt zu werden, wo es im Grunde nur als Ersatz für eine vielleicht recht umfangreiche, undurchsichtige Digitalschaltung dient. So etwa in einer numerischen Steuerung, in einer Schaltuhr, in einem rechnenden Meßgerät usw., wo der Benutzer nicht selbst programmiert.

Dieses Konzept gestattet es, einen Mikrocomputer als Minimalkonfiguration mit absichtlichem Verzicht auf spätere Erweiterbarkeit und gleichzeitig als äußerst preiswerte Schaltung aufzubauen. Natürlich gibt es für diesen Zweck auch Ein-Chip-Mikrocomputer, z. T. sogar mit UV-löschbaren EPROMs – aber: ein Entwicklungssystem für einen solchen Computer kostet leider -zigtausend Mark. Bei geringen Stückzahlen treten daher enorme Kostenbelastungen auf, die die Verwendung der Ein-Chip-Mikrocomputer wieder oft als fraglich erscheinen lassen.

Unser Mikrocomputer arbeitet daher mit einer CPU, die es zuläßt, die benötigten Programme mit preiswerten Tischcomputern zu entwickeln, so etwa mit CBM, PET, AIM-65, Apple-II usw., die alle mit dem Mikroprozessor 6502 arbeiten. Bei der Übertragung des Programms auf das EPROM, das in unser Computerchen gesteckt wird, brauchen dann lediglich noch einige Adressen geändert zu werden. Zum Beispiel diejenigen für die I/O-Ports. Verwendet man einen Assembler für die Programmentwicklung, so braucht man das nicht einmal einzeln von Hand zu tun.

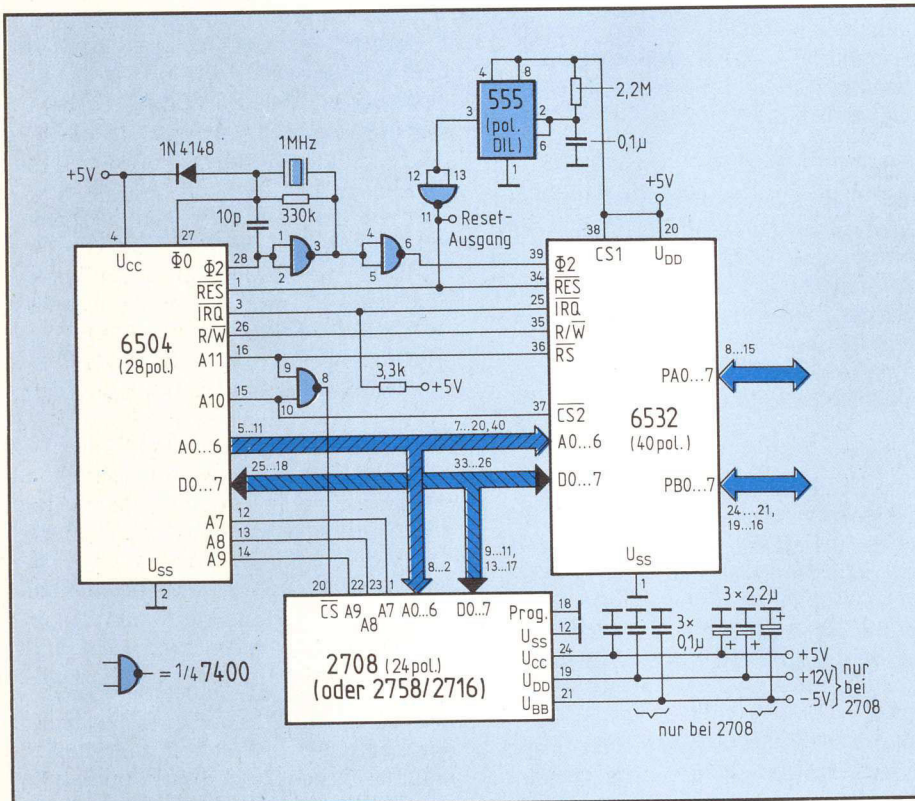


Bild 1. Gesamtschaltung des 6504-Computers mit 1 KByte ROM, 128 Byte RAM, einem programmierbaren Interrupt-Timer und 16 I/O-Leitungen

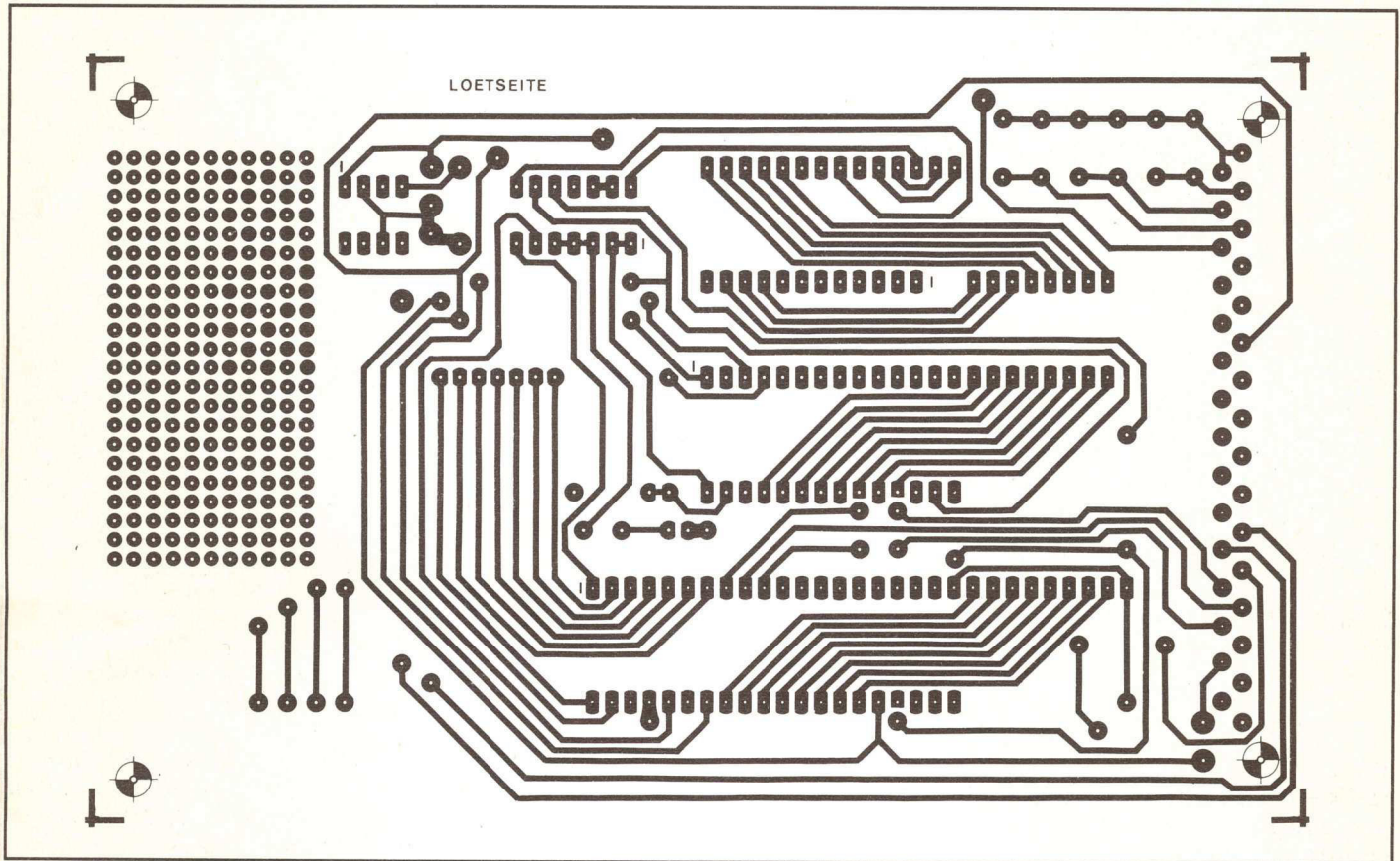


Bild 2. Lötseite der Platine. Sie enthält ein Lochraster-Feld, das vom Anwender für besondere Aufgaben frei verdrahtet werden kann, z. B. für die Nachrüstung eines D/A-Wandlers

Die Drei-Chip-Lösung hat es in sich

Wegen der Verbreitung des Prozessors 6502 bei den preiswerteren Tischcomputern wurde eine CPU aus dieser Familie gewählt, nämlich der Typ 6504. Er unterscheidet sich von der „Mutter“ 6502 dadurch, daß er statt 16 nur 12 Adressenleitungen besitzt, nur einen Interrupt-Eingang herausführt (IRQ), in einem 28-Pin-Gehäuse untergebracht ist (6502: 40 Pins) und nicht zuletzt deshalb auch preiswerter ist.

Wie der geneigte Leser weiß, braucht man in einem Mikrocomputer neben der CPU noch drei Dinge, nämlich einen Arbeitsspeicher (RAM), einen Eingabe/Ausgabe-Baustein (I/O), über den die Verbindung zur Außenwelt hergestellt wird und der somit dafür sorgt, daß der Computer kein Selbstzweck ist, sowie einen Programmspeicher, der hier gemäß dem Verwendungszweck als Festwertspeicher ausgeführt ist.

Um die Chip-Anzahl gering zu halten, findet hier ein Baustein namens 6532

Verwendung, der nicht nur zwei 8-Bit-I/O-Ports sowie 128 Byte RAM enthält, sondern auch einen für mancherlei Zwecke äußerst nützlich programmierbaren Interrupt-Timer, der Zeiten bis zu 261 ms liefern kann. Dazu wird nun nur noch ein EPROM benötigt, das das Betriebsprogramm enthält – in unserem Fall z. B. ein 1-KByte-Typ namens 2758, der ebenfalls schon recht preiswert zu haben ist.

Reicht denn das wirklich aus?

Wenn hier von kläglichen 128 Byte RAM und 1 KByte EPROM die Rede ist, wird manch Tischcomputer-Benutzer sagen, was soll ich damit schon anfangen? Für einen Basic-Computer wäre das tatsächlich viel zu wenig, denn allein ein Basic-Interpreter belegt ja schon rund 4..12 KByte ROM bzw. EPROM. Da Basic aber für die meisten Steuerungszwecke und für zeitkritische Aufgaben völlig ungeeignet ist, wird unser Mikro-Mikrocomputer in der Maschinensprache des verwendeten Prozessors programmiert.

Hier sei gleich vermerkt, daß der 6504 genau den gleichen Befehlssatz wie sein großer Bruder 6502 besitzt und somit zumindest softwaremäßig keinerlei Einschränkungen unterliegt. Und in 1 KByte bringt man z. B. schon ein kleines Schachprogramm unter, ein Programm zur Ansteuerung einer Schreibmaschine über eine serielle Schnittstelle, die Software zum Betrieb eines „dummen“ Matrixdruckers oder vieles andere mehr. Übrigens sitzt solch ein 6504-Prozessor auch in der Floppy-Disk-Einheit CBM-3032 von Commodore – auch das ist eine Steueraufgabe, die mit einer Mikrocomputer-Minimalkonfiguration wunderbar zu lösen ist. Also keine Angst vor zu wenig Speicherplatz!

Adressierungs-Kniffe müssen sein

6502-Kenner wissen, daß dieser Prozessor zwei besondere Speicherbereiche besitzt, die beide vorhanden sein müssen, aber hardwaremäßig in ihrer Adressenlage leider mehr als 128 Bytes auseinanderliegen. Unsere 128 Byte zusammen-

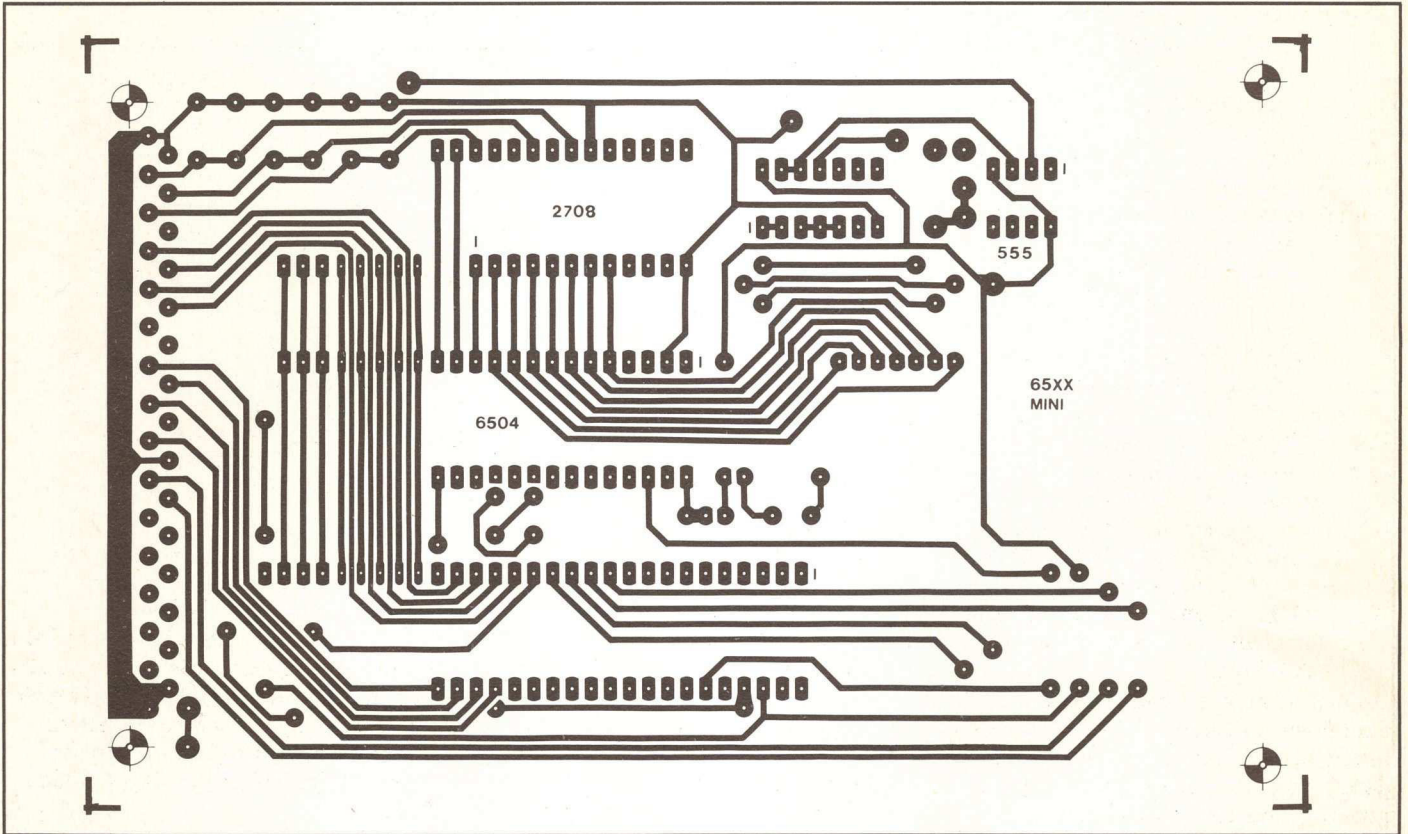


Bild 3. Bestückungsseitige Leiterbahnen der (doppelseitigen, durchkontaktierten) Platine. Die 31polige Steckerleiste ist später auf diese Seite zu löten

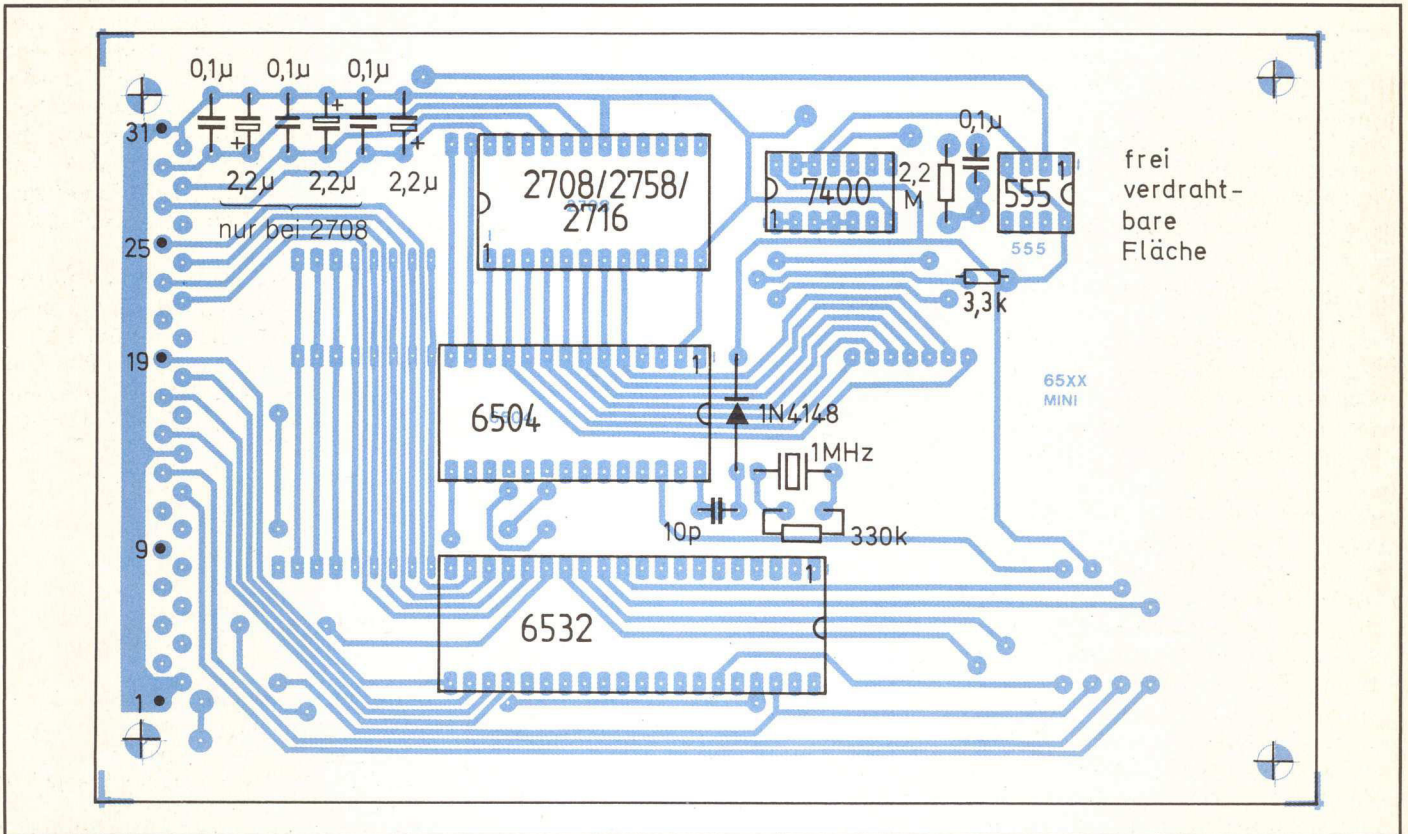


Bild 4. Bestückungsplan des 6504-Computers. Es sei erwähnt, daß das 2-KByte-EPROM 2716 z. T. schon preiswerter angeboten wird als der 5-V-/1-KByte-Typ 2758. Beim 2716 kann man entweder eine Hälfte „verschenken“ oder auch mit einem Schalter zwischen zwei 1-KByte-Betriebsprogrammen wählen

hängendes RAM würden dafür nicht ausreichen: Wir brauchen einen Bereich in der „Zero Page“ (0000...00FF), die nützliche Adressierungsarten bei vielen Maschinensprache-Befehlen des 6502 und die Verwendung speichersparender 2-Byte-Befehle ermöglicht, und einen weiteren in der Page 1 (0100...01FF), der die für Unterprogrammssprünge erforderlichen Rücksprungadressen speichert und gemeinhin als Stack bezeichnet wird.

Dieses Problem wurde hier aber auf eine listige Art umgangen: nämlich mit der sonst mit Recht verpönten Technik, den Adressenbus nicht vollständig zu decodieren und dadurch Speicherplätze scheinbar an mehreren Adressen gleichzeitig erscheinen zu lassen. Und so erscheinen unsere 128 Byte RAM nicht nur an den Zero-Page-Adressen 0000...007F, sondern – mit dem gleichen Speicherinhalt – bei 0180...01FF, also im Stack-Bereich.

Dabei muß man nur bedenken, daß das Schreiben z. B. an die Adresse 01FE den Inhalt bei 007E gleichermaßen verändert. Man muß sich also beim Programmieren überlegen, wieviel Platz man für Unterprogrammssprünge in Stack und wieviele Bytes man in der Zero Page benötigt. Die Verteilung der 128 Bytes RAM könnte dann typischerweise so aussehen, daß 01F0...01FF als Stack dient, um maximal sechs Unterprogramm-ebenen plus eine Interrupt-Ebene zuzulassen, und 0000...006F als frei verwendbarer Zero-Page-Bereich.

Der anderswo große Nachteil, daß eine Systemerweiterung wegen der unvollständigen Adressendecodierung schwierig ist, wurde hier im Interesse möglichst geringer Hardware-Kosten bewußt in Kauf genommen.

Die restliche Adressenbelegung entstand ebenfalls unter diesem Aspekt; es ist nur noch ein einziges TTL-IC nötig, um die Decodierung der Adressen vorzunehmen. Die genaue Zuordnung geht aus **Tabelle 1** hervor.

Die Eigenschaft der Adressenduplizierung kann u. U. auch einen Vorteil darstellen. Denn nicht immer steht in dem Tischcomputer, der zur Entwicklung des Programms Verwendung findet, derjenige Adressenbereich zur Verfügung, in dem der EPROM-Bereich unseres kleinen Systems eigentlich liegt. Möglicherweise besitzt der Tischcomputer aber einen Speicherbereich, der identisch mit einem duplizierten Bereich des EPROM ist. Eine Adressenanpassung ist dann nicht mehr nötig. Dies gilt selbstverständlich auch für die Zero-Page- und Stack-Bereiche.

Tabelle 1: Adressenbelegung des 6504-Computers

| Adressenbits | Inhalt | Adressenbereiche |
|----------------------------------|-------------------------------------|--|
| 00XX XAAA AAAA | 128 Byte RAM im 6532 | 000...07F; 080...0FF; 100...17F; 180...1FF; 200...27F; 280...2FF; 300...37F; 380...3FF 800...81F u.a. (32mal dupliziert bis BFF) |
| 10XX XXXA AAAA | I/O-Ports und Timer im 6532 | C00...FFF |
| 11AA AAAA AAAA 01AA AAAA AAAA | EPROM (1 KByte) Expansion (1 KByte) | 400...7FF |

(A = gültiges Adressen-Bit, X = ignoriertes Adressenbit)

6532-Adressen: 800 = Port A, 801 = Port-A-Richtungsregister, 802 = Port B, 803 = Port-B-Richtungsregister; 814 = Timer 1 µs, 815 = Timer 8 µs, 816 = Timer 64 µs, 817 = Timer 1024 µs; 81C...81F wie 814...817, jedoch mit Interrupt bei abgelaufener Zeit. Timer auslesen: 816; Timer testen: 817 (N-Flag).

Die Inbetriebnahme des Systems

Nehmen wir an, wir hätten ein EPROM mit dem nötigen Betriebsprogramm für unseren individuellen Verwendungszweck programmiert. Dann können wir alle Bauelemente auf die doppelseitige durchkontaktierte Epoxy-Platine löten (Bilder 2 bis 4; beziehbar u. a. bei Fa. Walter, Am Starzenbach 9, 8069 Woln-

zach), wobei es sich dringend empfiehlt, für die drei LSI-ICs 6504, 6532 und 2758 Fassungen und eine 31polige Steckerleiste (**Tabelle 2**) zu verwenden. Einen Bausatz liefert die Firma Elektronikladen, Wilhelm-Mellies-Str. 88, 4930 Detmold 1.

Beim Anschalten der 5-V-Versorgungsspannung (Netzteil-Belastbarkeit min. 200 mA) erfolgt über das auf der Platine befindliche Monoflop automatisch ein Reset, so daß der Prozessor mit dem Arbeiten des Programms beginnt, dessen Startadresse in den Zellen FFFC (niederwertiges Byte) und FFFD (höherwertiges Byte) abgelegt ist. Diese Adressen gibt es in unserem System natürlich nicht wirklich; sie finden sich aber dupliziert am oberen Ende des EPROM-Bereichs bei 0FFC und 0FFD.

Im nächsten Heft werden wir ausführlich auf die Programmierung von Anwenderprogrammen für den 6504-Mikrocomputer eingehen und häufig benötigte Routinen für Tastaturabfrage, Display-Ansteuerung und Timer-Verwendung vorstellen.

Tabelle 2: Steckerbelegung

| |
|----------------------------|
| 1 Masse |
| 2 Masse |
| 4 IRQ |
| 6 PA0 |
| 7 PA1 |
| 8 PA2 |
| 9 PA7 |
| 10 PA6 |
| 11 PA5 |
| 12 PA4 |
| 13 PA3 |
| 14 Masse |
| 15 PB0 |
| 17 PB1 |
| 18 PB2 |
| 19 PB3 |
| 21 Reset-Ausgang |
| 22 PB7 |
| 23 PB6 |
| 24 PB5 |
| 25 PB4 |
| 26 Reset-Eingang |
| 27 + 5 V |
| 28 - 5 V (bei 2716 + 5 V) |
| 29 + 12 V (bei 2716 Masse) |
| 30 Masse |
| 31 Masse |

Literatur

- [1] R 6532 Data Sheet. Rockwell Doc. Nr. 29 000 D42.
- [2] R 650X Data Sheet. Rockwell Doc. Nr. 29 000 D39.
- [3] R 6500/6532 Timer Interrupt Precautions. Rockwell Doc. Nr. R 6500 N02.
- [4] EMUF-Programmiertips. mc 1981, Heft 2.
- [5] Bits und Bytes: 6502-Programmierung. Sonderheft „Hobbycomputer 2“, Franzis-Verlag.

Herwig Feichtinger

Über die EMUF-Chips

Der Einplatinencomputer EMUF ist mit dem weit verbreiteten Mikroprozessor 6502 voll software-kompatibel; seine CPU 6504 besitzt nämlich exakt den gleichen Befehlssatz und unterscheidet sich nur in der Zahl herausgeführter Pins. Ein paar Daten über sie, über EPROMs und über den schon recht komplexen Multifunktions-Baustein 6532 folgen in diesem Beitrag.

Auf der EMUF-Platine sitzen insgesamt drei „große“ Chips: der Mikroprozessor 6504, der das Hirn des Computers darstellt, der Multifunktionsbaustein 6532 mit 128 Byte RAM, einem Timer und zwei 8-Bit-I/O-Ports, sowie ein EPROM, bei dem der Anwender die Wahl zwischen den Typen 2708, 2758 und 2716 hat.

Die CPU 6504

Die Mikroprozessor-Familie 65XX umfaßt neben dem Standard-Chip 6502 zahlreiche weitere Typen, die sich in der Art der herausgeführten Taktleitungen und Interrupts sowie in der durch die Zahl der verfügbaren Adressenleitungen begrenzten Adressierfähigkeit unterscheiden. Die Hersteller MOS Technology, Commodore, Rockwell und Synertek

haben solche Beschränkungen deshalb eingeführt, um auch CPU-Versionen mit weniger Pins und deshalb in preiswerteren, kleineren Gehäuseformen anbieten zu können.

Der Mikroprozessor 6504 kann maximal 8 KByte adressieren; Bild 1 zeigt, daß er nur 28 Gehäusepins besitzt. Die Stromaufnahme ist bei 5 V Versorgungsspannung typisch knapp über 100 mA; eine CMOS-Version mit deutlich weniger Stromverbrauch steht seit neuestem alternativ zur Verfügung. Der Prozessor wird mit 1 MHz Taktfrequenz betrieben; eine 8-Bit-Addition kann er in 2 µs durchführen. Die minimale Taktfrequenz ist 100 kHz; sie ergibt sich aus der dynamischen Registerstruktur der 65XX-Familie. Alle Anschlüsse sind pegelmäßig TTL-kompatibel und können eine TTL-Normallast treiben. Die „Com-

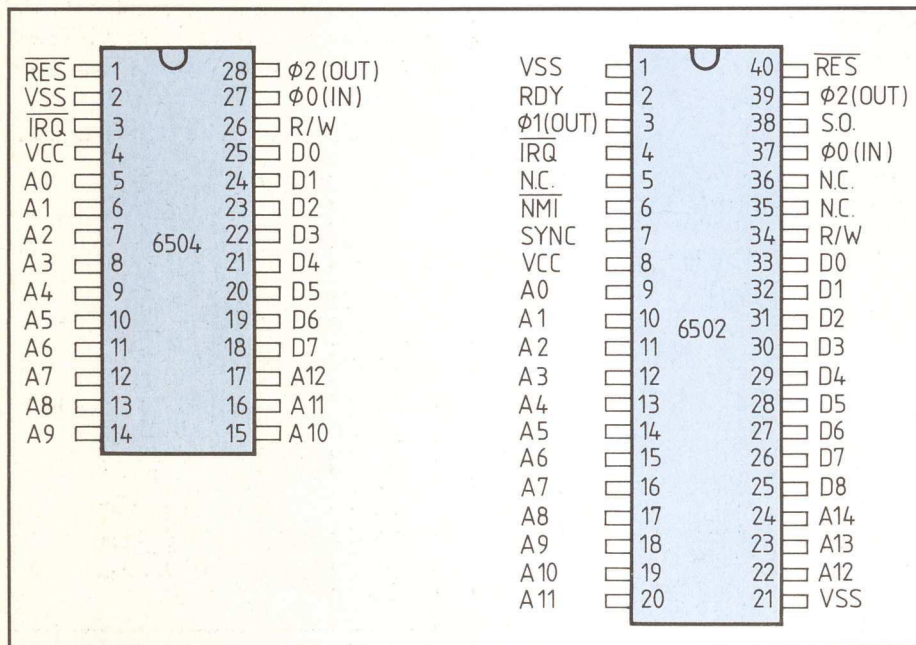


Bild 1. Anschlußbelegung der 28-Pin-CPU 6504, die im EMUF verwendet wird, im Vergleich zur software-kompatiblen CPU 6502, die man in vielen Tischcomputern findet

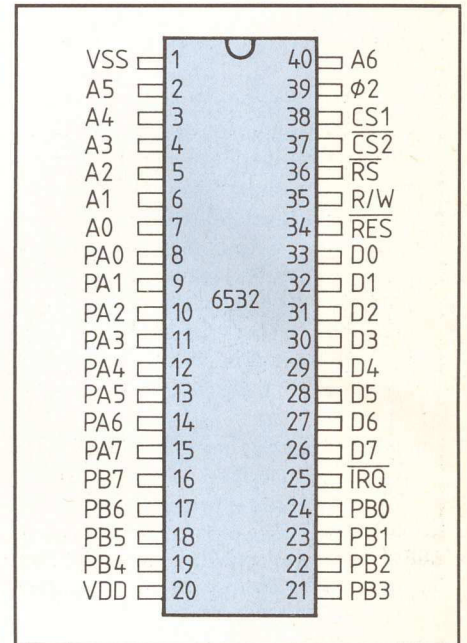


Bild 2. Anschlußbelegung des RIOT-Bausteins 6532 (RAM, I/O, Timer). Er nimmt typisch 100 mA bei 5 V Versorgungsspannung auf

mercial“-Version im Kunststoffgehäuse ist für einen Temperaturbereich von 0 bis 70° C spezifiziert, die Versorgungsspannung darf 4,75...5,25 V betragen.

RAM, I/O und Timer in einem

Der Multifunktionsbaustein 6532 enthält auf einem gemeinsamen Chip 128 × 8 Bit RAM, zwei bidirektional programmierbare 8-Bit-I/O-Ports und einen programmierbaren Intervall-Timer, der auch Interrupts auslösen kann. Bild 2 zeigt sein Anschlußbild; natürlich hat hier ein kleines 28-Pin-Gehäuse nicht mehr ausgereicht, es sind 40 Pins erforderlich. In Bild 3 ist der interne Aufbau dargestellt. Die Chipwahl erfolgt über CS1 und CS2. Der RS-Eingang (Register Select) dient zur wahlweisen Anwahl von RAM oder der I/O- und Timer-Register. Ein RES-Eingang (Reset) erlaubt es, nach dem Einschalten des Systems alle Register mit Null zu laden; die I/O-Ports sind dann als Eingänge programmiert und liegen über interne Pull-Up-Widerstände auf High-Pegel.

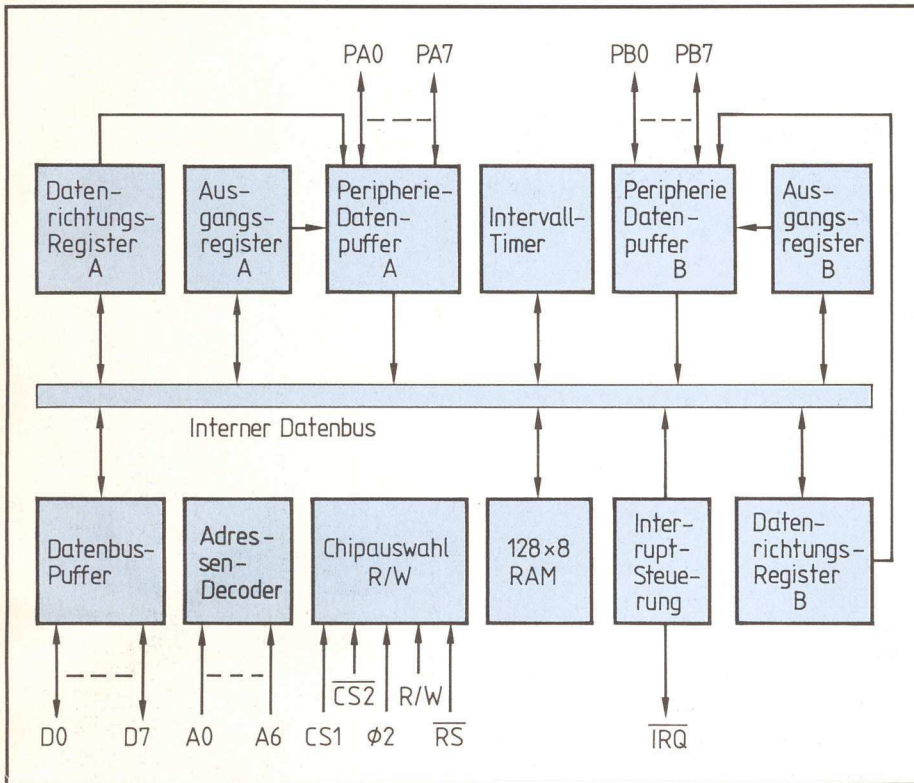


Bild 3. Internes Blockschaltbild des RIOT-Bausteins 6532. Die Anschlüsse des Port PB können bis 3 mA liefern, da sie über Gegentaktansgänge verfügen

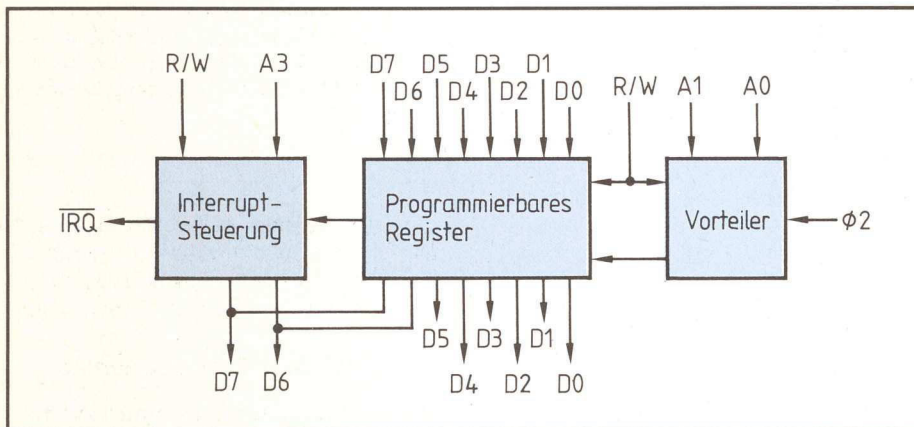


Bild 4. Struktur des programmierbaren Intervall-Timers im 6532. Der Vorteiler teilt den 1-MHz-Systemtakt je nach Programmierung durch 1, 8, 64 oder 1024. Das Timerregister kann mit einem 8-Bit-Wert geladen werden, von dem aus abwärts gezählt wird

Tabelle: Wahl der PA7-Interrupt-Betriebsart

| EMUF-Befehl | A1 | A0 | Wirkung |
|-------------|----|----|---|
| STA 0806 | 1 | 0 | Interrupt bei negativer PA7-Flanke |
| STA 0807 | 1 | 1 | Interrupt bei positiver PA7-Flanke |
| STA 0804 | 0 | 0 | PA7-Interrupt abschalten |
| BIT 0807 | 1 | 1 | Interrupt-Flag abfragen (Bit 6 = PA7, Bit 7 = Intervall-Timer) und gleichzeitig löschen |

Der Baustein läßt neben Timer-Interrupts auch flankengesteuerte Interrupts zu, die durch PA7 ausgelöst werden; diese Betriebsart läßt sich wählen, in dem man \overline{RS} und A2 auf 1 legt und R/W sowie A4 auf Null (d. h. Schreiben an die Adressen 0804...0807). Welche Daten dabei auf dem Datenbus liegen, ist belanglos. Die Tabelle nennt die vier möglichen Befehle (statt STA kann auch ein anderer Abspeicherungsbehl verwendet werden).

Die Programmierung des Intervall-Timers ist an anderer Stelle dieses Heftes ausführlich beschrieben, so daß hier nicht darauf eingegangen werden braucht (Bild 4).

Die beiden Ports PA und PB unterscheiden sich in der Ausgangsbelastbarkeit. Während PA nur über Pull-UP-Widerstände verfügt, besitzt PB Gegentakt-Ausgänge mit bis zu 3 mA Belastbarkeit. Als Eingang geschaltet, ist PB aber ebenso hochohmig wie PA.

Mehrere EPROM-Typen zur Auswahl

Der EMUF kann wahlweise mit EPROMs der Typen 2708 (drei Betriebsspannungen, 1 KByte), 2758 (eine Betriebsspannung, 1 KByte) und 2716 (eine Betriebsspannung, 2×1 KByte bestückt werden. Bild 5 zeigt die Anschlußbelegung des Typs 2708. Er benötigt die Spannungen 12 V (V_{DD}), 5 V (V_{CC}) und -5 V (V_{BB}). V_{SS} liegt auf Massepotential.

Der Typ 2758 unterscheidet sich vom 2708 nur dadurch, daß er nur +5 V als Versorgungsspannung benötigt. Er ist allerdings sehr wenig verbreitet und nicht einmal preiswerter als der ebenfalls nur mit 5 V betreibbare 2-KByte-Typ 2716.

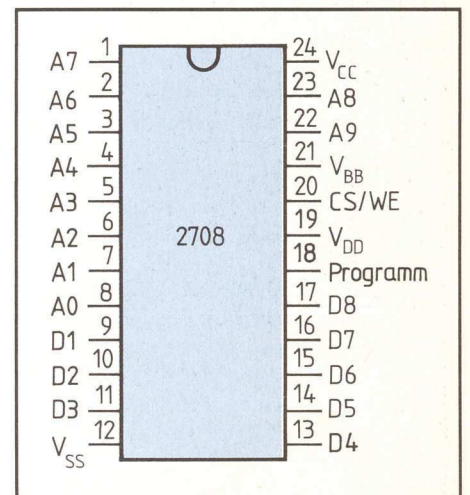


Bild 5. Das EPROM 2708, lange Zeit ein Standard-Chip, braucht noch drei Versorgungsspannungen

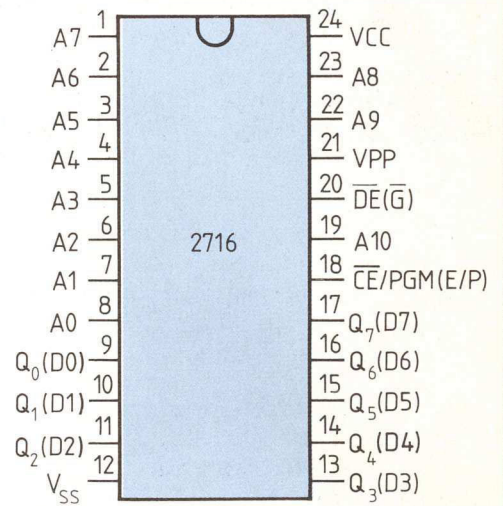
Bild 6 gibt das Anschlußbild des EPROMs 2716 wieder. Es sei darauf hingewiesen, daß Texas Instruments eine 2716-Version fertigt, die drei Versorgungsspannungen benötigt und zu den „normalen“ 2716-Versionen nicht kompatibel ist, so daß sie nicht im EMUF verwendet werden kann.

Da für 2 KByte eine Adressenleitung mehr benötigt wird als für 1 KByte, befindet sich die zusätzliche Leitung A10 am jetzt überflüssigen 12-V-Anschluß des 2708. Am 12-V-Anschluß der EMUF-Steckerleiste kann daher durch Anlegen von Masse oder + 5 V die „untere“ oder „obere“ Hälfte des 2716 ausgewählt werden, wenn in den beiden KByte zwei unterschiedliche Programme stehen. Unter der Bezeichnung 27C16 ist der Typ 2716 auch in stromsparender CMOS-Technologie zu haben (z. B. von National Semiconductor). Zusammen

mit der neuen CMOS-CPU 65C04 kann man also auch einen stromsparenden

EMUF für Batteriebetrieb aufbauen, ohne die Platine ändern zu müssen. Fe.

Bild 6. Beim 2-KByte-EPROM 2716 sitzt an Pin 19 die höchstwertige Adressenleitung A10 statt – wie beim 2708 – die 12-V-Versorgungsspannung



AIM-65 emuliert EMUF

Die hier vorgestellte Schaltung erlaubt es, den EMUF mit einem AIM-65 (4-KByte-Version) zu emulieren. Damit stehen sämtliche Programmierhilfen des AIM-65 (z. B. Editor, Assembler, Breakpoints usw.) auch für den EMUF zur Verfügung.

Die in [1] vorgeschlagene Methode der Programmentwicklung für den EMUF auf einem anderen Rechner (z. B. CBM, AIM-65, KIM) hat den Nachteil, daß möglichst ein „freier“ RIOT-Baustein 6532 zur Verfügung stehen sollte, was bei keinem der genannten Mikrocomputer der Fall ist.

Mit der hier vorgestellten Schaltung eines „In-Circuit-Emulators“ (Bild) wird dieser Nachteil beseitigt. Der Interface-Baustein erscheint im Adressbereich 0800...081F des AIM-65. Sein RAM-Bereich wird dabei abgeschaltet. Für die Programmentwicklung wird der entsprechende RAM-Teil des AIM-65 verwendet.

Die Adressen des EPROMs (0C00...0FFF) liegen im oberen Viertel des AIM-RAMs, so daß mit „echten“ Adressen gearbeitet werden kann. Hierdurch entfällt das umständliche Ändern der Adressen vor der Übertragung des Programms in das EPROM.

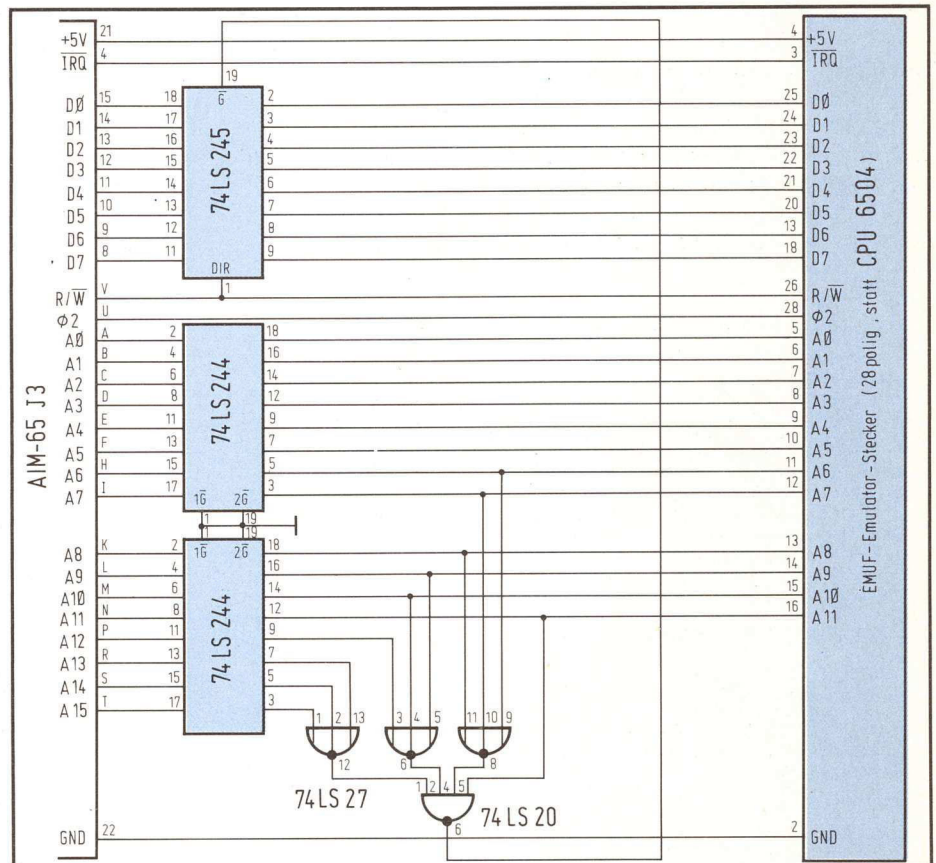
Wird der Interrupt verwendet, so muß die IRQ-Adresse des AIM in A400 auf den Anfang der Anwenderoutine umgestellt werden.

Die Schaltung des Emulators ist sehr einfach. Sie besteht aus den Adressen- und Datenpuffern sowie der Adressendecodierung, die verhindert, daß der Datenpuffer außerhalb des Adressbereiches 0800...083F freigegeben wird. Als Emulatorstecker kann eine handelsübliche 28polige Fassung verwendet werden. Er

ersetzt dann während der Programmentwicklung die CPU 6504. Günter Egle

Literatur

- [1] Feichtinger, Herwig: EMUF-Programmier-tips. mc 2/81 und EMUF-Sonderheft.
- [2] Feichtinger, Herwig: Mädchen für alles. mc 2/81 und EMUF-Sonderheft.



Schaltung eines In-Circuit-Emulators für den EMUF. Sein 28poliger Stecker ersetzt in der Entwicklungsphase die EMUF-CPU 6504

Herwig Feichtinger

So entsteht eine EMUF-Applikation

Am Beispiel des in mc 2 vorgestellten EMUF (Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung) wird hier gezeigt, wie eine Kostenabschätzung für eine bestimmte Anwendung möglich ist.

Einplatinen-Mikrocomputer wie der EMUF dienen in erster Linie als Ersatz für umfangreiche Digitalschaltungen, z. B. bei Interface-Problemen. Ein erster Schritt wird also sein, sich zu überlegen, wie groß der Aufwand der Computer-Problemlösung gegenüber einer herkömmlichen TTL- oder CMOS-Lösung ist.

Dabei muß unbedingt bedacht werden, daß bei Verwendung des EMUF erhebliche Hardware-Entwicklungskosten eingespart werden, da es sich hier um eine Standard-Platine handelt, deren Funktion ausschließlich von der Software im EPROM bestimmt wird.

Um die ungefähren Kosten einer Problemlösung mit einem Einplatinen-Computer abschätzen zu können, gibt Bild 1 ein kleines Basic-Programm wieder, das die Herstellkosten pro Stück errechnet. Als Eingangsgrößen werden dabei die Hardware-Kosten (EMUF: ca. 120 DM), die Kosten für Aufbau und

Funktionstest, die Software-Entwicklungskosten für dieses Gerät und seine voraussichtlich absetzbare Stückzahl abgefragt. Sollte noch kein geeignetes Entwicklungssystem vorhanden sein, so werden seine Kosten auf die Gesamtzahl aller voraussichtlich absetzbaren Geräte auf der Basis der im EMUF vorhandenen Prozessorfamilie 650X umgelegt.

Wie schon in Heft 2 erwähnt, ist einer der Hauptvorteile des EMUF, daß kein teures Entwicklungssystem benötigt wird; Geräte wie AIM-65 (baugleich mit dem PC-100) oder KIM-1 sind ideal geeignet, aber auch Apple-II, PET oder CBM eignen sich im Prinzip für diesen Zweck. Zum Preis des Computers sind noch die Kosten für eine EPROM-Programmiermöglichkeit zu addieren (beim AIM-65 und PC-100 ist dazu nur eine Diode und ein EPROM-Sockel nötig). Aus dem Probelauf des Basic-Programms geht u. a. recht deutlich hervor, daß die Hardware-Kosten immerhin zu

mehr als 50 % der gesamten Herstellkosten eingehen. Es ist also durchaus sinnvoll, für einfache Steueraufgaben eine wirkliche Minimal-Konfiguration wie den EMUF einzusetzen, da eine spätere Erweiterbarkeit hier nicht gefordert wird.

Als Nachtrag zu dem ebenfalls in Heft 2 erschienenen Beitrag „EMUF-Programmiertips“ zeigt Bild 2 schließlich eine Gegenüberstellung der Systemadressen in den Computern EMUF, AIM-65 (PC-100), CBM-3001, PET-2001 und KIM-1. Nach dem Programmtest auf dem Entwicklungssystem genügt es, nur diese Adressen zu ändern, bevor die Software in ein EPROM für den EMUF gebrannt wird.

Möglicherweise haben Sie eine EMUF-Platine erhalten, die nicht ganz mit der in Heft 2 abgebildeten identisch ist: Bedauerlicherweise erhielt die Fa. Elektronikladen (Detmold) vom Franzis-Labor ein fehlerhaftes Layout der EMUF-Platine. Die Leitung an Pin 1 des 555 ist aufzutrennen und Pin 1 an Masse zu legen. Ferner ist es günstig, Pin 18 des EPROM an Masse zu legen, um definierte Verhältnisse zu erzielen.

```

10 REM EMUF-SERIENKALKULATION
20 INPUT"HARDWARE-KOSTEN";HK
30 INPUT"KOSTEN F.BAU U.TEST";K
40 K=K+HK*PRINT"IST EIN ENTWICKLUNGS-"
50 INPUT"SYSTEM VORHANDEN (J/N)";A$
60 ES=0:IF A$="J"GOTO 110
70 INPUT"ENTW.-SYST.-KOSTEN";ES
80 PRINT"WIEVIELE GERAETE AUF 650X-"
90 PRINT"BASIS WERDEN VORAUSS. INSGES."
100 INPUT"VERKAUFT";N:ES=ES/N
110 K=K+ES:INPUT"SOFTW.-ENTW.-KOSTEN";S
120 INPUT"GEPL.STUECKZAHL F.D.PROJEKT";N
130 S=S/N:PRINT"HERSTELLKOSTEN PRO "
140 PRINT"STUECK ="$INT(K+S+.5)$"DM"

RUN
HARDWARE-KOSTEN? 120
KOSTEN F.BAU U.TEST? 60
IST EIN ENTWICKLUNGS-
SYSTEM VORHANDEN (J/N)? N
ENTW.-SYST.-KOSTEN? 2800
WIEVIELE GERAETE AUF 650X-
BASIS WERDEN VORAUSS. INSGES.
VERKAUFT? 1500
SOFTW.-ENTW.-KOSTEN? 1200
GEPL.STUECKZAHL F.D.PROJEKT? 500
HERSTELLKOSTEN PRO STUECK = 184 DM
    
```

Bild 1. Basic-Programm zur groben Abschätzung der Stückkosten für eine EMUF-Anwendung. Die im Probelauf angegebenen Stückzahlen sind fast untypisch hoch; bei geringeren Zahlen gehen die Kosten des Entwicklungssystems sowie die Software-Entwicklungskosten deutlich mehr ein

Bild 2. Gegenüberstellung der Systemadressen im Einplatinen-Mikrocomputer EMUF und einigen typischen Entwicklungssystemen. Leider ist bei PET und CBM kein 6532-Baustein und damit kein äquivalenter Timer vorhanden; ferner ist der zweite 8-Bit-I/O-Port nicht ausnützlich

| | EMUF | AIM- | CBM- | PET- | KIM- | |
|------|------|------|------|------|------|-----------------|
| | 6504 | 65 | 3001 | 2001 | 1 | |
| PA | 0800 | A00F | E84F | E84F | 1700 | PORT A |
| PAD | 0801 | A003 | E843 | E843 | 1701 | PORT-A-RICHTUNG |
| PB | 0802 | A000 | ---- | ---- | 1702 | PORT B |
| PBD | 0803 | A002 | ---- | ---- | 1702 | PORT-B-RICHTUNG |
| T1 | 0814 | A474 | ---- | ---- | 1704 | TIMER 1 USEC |
| T8 | 0815 | A495 | ---- | ---- | 1705 | TIMER 8 USEC |
| T64 | 0816 | A496 | ---- | ---- | 1706 | TIMER 64 USEC |
| TK | 0817 | A497 | ---- | ---- | 1707 | TIMER 1024 USEC |
| TI1 | 081C | A49C | ---- | ---- | 170C | T1 M.INTERR. |
| TI8 | 081D | A49D | ---- | ---- | 170D | T8 M.INTERR. |
| TI64 | 081E | A49E | ---- | ---- | 170E | T64 M.INTERR. |
| TIK | 081F | A49F | ---- | ---- | 170F | TK M.INTERR. |
| IRQL | 0FFE | A404 | 0090 | 0219 | 17FE | IRQ-VEKTOR L |
| IRQH | 0FFF | A405 | 0091 | 021A | 17FF | IRQ-VEKTOR H |

Herwig Feichtinger

EMUF-Programmierertips

Wer ist „EMUF“? Nun, der in diesem Heft beschriebene Einplatinen-Mikrocomputer wurde unter dieser redaktionsinternen Bezeichnung ursprünglich für einen ganz anderen Verwendungszweck entwickelt, nämlich zur Steuerung eines Kurzwellenempfängers, eines „Empfängers mit unzulässigen Frequenzbereichen“. Inzwischen ist es allerdings besser, EMUF als „Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung“ zu interpretieren. Hier nun einige Tips, wie man Programme für den EMUF entwickelt.

Der Einplatinen-Mikrocomputer „EMUF“ arbeitet mit der CPU 6504, die exakt über den gleichen Befehlssatz wie ihre größere Schwester 6502 verfügt. Dies ermöglicht die Programmentwicklung auf praktisch allen 6502-Tischcomputern wie PET, CBM, PC-100, Apple II usw.

Erster Schritt: I/O-Festlegung

Bevor das EMUF-Maschinenprogramm geschrieben wird, muß man sich zunächst über die Hardware-Voraussetzungen klar werden. Üblicherweise werden an den EMUF verschiedene Ein- und Ausgänge, Schalter, Leuchtdioden-Treibertransistoren usw. angeschlossen. Dafür stehen zwei Ports (PA und PB) mit je acht Leitungen zur Verfügung. Für jeden Port enthält der Baustein 6532 ein Datenrichtungsregister (PAD bzw. PBD), das es erlaubt, einzelne Leitungen (also Bits) eines Ports wahlweise als Ein- oder Ausgang zu deklarieren (0 = Eingang, 1 = Ausgang). Nach einem Reset und damit nach dem Einschalten des EMUF sind alle Ports zunächst als Eingang geschaltet.

Nicht beschaltete Eingänge liegen (wie offene TTL-Gatter-Eingänge) auf log. 1; legt man an eine Portleitung also einen Schalter nach Masse, so ist ein Pull-Up-Widerstand nicht unbedingt nötig, wenn auch empfehlenswert, um die Empfindlichkeit gegenüber Störeinflüssen zu reduzieren. Bild 1 zeigt, wie man an drei

Leitungen des Ports PA einen Schalter und zwei Leuchtdioden wahlweise über einen Transistor oder einen TTL-Inverter (ev. mit offenem Kollektor, also 7404 oder 7406) als Treiber anschließt, nachdem der Port selbst nicht genügend Strom für eine LED zur Verfügung stellen kann.

Sobald jedem Ein- und Ausgang der Anwenderschaltung eine Portleitung zugeordnet ist, kann man sich der Software-Erstellung zuwenden.

Zweiter Schritt: Programmtest mit Tischrechner

Für die Bewältigung der gestellten Aufgabe sollte man das Programm nicht aufs Geratewohl schreiben und sofort in ein EPROM für den EMUF schießen. Viel-

mehr ist es sinnvoll, es zunächst auf dem Tischrechner zu testen, der auch der Programmentwicklung dient, und eventuell zu korrigieren.

Dies stellt an den Tischrechner aber bestimmte Anforderungen. Er sollte möglichst über die gleichen Ein- und Ausgabemöglichkeiten verfügen, die man später beim EMUF benutzen möchte, und auch einen Timer gleicher Struktur wie im 6532 des EMUF besitzen. Die Computer AIM-65 (Rockwell) und PC-100 (Siemens) sind hier ideal geeignet, da auch sie zwei User-Ports und einen 6532-Timer enthalten. Ferner ist bei ihnen der gesamte benutzbare Adressenbereich des EMUF im Arbeitsspeicher als RAM vorhanden, so daß z. B. das später im EPROM stehende EMUF-Programm sofort im endgültigen Adressenbereich 0C00...0FFF entwickelt werden kann. An Änderungen vor dem Programmieren eines EPROM ist dann lediglich noch die Anpassung der Timer- und Portadressen nötig.

Kommt man mit nur einem Port aus und verzichtet auch auf die Möglichkeit, den 6532-Timer noch im Entwicklungssystem zu testen, ist das Erstellen der Software auch mit Rechnern wie PET und CBM möglich. Hierbei kann es auch nützlich sein, den 6532-Timer während der Entwicklungsphase durch den VIA-Timer im 6522 des CBM zu simulieren, was aber eine größere Softwareänderung erforderlich macht.

Gleich sind wir fertig...

Hat man das Maschinenprogramm auf dem Entwicklungssystem (AIM, PC-100, CBM usw.) zum Laufen gebracht, so muß man vor dem Programmieren eines EPROM noch beachten, daß manche Dinge, die sonst das Monitorprogramm des Systems erledigt, für den EMUF explizit programmiert werden müssen. Dies gilt speziell für die Befehlsfolge, die nach einem Reset nötig ist, um die CPU in einen definierten Zustand zu bringen. Normalerweise sieht der Programmumfang deshalb etwa so aus wie in Bild 2: Der Stackpointer wird auf den Anfangs-

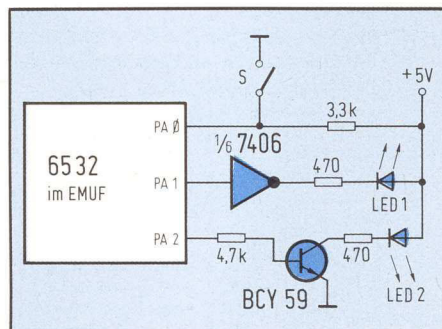


Bild 1. Will man einen Schalter an eine I/O-Leitung anschließen, empfiehlt sich ein Pull-Up-Widerstand (hier 3,3 k Ω). Leuchtdioden können über eine Transistor- oder TTL-Treiberschaltung angesteuert werden

wert FF gesetzt, das Dezimal-Flag wird gelöscht (dies ist normalerweise sinnvoll, um die hexadezimale Arbeitsweise bei ADC- und SBC-Befehlen sicherzustellen). Dann kann man die Port-Datenrichtungsregister initialisieren, um bestimmte Portleitungen als Ein- bzw. Ausgang zu deklarieren.

Wenn schließlich noch alle Port- und Timer-Adressen an den EMUF angepaßt sind und das Programm auch im EPROM-Bereich ab 0C00 lauffähig ist, kann man ein EPROM programmieren und auf die EMUF-Platine stecken.

Sehr nützlich: ein Assembler

Das Umschreiben des mühsam entwickelten Programms vom Entwicklungssystem auf die veränderte Adressenlage des EMUF von Hand wäre unvermeidbar zeitraubend und fehlerträchtig, speziell bei Programmen von mehr als einem halben KByte. Es ist deshalb sinnvoll, für die Programmerstellung einen symbolischen Assembler zu verwenden, dem man die Adressenlage nur einmal am Anfang des Quelltext-Programms mitzuteilen braucht und der bei der Umsetzung in den Objektcode, wie er ins EPROM muß, alle nötigen Anpassungen selbständig vornimmt. Im Quelltext selbst steht nämlich z. B. für den Port A nicht eine bestimmte hexadezimale Adresse, sondern nur der symbolische Name PA, und überall, wo der Assembler auf PA trifft, setzt er dafür jene Adresse ein, die der Programmierer am Anfang dafür einmal definiert hat. Für solche Assembler-Listings werden Sie in mc noch mehrfach typische Beispiele finden.

Zwei Programme im EPROM

Da der EMUF immer nur ein Kilobyte im EPROM-Bereich 0C00..0FFF adressiert, was genau dem Speicherplatz eines EPROM 2708 bzw. 2758 entspricht und für die meisten Anwendungsfälle auch vollkommen ausreicht, ergibt sich die Möglichkeit, bei Verwendung eines 2-KByte-EPROM (2716) an dessen höchstwertige (hier unbeschaltete) Adressenleitung einen Schalter zu legen.

Damit ist es möglich, im „unteren“ und „oberen“ Kilobyte zwei voneinander unabhängige Programme von je maximal 1 KByte Länge unterzubringen und mit einem Schalter auszuwählen, welche Aufgabe der EMUF gerade zu erfüllen hat. Nach der Schalterbetätigung ist na-

Bild 2. Typisches Assemblerlisting zur Initialisierung des EMUF nach einem Reset. Die CPU holt sich die Reset-Adresse aus 0FFC und 0FFD. Dann wird der Stackpointer auf FF gesetzt, die Port-Datenrichtungsregister werden mit dem jeweiligen Bitmuster geladen, und das Dezimal-Flag wird gelöscht. Das Listing entstand mit dem 2-Pass-Assembler von AIM-65 und PC-100

```

0C0E      PASS 2
0000
0000      ;EMUF-INITIALISIERUNG
0000 PA      = $B00
0000 PAD     = $B01
0000 PB      = $B02
0000 PBD     = $B03
0000      *=$FFC      ;RESET
0FFC      0000      .WOR RESET ;VECTOR
0FFE      *=$C00
0C00 RESET A2FF      LDX #$FF      ;STACK-
0C02      9A        TXS              ;POINTER
0C03      A9FB      LDA #Z1111000
0C05      B00100    STA PAD          ;PORT-
0C08      A93E      LDA #Z0011110
0C0A      B00300    STA PBD         ;RICHTUNG
0C0D      08        CLD
0C0E      ;WEITER IM PROGRAMM
0C0E      .END
    
```

türlich stets ein Reset (Betriebsspannungsunterbrechung oder Reset-Taste) notwendig.

Die Programmentwicklung hat für beide EPROM-Hälften so zu geschehen, daß die Adressenbelegung stets dem Bereich 0C00..0FFF oder einem seiner Speicherbereichs-Duplikate (z. B. 6C00..6FFF) entspricht. Wenn im Entwicklungssystem genügend Speicherplatz zur Verfügung steht, kann man deshalb das „unte-

re“ Programm von 0C00..0FFF und das „obere“ von 1C00..1FFF entwickeln, ohne daß im EMUF später Schwierigkeiten auftreten.

Wie man das Programm dann mit AIM-65 oder PC-100 in ein EPROM schießt, ist sehr ausführlich in dem von Siemens erhältlichen „Applikationsbuch PC-100“ beschrieben; an Hardware ist dafür nur eine Diode und eine EPROM-Fassung erforderlich.

Antiquarisches

Die Mathematiker, die nichts als Mathematiker sind, haben einen klaren Verstand, vorausgesetzt, daß man ihnen alles durch Definitionen und Prinzipien erklärt, sonst sind sie wirr und unerträglich, denn sie denken nur richtig an Hand deutlich gemachter Prinzipien.

Und die Feinsinnigen, die nichts als feinsinnig sind, sind unfähig, die Geduld aufzubringen, bis zu den ersten Prinzipien der Spekulation und Abstraktion vorzudringen, denen sie in der Welt niemals begegnet sind und die man dort nie braucht.

Blaise Pascal



Rudolf Hofer

Standardschnittstellen für Schreibmaschinen-Drucker

Das in FUNKSCHAU 1980, Heft 4, vorgestellte Interface, das eine elektrische Schreibmaschine zum Drucker erweitert, hat keine standardisierte Schnittstelle. Das ist kein wesentlicher Nachteil, wenn man es an einen Computer anschließen will, bei dem ein eigenes Treiberprogramm pro-

blemlos eingefügt werden kann. Ist dies nicht der Fall oder bereitet die Entwicklung der Software Schwierigkeiten, dann kann man das Problem mit Hilfe des in Heft 2 beschriebenen „EMUF“-Minimalcomputers lösen, der eine serielle oder Parallel-Schnittstelle realisiert.

```

0800      1  ;CENTRONICS/SCHREIBMASCHINE
0800      2  ;UND V24/SCHREIBMASCHINE
0800      3  ;AUF EMUF
0800      4  ;=====
0800      5  ;PB0...6=DATENEING. F. CENTRONICS
0800      6  ;PA6=STROBE F. CENTRONICS
0800      7  ;PB7=BUSY FUER CENTR.
0800      8  ;=====
0800      9  ;PB1/2 GESCHW. FUER V24
0800     10  ; 0 0 = 110 BD
0800     11  ; 1 0 = 300 BD
0800     12  ; 0 1 = 600 BD
0800     13  ; 1 1 = 1200 BD
0800     14  ;PB6=EMPFANGSBEREIT (0)
0800     15  ;PB7=DATENEINGANG
0800     16  ;=====
0800     17  ;PA0...5=DATEN F. SCHREIBM.
0800     18  ;PA7=STROBE F. SCHREIBMASCHINE
0800     19  ;*****
0800     20  ;
0800     21  ;RESET-VEKTOR 6FFC:00
0800     22  ;                      6FFD:6C
0800     23  ;
0800     24  ;*****
0800     25  PA      EQU $800
0800     26  PB      EQU $802
0800     27  DRA     EQU $801
0800     28  DRB     EQU $803
0800     29  WTIMER  EQU $814
0800     30  RTIMER  EQU $815
0800     31  XTEMP   EPZ 1
0800     32  FLG     EPZ 0
0800     33  ZEIT1   EPZ 2
0800     34  ZEIT2   EPZ 3
0800     35  ;*****
6C00     36  ORG $6C00
6C00     37  ;INITIALISIERUNG
6C00 A2FF  38  INIT   LDX #$FF
6C02 9A   39  TXS
6C03 DB   40  CLD
6C04 78   41  SEI
    
```

Bild 1. Listing der beiden Schnittstellen-Programme, die sich lediglich ab Zeile 110 unterscheiden. Anschluß 29 der EMUF-Steckerleiste entscheidet, welches aktiv ist

Die Mehrzahl der heute auf dem Markt befindlichen Drucker ist entweder mit Centronics- oder V.24-Schnittstelle ausgerüstet. Demzufolge wird auch für die meisten Computer zumindest eine der beiden angeboten. Die in Bild 1 abgedruckten Programme versetzen den EMUF in die Lage, sowohl V.24- als auch Centronics-Schnittstellen direkt mit dem Schreibmaschinen-Interface zu verbinden – eine typische Aufgabe für den EMUF. Als Programmspeicher wird ein EPROM des Typs 2716 verwendet. Das hat einerseits den Vorteil, daß man nur eine Versorgungsspannung von 5 V braucht, andererseits bietet es doppelt so viel Speicherraum, wie der EMUF adressiert. Mit der Adreßleitung A10, die am Steckeranschluß 29 herausgeführt ist, kann man nun wählen, welche Hälfte des EPROMs aktiv sein soll. Damit erledigt der Mikrocomputer völlig verschiedene Aufgaben, je nachdem, ob A10 an Masse oder 5 V liegt.

Anpassung an eine serielle Schnittstelle

In Bild 2 ist die untere Hälfte des EPROMs aktiv. an PB1 und PB2 lassen sich vier verschiedene Geschwindigkeiten einstellen, je nachdem ob man die Eingänge an 5 V oder Masse legt. Die Datenleitung der seriellen Schnittstelle wird mit PB7 verbunden, vorausgesetzt,

| | | | | | |
|------|--------|-----|-------|------------|---------------------|
| 6C05 | A9BF | 42 | LDA | #%10111111 | |
| 6C07 | 8D0108 | 43 | STA | DRA | ;NUR PA6=EING. |
| 6C0A | A980 | 44 | LDA | #%10000000 | |
| 6C0C | 8D0308 | 45 | STA | DRB | ;PB7=AUSG. (BUSY) |
| 6C0F | 20E56C | 46 | LOOP | JSR | ZEIN |
| 6C12 | 20186C | 47 | | JSR | ZAUS |
| 6C15 | 18 | 48 | | CLC | |
| 6C16 | 90F7 | 49 | | BCC | LOOP |
| 6C18 | | 50 | | | ;***** |
| 6C18 | | 51 | | | ;ZEICHENAUSGABE |
| 6C18 | 8601 | 52 | ZAUS | STX | XTEMP |
| 6C1A | AA | 53 | | TAX | |
| 6C1B | BD656C | 54 | LDA | TAB,X | ;TABELLENWERT |
| 6C1E | 48 | 55 | | PHA | |
| 6C1F | 2940 | 56 | AND | ##%40 | ;SHIFT ISOLIEREN |
| 6C21 | C500 | 57 | CMP | FLG | |
| 6C23 | F00F | 58 | BEQ | SAME | |
| 6C25 | A600 | 59 | LDX | FLG | |
| 6C27 | 8500 | 60 | STA | FLG | |
| 6C29 | F004 | 61 | BEQ | LOCK | |
| 6C2B | A936 | 62 | LDA | ##%36 | ;SHIFT CODE |
| 6C2D | D002 | 63 | BNE | SHIFT | |
| 6C2F | A933 | 64 | LOCK | LDA | ##%33 |
| 6C31 | 20576C | 65 | SHIFT | JSR | KEY3 |
| 6C34 | 68 | 66 | SAME | PLA | |
| 6C35 | | 67 | | | ;PRESS 1 KEY |
| 6C35 | A205 | 68 | KEY | LDX | ##%5 |
| 6C37 | 48 | 69 | | PHA | ;DURATION |
| 6C38 | 293F | 70 | | AND | ##%3F |
| 6C3A | 20466C | 71 | | JSR | KEY1 |
| 6C3D | A209 | 72 | LDX | ##%9 | ;SHORT DELAY |
| 6C3F | 68 | 73 | RT | PLA | |
| 6C40 | 1002 | 74 | | BPL | LG |
| 6C42 | A235 | 75 | LDX | ##%35 | ;LONG DELAY |
| 6C44 | A9FF | 76 | LG | LDA | ##%FF |
| 6C46 | A000 | 77 | KEY1 | LDY | #0 |
| 6C48 | 8D0008 | 78 | KEY2 | STA | PA |
| 6C4B | 20566C | 79 | | JSR | RTN |
| 6C4E | 88 | 80 | | DEY | |
| 6C4F | D0F7 | 81 | | BNE | KEY2 |
| 6C51 | CA | 82 | | DEX | |
| 6C52 | D0F4 | 83 | | BNE | KEY2 |
| 6C54 | A601 | 84 | | LDX | XTEMP |
| 6C56 | 60 | 85 | RTN | RTS | |
| 6C57 | | 86 | | | ;SCHNELLER DRUCKEN |
| 6C57 | A20F | 87 | KEY3 | LDX | ##%0F |
| 6C59 | 48 | 88 | | PHA | ;DURATION FOR SHIFT |
| 6C5A | 293F | 89 | | AND | ##%3F |
| 6C5C | 20466C | 90 | | JSR | KEY1 |
| 6C5F | A214 | 91 | | LDX | ##%14 |
| 6C61 | 4C3F6C | 92 | | JMP | RT |
| 6C64 | 00 | 93 | | BRK | |
| 6C65 | 3F3F3F | 94 | TAB | HEX | 3F3F3F3F3F3F3F3F |
| 6C68 | 3F3F3F | | | | |
| 6C6B | 3F3F | | | | |
| 6C6D | C6343F | 95 | | HEX | C6343F3F3FB53F3F |
| 6C70 | 3F3FB5 | | | | |
| 6C73 | 3F3F | | | | |
| 6C75 | 3F3F3F | 96 | | HEX | 3F3F3F3F3F3F3F3F |
| 6C78 | 3F3F3F | | | | |
| 6C7B | 3F3F | | | | |
| 6C7D | 3F3F3F | 97 | | HEX | 3F3F3F3F3F3F3F3F |
| 6C80 | 3F3F3F | | | | |
| 6C83 | 3F3F | | | | |
| 6C85 | 344370 | 98 | | HEX | 3443706C69606B44 |
| 6C88 | 6C6960 | | | | |
| 6C8B | 6B44 | | | | |
| 6C8D | 586A24 | 99 | | HEX | 586A24080C040348 |
| 6C90 | 080C04 | | | | |
| 6C93 | 0348 | | | | |
| 6C95 | 093130 | 100 | | HEX | 0931302C202B182A |
| 6C98 | 2C202B | | | | |
| 6C9B | 182A | | | | |
| 6C9D | 102968 | 101 | | HEX | 102968710A6C1D4C |
| 6CA0 | 710A6C | | | | |
| 6CA3 | 1D4C | | | | |
| 6CA5 | 2E6E53 | 102 | | HEX | 2E6E535B65615A5D |
| 6CAB | 5B6561 | | | | |
| 6CAD | 5A5D | | | | |
| 6CAD | 525655 | 103 | | HEX | 5256554A4D4B5449 |
| 6CBO | 4A4D4B | | | | |

der Computer liefert die Daten im TTL-Pegel. Liegt ein V.24-Signal (üblicherweise ± 12 V) vor, dann muß ein Pegelumsetzer (MC1489, SN75189) dazwischengeschaltet werden. Die Leitung \overline{EB} (empfangsbereit) dient dazu, dem Computer mitzuteilen, ob er weitere Daten senden darf. Ist sie auf H, können keine Zeichen empfangen werden, da der „Drucker“ gerade beschäftigt ist. Auch ihr Pegel muß angepaßt werden, wenn sie mit einer V.24-Schnittstelle verbunden ist (MC1488, SN75188). In manchen Fällen schickt der Computer einfach Zeichen für Zeichen, ohne sich darum zu kümmern, ob das empfangende Gerät schnell genug ist. Er fragt also \overline{EB} nicht ab. Hier wäre es denkbar, mit \overline{EB} einen Interrupt auszulösen und das Interruptprogramm die Empfangsbereitschaft abfragen zu lassen.

Auch Anschluß an Centronics-Schnittstelle möglich

Bild 3 zeigt, wie der EMUF Centronics-Schnittstelle und Schreibmaschinen-Interface miteinander verbindet. Das Programm – ausgewählt durch A10 des EPROMs (Steckeranschl. 29 auf Masse) – ist bis auf den Eingabeteil ab Zeile 110 identisch mit dem V.24-Programm. Pegelprobleme gibt es hier keine. Allerdings muß man das Strobe-Signal, das vom Computer kommt, unter Umständen mit einem Monoflop auf einige Mikrosekunden verlängern. Ein unbenutztes Monoflop findet sich im Schreibmaschinen-Interface.

Die Ansteuerung des Magnet-Interface

Das Unterprogramm zur Ausgabe eines Zeichens wandelt den ASCII-Code des Computers mit Hilfe einer Tabelle in den speziellen Drucker-Code um. Diese Tabelle beginnt bei Adresse 6C65 und gilt für die meisten kleineren Schreibmaschinen (z.B. Gabriele 2000). Unterscheidet sich die Tastenbelegung nur geringfügig, dann ist es am einfachsten, man lötet die falsch betätigten Magneten um. Sind sehr viele Änderungen nötig, dann erstellt man am besten eine neue Tabelle. Dazu muß man folgendes wissen: Die Position des Tabellenwertes relativ zum Tabellenbeginn repräsentiert das ASCII-Zeichen. Der Tabellenwert selbst ist der Druckercode. An der Stelle 6C65 steht also der Druckercode, der ausgegeben wird, wenn das ASCII-Zei-

```

6CB3 5449
6CB5 4E7266 104      HEX 4E72666259515C6D
6CB8 625951
6CB8 5C6D
6CBD 64635E 105      HEX 64635E4542344028
6CC0 454234
6CC3 4028
6CC5 342E13 106      HEX 342E131B25211A1D
6CC8 1B2521
6CC8 1A1D
6CCD 121615 107      HEX 1216150A0D0B1409
6CD0 0A0D0B
6CD3 1409
6CD5 0E3226 108      HEX 0E32262219111C2D
6CD8 221911
6CDB 1C2D
6CDD 24231E 109      HEX 24231E0502340146
6CE0 050234
6CE3 0146
6CE5 110      ; *****
6CE5 111      ; V24-EINGABE
6CE5 112      ; PB7=DATEN
6CE5 113      ; PBO=7/8 BIT (MASSE/5 V)
6CE5 114      ; PB1 U. 2=GESCHW.
6CE5 115      ; PB6=EMPFANGSBEREIT-ANZ.
6CE5 20216D 116      ZEIN   JSR   PARAM
6CE8 A900 117          LDA   #0           ; EB=0 EMPFANGSBEREIT
6CEA 8D0208 118          STA   PB
6CED 2C0208 119      EMP    BIT   PB
6CF0 30FB 120          BMI   EMP           ; WARTEN AUF L
6CF2 A209 121          LDX   #9
6CF4 A502 122          LDA   ZEIT1
6CF6 4A 123           LSR
6CF7 20176D 124          JSR   WAIT12
6CFA A900 125          LDA   #0
6CFC 48 126           NBIT   PHA
6CFD 20106D 127          JSR   WAIT
6D00 68 128           PLA
6D01 0E0208 129          ASL   PB
6D04 6A 130           ROR
6D05 CA 131           DEX
6D06 D0F4 132          BNE   NBIT
6D08 A040 133          LDY   #40           ; EB=1
6D0A 8C0208 134          STY   PB
6D0D 297F 135          AND   #7F
6D0F 60 136           RTS
6D10 137      ; *****
6D10 138      ; WARTEN AUF TIME-OUT
6D10 139      ; UND TIMER NEU LADEN
6D10 2C1508 140      WAIT   BIT   RTIMER
6D13 10FB 141          BPL   WAIT
6D15 A502 142          WAIT1  LDA   ZEIT1
6D17 8601 143          WAIT12 STX   XTEMP
6D19 A603 144          LDX   ZEIT2
6D1B 9D1408 145          STA   WTIMER, X
6D1E A601 146          LDX   XTEMP
6D20 60 147           RTS
6D21 148      ; *****
6D21 A940 149          PARAM  LDA   #40           ; PB6=AUSG.
6D23 8D0308 150          STA   DRB
6D26 AD0208 151          LDA   PB           ; GESCHW.
6D29 4A 152           LSR
6D2A 2903 153          AND   #3
6D2C AA 154           TAX
6D2D BD386D 155          LDA   GTAB, X       ; GESCHW.-TABELLE
6D30 8502 156          STA   ZEIT1
6D32 BD3C6D 157          LDA   GTAB+4, X
6D35 8503 158          STA   ZEIT2
6D37 60 159           RTS
6D38 8C34CC 160      GTAB   HEX 8C34CC62       ; 110, 300
6D3B 62
6D3C 161      ; 600 U. 1200 BAUD
6D3C 020201 162          HEX 02020101
6D3F 01
6D40 163      ; *****
6D40 164      ; INTERRUPT UND RESET
6FFC 165          ORG   #6FFC
6FFC 006CF1 166          HEX 006CF16C
6FFF 6C
7000 167      ; *****
168      END

```

chen 00 empfangen wird, an der Stelle 6C66 der Druckercode für das ASCII-Zeichen 01 usw. Wie man sieht, steht bis ASCII 1F fast immer der Wert 3F in der Tabelle. Der Grund dafür besteht darin, daß es sich hierbei fast ausschließlich um Steuerzeichen handelt, die nicht gedruckt werden (3F betätigt keinen Magneten). Welchen Druckercode man in die Tabelle einsetzt, geht aus Bild 4 hervor. Es zeigt die Positionen der einzelnen Magneten und die zugehörigen Codes. Die höhere der beiden Hexadezimalzahlen gibt jeweils den Code für das Zeichen mit betätigter Shift-Taste an. Beim Betrieb des EMUF sollte man beachten, daß an den Eingängen des 6532 möglichst keine Spannungen anliegen sollten, bevor die Versorgungsspannung anliegt. Es empfiehlt sich deshalb, den EMUF gemeinsam mit dem Computer oder nach ihm einzuschalten. Wer völlig

Antiquarisches



„Manche Leute beanstanden die Einführung von solchen Maschinen, welche die Arbeit der Menschen entlasten können, mit der Begründung, daß man dadurch den Armen ihr Brot wegnehme. Aus diesem Beweggrund verbot man vor einigen Jahren in Regensburg die Strumpf- und Bandmühlen. Abgesehen davon, daß dieses Verbot wirkungslos blieb, bin ich der Meinung, daß immer genug Gelegenheit besteht, Menschen auf andere Weise mit nützlicher Arbeit zu beschäftigen, selbst wenn diese zunächst ungewohnt ist. Denn das ist sie nur, solange die Menschen sich noch nicht daran gewöhnt haben, eine andere Arbeit zu verrichten“ [Handschr.: LBr 302].

Gottfried Wilhelm Leibniz (1646–1716)

sichergehen will, puffert Ein- und Ausgänge mit TTL- oder CMOS-Bausteinen. Übertriebene Vorsicht ist jedoch nicht angebracht. Beim Verfasser überlebten die „empfindlichen LSIs“ bisher ausnahmslos alle unbeabsichtigten Dauerbelastungsversuche.

Literatur

- [1] Hofer, R.: Schreibmaschine wird zum Schön-schreibdrucker. FUNKSCHAU 1980, Heft 4.
- [2] Feichtinger, H.: Mädchen für alles (Einplatinen-Mikrocomputer) mc 1981, Heft 1.

```

6CE5      110  ; *****
6CE5      111  ; CENTRONICS-EINGABE
6CE5 A900  112  ZEIN  LDA #0
6CE7 8D0208 113          STA PB          ; BUSY=0
6CEA 2C0008 114  STROBE BIT PA          ; STROBE=0?
6CED 70FB  115          BVS STROBE
6CEF A0B0  116          LDY ##80
6CF1 8C0208 117          STY PB          ; BUSY=1
6CF4 AD0208 118          LDA PB
6CF7 297F  119          AND ##7F
6CF9 60    120          RTS
6CFA      121  ; *****
6CFA      122  ; RESETVEKTOR
6FFC      123          ORG $6FFC
6FFC 006C  124          HEX 006C
6FFE      125  ; *****
          126          END
    
```

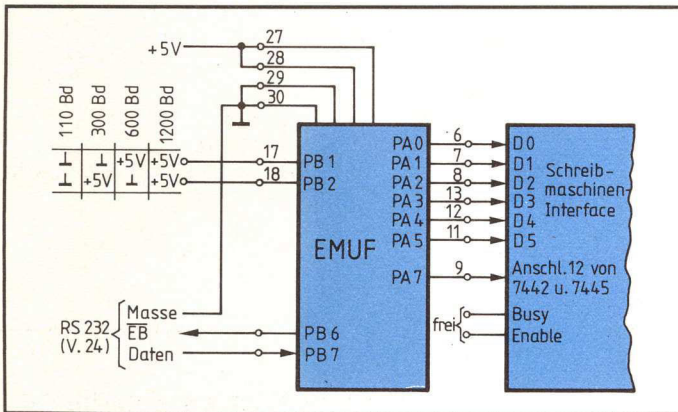


Bild 2. EMUF als V.24-Schnittstelle für den Schreibmaschinen-Drucker. Format des Eingangssignals: 8 Bit oder 7 Bit mit Parität (das 8. Bit wird in jedem Fall ignoriert)

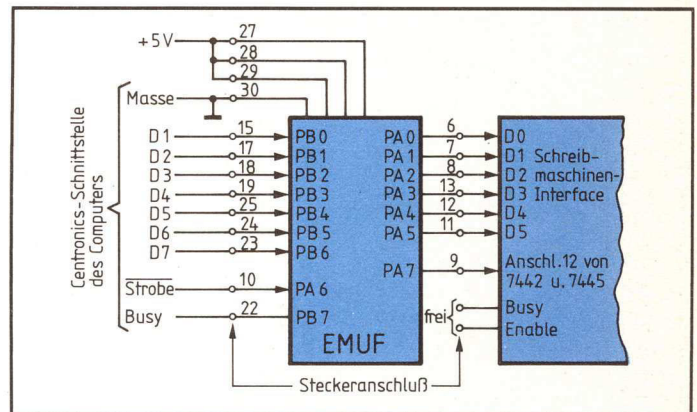


Bild 3. EMUF als Centronics-Schnittstelle für den Schreibmaschinen-Drucker

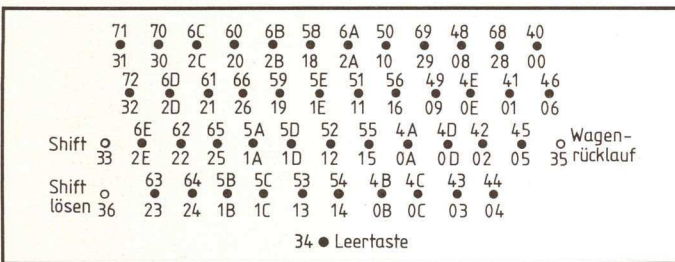


Bild 4. Position der Magneten und zugehörige Codes (unten: ohne Shift, oben: mit Shift)

AIM schießt EPROM

In dem von Siemens (Fürth) erhältlichen „Applikationsbuch PC-100“ ist beschrieben, wie man auf äußerst einfache Weise mit dem Computer PC-100 bzw. AIM-65 EPROMs programmiert. An Hardware sind dazu lediglich eine Diode und ein EPROM-Sockel erforderlich. Das auf Seite 41 dieses Applikationsbuches abgedruckte Programm dafür ist allerdings für den Adressenbereich ab D300 geschrieben, wo bei den meisten Anwendern kein RAM steht. Da besonders für das Erstellen von Programmen für den Einplatinen-Computer „EMUF“ (mc 2/1981) die Adressenlage ab 0200 besser geeignet ist, wird hier der nötige „Software-Umbau“ beschrieben.

Zunächst sind alle Bytes des Listings nicht ab D300, sondern ab 0200 in den PC-100 einzutippen; die letzte Adresse ist nun nicht mehr D47B, sondern 037B. Die Bytes an folgenden Adressen sind dann von D3 auf 02 zu ändern: 0264, 027A, 028C, 029B, 02DB, 037B. An folgenden Adressen ist D4 durch 03 zu ersetzen: 0227, 0242, 0267, 026F, 027D, 0294, 029E, 029E, 02A8, 02B5, 02C5, 02CA, 02CE. Die Startadresse ist dann 0200; ein Start mit der F1-Taste wird möglich, indem man ab 010C schreibt: 4C 00 02. Die Programmentwicklung für den „EMUF“ kann dann wie folgt geschehen: Zunächst schreibt man den Quellcode

mit dem Texteditor, der von 0200 bis 0A00 initialisiert wird. Mit dem ROM-residenten Assembler kann man dann den Objektcode in der endgültigen EPROM-Lage 0C00...0FFF erzeugen, wobei man die Assembler-Symboltabelle von 0A00 bis 0C00 initialisieren kann. Nachdem der Quellcode auf Kassette „gerettet“ ist, lädt man das gerade beschriebene EPROM-Programm, startet es (FROM = C00, TO = FFF, PROM STARTS = 0) und kann ohne weitere Adressenanpassungen direkt das EPROM „schießen“. Das beschriebene Vorgehen macht den PC-100 zu einem denkbar komfortablen und leistungsfähigen Entwicklungssystem für den Einplatinen-Computer EMUF.

Fe.

Rolf-Dieter Klein

EMUF bringt Strichcode zum IEC-Bus

Hier wird ein Programm vorgestellt, das es ermöglicht, mit dem Einplatinencomputer EMUF (s. Heft 2/1981) ein IEC-Bus-Interface z. B. für CBM-Rechner herzustellen, das den Anschluß eines mc-Barcode-Lesers erlaubt. Die IEC-Bus-Funktionen werden vom EMUF per Software simuliert, so daß praktisch keinerlei zusätzliche Hardware nötig ist.

Bild 3 zeigt das Listing unseres Programms. Die Barcode-Routine ist im wesentlichen dabei aus Heft 1/1981 übernommen. Nach dem Einschalten des EMUF erfolgt ein kurzes Piep-Signal als Zeichen, daß das Interface betriebsbereit ist. Danach wird in das IEC-Hauptprogramm gesprungen. Nach Empfang einer

Bild 1 zeigt die Anordnung. Der Rechner ist mit dem EMUF über den IEC-Bus verbunden, der Leser mit einem Anschluß des EMUF. Ein Lautsprecher gibt einen kurzen Piep-Ton ab, wenn eine Barcode-Zeile erfolgreich eingelesen wurde. Bild 2 zeigt die Anschlußbelegung am EMUF. Der IEC-Datenbus wird direkt mit dem Port PA verbunden. Die fünf Handshakesignale des IEC-Bus werden vom Port PB gewonnen. Dabei werden die restlichen Signale, die noch am IEC-Bus vorhanden sind, für unsere Zwecke nicht gebraucht. Im Bild sind zwei verschiedene Pinbelegungen für den IEC/IEEE-Stecker angegeben. Beim einen handelt es sich um die Version mit dem 25poligen Stecker; die 24polige Version ist am CBM-Rechner zu finden. Der Lautsprecher wird über eine einfache Schaltung mit Bit 6 des Ports PB verbunden. Der Lesestift wird an Bit 7 des Ports PB angeschlossen.

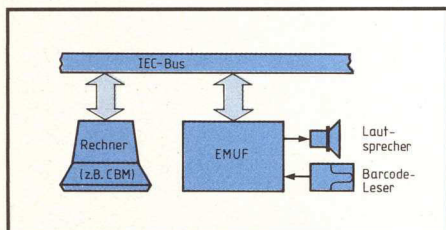


Bild 1. Am IEC-Bus muß der EMUF immer zuerst angesprochen werden. Die dabei verwendete Adresse merkt er sich

Bild 3. Assembler-Listing der EMUF-Software. Sie wurde auf einem Z-80-System mit einem 6502-Crossassembler entwickelt. Vom Basic-Rechner aus muß der IEC-Bus-Barcode-Leser mit GET (A\$) angesprochen werden

```

0056          WIDTH 80
;PORT A
; 7 .. 0 IEC DATABUS
;PORT B
; 7 6 5 4 3 2 1 0
; BAR BEL -EOI -NOAC -NRDF -DAC -ATN
;
0000 PA EQU $800
0001 PAD EQU $801
0002 PB EQU $802
0003 PBD EQU $803
;
0000 XTEMP1 EQU $0
0001 REF0 EQU $1
0002 REF1 EQU $2
0003 REF EQU $3
0004 BUF EQU $4
0005 CHCKL EQU $5
0006 CHCKH EQU $6
0007 XTEMP EQU $7
0008 PFFLAG EQU $8
0009 PNT EQU $9
000A FLAG EQU $A ;IEC MERKER ATN ..
000B ZEICH EQU $B ;ZWSPEICHER
000C COUNT EQU $C ;ZWSPEICHER
000D PADR EQU $D ;FIRST TIME
000E ZEIT EQU $E
;
; INIT ROUTINE
00FC ORG $FFC
00FC 000C DW $C00
;
0C00 ORG $C00 ;START
0C00 A2FF RESET: LOX #$FF ;STACKPOINTER
0C02 9A TXS
0C03 A900 LDA #00000000 ;ALL INPUT
0C05 8D0108 STA PAD
0C08 A94C LDA #01001100 ;SET UP
0C0A 8D0308 STA PBD
0C0D A900 LDA #00000000 ;NOT READY NOT ACCEPT
0C0F D8 CLD
0C10 78 SEI
0C11 A9FF LDA #$FF
0C13 8508 STA PFFLAG
0C15 850D STA PADR ;START WERT
0C17 261D0C JSR BELL1 ;AUSGEBEN TON FUER CHECK
0C1A
0C1A 4CA20D JMP MAIN
; UPRGE
0C1D A9FF BELL1: LDA #$FF ;BELL AUSGEREN
0C1F 48 LOP1: PHA
0C20 AD0208 LDA PB
0C23 4940 EOR #01000000 ;CHANGE

```

| | | | |
|-------------|---------|------------|-----------|
| 0C25 8D0208 | STA | PB | |
| 0C28 A950 | LDA | ##60 | |
| 0C2A EA | LOP2: | NOP | |
| 0C2B EA | | NOP | |
| 0C2C EA | | NOP | |
| 0C2D E901 | SBC | #1 | |
| 0C2F D0F9 | BNE | LOP2 | |
| 0C31 48 | PLA | | |
| 0C32 E901 | SBC | #1 | |
| 0C34 D0E9 | BNE | LOP1 | |
| 0C36 AD0008 | LDA | PA | |
| 0C39 298F | AND | ##10111111 | |
| 0C3B 8D0008 | STA | PA | |
| 0C3E 60 | RTS | | |
| ; | | | |
| 0C3F 8500 | READER: | STX | XTEMP1 |
| 0C41 2408 | | BIT | PFFLAG |
| 0C43 300E | | BMI | START |
| 0C45 A609 | | LDX | PNT |
| 0C47 E404 | EING: | CPX | BUF |
| 0C49 F008 | | BEQ | START |
| 0C4B 3D1000 | | LDA | BUFFER, X |
| 0C4F E609 | | INC | PNT |
| 0C50 A400 | | LDX | XTEMP1 |
| 0C52 60 | | RTS | |
| ; | | | |
| 0C53 2C0208 | START: | BIT | PB |
| 0C54 30FB | | BMI | START |
| 0C58 20F30C | | JSR | SYNCH |
| 0C5B 8502 | | STX | REF1 |
| 0C5D 8601 | | STX | REF0 |
| 0C5F 4601 | | LSR | REF0 |
| 0C61 20F30C | | JSR | SYNCH |
| 0C64 20A80C | | JSR | AUSW |
| 0C67 9076 | | BCC | FEHLER |
| 0C69 20E20C | | JSR | BYTE |
| 0C6C 8504 | | STA | BUF |
| 0C6E A200 | | LDX | #0 |
| 0C70 A900 | | LDA | #0 |
| 0C72 8505 | | STA | CHCKL |
| 0C74 8506 | | STA | CHCKH |
| 0C76 8607 | VOR: | STX | XTEMP |
| 0C78 20E20C | | JSR | BYTE |
| 0C7B A607 | | LDX | XTEMP |

| | | | |
|-------------|---------|-----------|---------------------|
| 0C7D 9D1000 | STA | BUFFER, X | |
| 0C80 18 | CLC | | |
| 0C81 6505 | ADC | CHCKL | |
| 0C83 8505 | STA | CHCKL | |
| 0C85 9002 | BCC | NULL | |
| 0C87 E606 | INC | CHCKH | |
| 0C89 E8 | NULL: | INX | |
| 0C8A E404 | | CPX | BUF |
| 0C8C D0E8 | | BNE | VOR |
| 0C8E 20E20C | | JSR | BYTE |
| 0C91 C505 | | CMP | CHCKL |
| 0C93 D04A | | BNE | FEHLER |
| 0C95 20E20C | | JSR | BYTE |
| 0C98 C506 | | CMP | CHCKH |
| 0C9A D043 | | BNE | FEHLER |
| 0C9C 201D0C | | JSR | BELLI ;READY MELDEN |
| 0C9F A200 | | LDX | #0 |
| 0CA1 8609 | | STX | PNT |
| 0CA3 8608 | | STX | PFFLAG |
| 0CA5 4C470C | | JMP | EING |
| ; | | | |
| 0CA8 A501 | AUSW: | LDA | REF0 |
| 0CAA 4A | | LSR | A |
| 0CAB 6501 | | ADC | REF0 |
| 0CAD 8503 | | STA | REF |
| 0CAF E403 | | CPX | REF |
| 0CB1 1009 | | BPL | E.INS |
| 0CB3 8601 | | STX | REF0 |
| 0CB5 A501 | | LDA | REF0 |
| 0CB7 0A | | ASL | A |
| 0CB9 8502 | | STA | REF1 |
| 0CBA 18 | | CLC | |
| 0CBB 60 | | RTS | |
| 0CBC 8602 | EINS: | STX | REF1 |
| 0CBE A502 | | LDA | REF1 |
| 0CC0 4A | | LSR | A |
| 0CC1 8501 | | STA | REF0 |
| 0CC3 38 | | SEC | |
| 0CC4 60 | | RTS | |
| ; | | | |
| 0CC5 48 | BITTST: | PHA | |
| 0CC6 A200 | | LDX | #0 |
| 0CC8 2C0208 | BIT: | BIT | PB |
| 0CCB 10FB | | BPL | BIT |

Primäradresse wird verglichen, ob dies die erste empfangene ist; wenn ja, so wird ab sofort nur noch auf diese Adresse reagiert. Damit muß der EMUF das erste Gerät sein, daß auf dem IEC-BUS angesprochen wird(!). Die dabei verwendete Primäradresse wird von da ab verwendet.

Bild 4 zeigt das Handshake-Verhalten des IEC-Bus. Wird ein Datenwert auf dem IEC-Bus übertragen, so wird zunächst geprüft, ob alle Geräte fertig sind. Wenn ja, so wird das Datum angelegt und ein DAV-Signal gegeben. Dann reagieren die Geräte mit NDAC und zeigen damit, daß sie das Datum aufgenommen haben. DAV wird nun weggenommen, danach auch NDAC. Das Handshaking gilt für alle Richtungen, also CBM an EMUF und umgekehrt.

Eine Leitung ATN zeigt zusätzlich an, wann eine Adresseninformation auf dem Bus liegt. Dabei wird bei Beginn eines IEC-Zyklus zuerst eine Primäradresse übertragen. Zusätzlich gibt es die Information, ob später eine Datenein- oder Ausgabe erfolgen soll. Die Unterscheidung dafür liegt in den höherwertigen Bits: Wird als Talker (Daten-

sender) adressiert, so fühlt sich unser EMUF angesprochen, denn dann gilt es, Daten zu übertragen. Dazu wird die Routine READER aufgerufen, die ein Zeichen vom Codeleser holt. Das Zeichen wird in ZEICH gerettet. Jetzt wird die Sekundäradresse geholt; der Wert wird ignoriert. Wir brauchen die Sekundäradresse, da sich beim CBM das Handshake nicht an jeder Stelle beliebig lang anhalten läßt. Nach der Sekundäradresse will der CBM schnell seine Daten haben. Dazu senden wir den Inhalt von ZEICH. Danach ist ein Zyklus beendet. Die Daten werden mit EOI gesendet, um zu zeigen, daß nur ein Wert kommt. Die Werte können dann mit GET# geholt werden. Der Befehl INPUT ist nicht geeignet, da er manche Zeichen („:“ und „,“,“) verschluckt.

Im CBM-Rechner muß bei Kleinbuchstaben ggf. eine Umrechnung erfolgen, da die Daten unverändert übertragen werden. Kleinbuchstaben in mc-Programmen haben den Wertbereich hex 60 bis 7F und müssen durch Subtraktion von hex 20 in den CBM-Bereich gebracht werden.

Programme, die nur Großbuchstaben

enthalten, können direkt eingelesen werden. Tabelle 1 und Tabelle 2 zeigen zwei mögliche Einleseprogramme beim CBM. Im ersten Fall wird immer eine Zeile eingelesen und auf dem Bildschirm ausgegeben. Danach fährt man mit dem Cursor an diese Zeile und betätigt Return, wodurch die Zeile ins Programm übernommen wird. Dann wird

Tabelle 1: Zeilenweises Einlesen beim CBM

```
1000 OPEN 1,9,15
10010 GET #1,A$
10020 PRINT A$;
10030 IF A$<> CHR$(13) THEN 10010
10040 PRINT CHR$(7)
```

Tabelle 2: Ausgabe aller Zeilen auf dem CBM-Schirm

```
10 OPEN 1,8,15
20 GET #1, A$
30 PRINT A$;
40 IF A$= CHR$(13) THEN PRINT CHR$(7);
50 GOTO 20
```

mit RUN 10000 das Programm neu gestartet und die nächste Zeile eingelesen. Tabelle 2 zeigt ein Programm, das alle Zeilen ausgibt; es muß allerdings danach irgendwie gestoppt werden, z. B. durch Herausziehen des IEC-Steckers. Die Version ist zwar von der Eingabe her bequemer, kann aber nur sehr kurze mc-Programme verarbeiten. Hier ist ein Betätigungsfeld für CBM-Spezialisten. Vielleicht ist es einfacher möglich, direkt vom IEC-Bus per Maschinenprogramm die Daten einzulesen und sofort als Programm abzulegen. Dazu muß z. B. eine Zeile in einem Puffer abgelegt werden und dann in die Interdarstellung umgewandelt werden. Das IEC-Businterface eignet sich übrigens auch für andere IEC-Bus-Rechner; ggf. kann dann die Sekundäradresse entfallen.

Das programmierte EPROM „IEC BAR“ ist von der Fa. Elektronikladen in Detmold erhältlich, die auch den EMUF-Bausatz liefert.

Literatur

- [1] Apple-II liest Strichcode. mc 1981, Heft 1.
- [2] Mädchen für alles (EMUF). mc 1981, Heft 2.
- [3] Strichcode-Programme: mc 1981, Hefte 1...4.
- [4] IEC-Bus. Sonderheft Nr. 47, Franzis-Verlag.

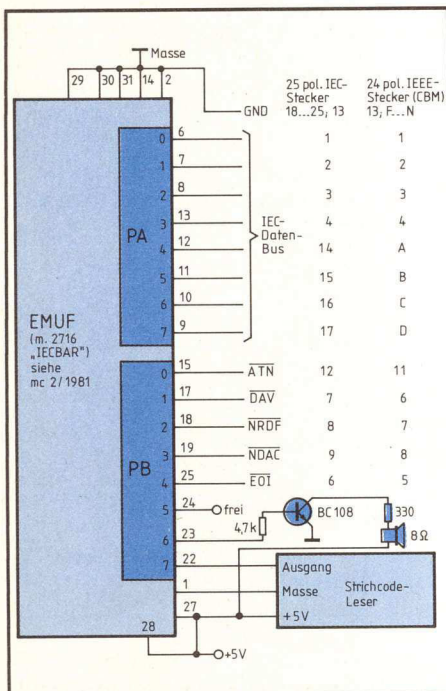


Bild 2. Anschluß von Strichcode-Leser, IEC-Bus-Stecker und Lautsprecher an den Einplatinen-Computer EMUF

```

0CC0 E8      ZAEHL:  INX
0CCE A908    LDA      #ZEIT
0CD0 E901    VERZ:   SBC      #1
0CD2 D0FC    BNE      VERZ
0CD4 E0FF    CPX      #255
0CD6 F021    BEQ      UEBRL
0CD8 2C6208  BIT      PB
0CDB 30F0    BMI      ZAEHL
0CDD 68      PLA
0CDE 60      RTS

;
0CDF 4C530C  FEHLER: JMP      START
;
0CE2 48      BYTE:   PHA
0CE3 A008    LDY      #8
0CE5 20C50C  NBIT:   JSR      BITTST
0CE8 26A80C  JSR      AUSW
0CEB 68      PLA
0CEC 6A      ROR      A
0CED 48      PHA
0CFE 88      DEY
0CFE D0F4    BNE      NBIT
0CF1 68      PLA
0CF2 60      RTS

;
0CF3 48      SYNCH:  PHA
0CF4 26C50C  JSR      BITTST
0CF7 68      PLA
0CF8 60      RTS

;
0CF9 68      UEBRL:  PLA
0CFA 68      PLA
0CFB 68      PLA
0CFC 68      PLA
0CFD 68      PLA
0CFE 68      PLA
0CFF 4C530C  JMP      START
; IEC ROUTINEN
;
0D02 A904    GETCHA:  LDA      #%00000100 ;RDF
0D04 8D0208  STA      PB
0D07 AD0208  LOPA:   LDA      PB
0D0A 2902    AND      #%00000010 ;DAV WARTEN
0D0C D0F9    BNE      LOPA
0D0E A900    LDA      #%00000000
0D10 8D0208  STA      PB
0D13 AD0008  LDA      PA ;DATA HOLEN
0D16 49FF    EOR      #%11111111
0D18 48      PHA
0D19 AD0208  LDA      PB
0D1C 850A    STA      FLAG
0D1E A908    LDA      #%00001000 ;DAC
0D20 8D0208  STA      PB
0D23 AD0208  LOPB:   LDA      PB
0D26 2902    AND      #%00000010
0D28 F0F9    BEQ      LOPB ;DAV HIGH
0D2A A900    LDA      #%00000000
0D2C 8D0208  STA      PB
0D2F 68      PLA ;DATA WERT
0D30 60      RTS

;
0D31 AD0208  TALKON: LDA      PB
0D34 2901    AND      #%00000001
0D36 F0F9    BEQ      TALKON ;WARTEN BIS ATN WEG
0D38 A912    LDA      #%00010010
0D3A 8D0208  STA      PB ;DAV HIGH
0D3D A9FF    LDA      #%11111111
0D3F 8D0108  STA      PAD ;DATA CHANGE
0D42 A952    LDA      #%01010010
0D44 8D0308  STA      PBD
0D47 A912    LDA      #%00010010
0D49 8D0208  STA      PB ;SAVETY
0D4C 60      RTS

;
0D4D A900    TALKOF: LDA      #%00000000
0D4F 8D0108  STA      PAD
0D52 A900    LDA      #%00000000
0D54 8D0208  STA      PB
0D57 A94C    LDA      #%01001100
0D59 8D0308  STA      PBD
0D5C A900    LDA      #%00000000
0D5E 8D0208  STA      PB
0D61 60      RTS
    
```

```

;
;
0D62 48      SEND1: PHA
0D63 208800 SEND: JSR   CKATN ;TESTER GGF
0D66 AD0200  LDA   PB
0D69 2904    AND   #%00000100
0D6B F0F6    BEQ   SEND ;WARTEN
0D6D 68      PLA
0D6E 49FF    EOR   #%11111111
0D70 8D0000 STA   PA
0D73 A910    LDA   #%00010000 ;DAV
0D75 800200 STA   PB
0D78 208800 CONSE: JSR   CKATN
0D7B AD0200  LDA   PB
0D7E 2908    AND   #%00001000
0D80 F0F6    BEQ   CONSE
0D82 A912    LDA   #%00010010 ;PASSIV
0D84 8D0200 STA   PB
0D87 60      RTS

0D88 60      ; CKATN: RTS ;Z.Z.
;
0D89 48      SENE01: PHA
0D8A 208800 SENE01: JSR   CKATN
0D8D AD0200  LDA   PB
0D90 2904    AND   #%00000100
0D92 F0F6    BEQ   SENE01
0D94 68      PLA
0D95 49FF    EOR   #%11111111
0D97 8D0000 STA   PA
0D9A A900    LDA   #%00000000 ;EOI DAV
0D9C 8D0200 STA   PB
0D9F 4C7800  JMP   CONSE

;
0DA2        ; MAIN:
0DA2 200200  JSR   GETCHA ;IEC ZEICHEN
0DA5 8508    STA   ZEICH ;RETZEN
0DA7 A50A    LDA   FLAG
0DA9 2901    AND   #%00000001 ;ATN
0DAB D0F5    BNE   MAIN ;HIGH DANN NEIN
0DAD A50B    LDA   ZEICH
0DAF 2960    AND   #%60 ;TEST LISTEN TALK
0DB1 C920    CMP   #%20
0DB3 F007    BEQ   LISTPA
0DB5 C940    CMP   #%40
0DB7 F006    BEQ   TALPA
0DB9 4CA200  JMP   MAIN

0DBC 4CA200  ; LISTPA: JMP MAIN ;HIER NICHT
;
0DBF A50D    ; TALPA: LDA PADR ;PRIM ADRESSE
0DC1 C9FF    CMP   #%FF
0DC3 D009    BNE   SK2 ;WEITER SONST
0DC5 A50B    LDA   ZEICH
0DC7 290F    AND   #%0F
0DC9 850D    STA   PADR ;NEUE ADRESSE
0DCB 4CD600  JMP   SK3
0DCE A50B    SK2: LDA ZEICH
0DD0 290F    AND   #%0F
0DD2 C50D    CMP   PADR ;VERGLEICH
0DD4 D0CC    BNE   MAIN
0DD6 203F00  SK3: JSR   READER ;HOLE EIN ZEICHEN
0DD9        ;FUER GET
0DD9        ;GET WEGEN : UND ,
0DD9 850B    STA   ZEICH
0DDB 200200  JSR   GETCHA ;IGNORE SA
0DDE        ;SA MUSS DA SEIN
0DDE 203100  JSR   TALKON ;CHANGE DIR
0DE1 A50B    LDA   ZEICH
0DE3 208800  JSR   SENE01 ;1 ZEICHEN SENDEN NUN
0DE6 204000  JSR   TALKOF ;UMSCHALTEN
0DE9 4CA200  JMP   MAIN ;ALLES VON VORNE

;
;
; RAM GEBIET
;
0010        ; ORG #10
0010        ; BUFFER DS 24 ;PUFFER BARCODE
0028        ; ENE DS 1 ;CHECK
;
0000        ; END

```

---- SYMBOL TABLE ----

| | |
|--------|------|
| AUSW | 0CA8 |
| BELL1 | 0C10 |
| BIT | 0CC8 |
| BITTST | 0CC5 |
| BUF | 0004 |
| BUFFER | 0010 |
| BYTE | 0CE2 |
| CHCKH | 0006 |
| CHCKL | 0005 |
| CKATN | 0D88 |
| CONSE | 0D78 |
| COUNT | 000C |
| EING | 0C47 |
| EING | 0C8C |
| ENE | 0028 |
| FEHLER | 0CDF |
| FLAG | 000A |
| GETCHA | 0D92 |
| LISTPA | 0DBC |
| LOP1 | 0C1F |
| LOP2 | 0C2A |
| LOPA | 0D67 |
| LOPB | 0D23 |
| MAIN | 0DA2 |
| NBIT | 0CE5 |
| NULL | 0C89 |
| PA | 0000 |
| PAD | 0001 |
| PADR | 000D |
| PB | 0002 |
| PBD | 0003 |
| PFFLAG | 0008 |
| PNT | 0009 |
| READER | 0C3F |
| REF | 0003 |
| REF0 | 0001 |
| REF1 | 0002 |
| RESET | 0C06 |
| SEND | 0D63 |
| SEND1 | 0D62 |
| SENE01 | 0D89 |
| SENE01 | 0D8A |
| SK2 | 0DCE |
| SK3 | 0DD6 |
| START | 0C53 |
| SYNCH | 0CF3 |
| TALKOF | 0D4D |
| TALKON | 0D31 |
| TALPA | 0DBF |
| UEBRL | 0CF9 |
| VERZ | 0CD0 |
| VOR | 0C76 |
| XTEMP | 0007 |
| XTEMP1 | 0000 |
| ZAHL | 0CCD |
| ZEICH | 000B |
| ZEIT | 0008 |

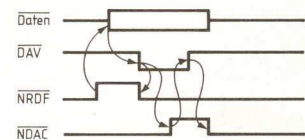


Bild 4. Handshake-Verhalten beim IEC-Bus. Die Signale können in beide Richtungen übertragen werden

Herwig Feichtinger

Türklingel und Alarmanlage

In nur 0,6 KByte läßt sich eine ganze Menge unterbringen – das beweist auch die folgende Applikation, die den EMUF im trauten Heim beschäftigt. Er fragt acht Alarmkontakt-Eingänge ab und spielt außerdem beim Druck auf den Türklingelknopf eine von vier vorprogrammierten Melodien ab.

Beim Einschalten, d. h. nach einem Reset, müssen alle acht Alarmeingänge des EMUF auf High-Potential liegen, andernfalls tritt sofort eine Alarmmeldung auf. Damit läßt sich z. B. überprüfen, ob alle Fenster und Türen beim Scharfmachen auch wirklich geschlossen sind. Anschließend führen irgendwelche Änderungen der logischen Pegel an diesen Eingängen zu einem maximal etwa 25 Sekunden dauernden Alarm (die Dauer ist gesetzlich begrenzt), der natürlich durch einen Schalter an PB7 jederzeit unterbrochen werden kann. Dieser Schalter dient auch zum Abschalten der Alarmanlage, solange man sich zu Hause aufhält (Bild 1).

Mit einem Siebensegment-Display ist es möglich, festzustellen, welcher Eingang den Alarm auslöste. Die Anzeige bleibt auch nach Ablauf der 25 s Alarmdauer bestehen und erlischt erst, wenn man den Alarm ausschaltet. Führen mehrere Eingangsänderungen zum Alarm, so wird derjenige Eingang mit der höchsten Nummer angezeigt. Die elektronische Türklingel funktioniert völlig unabhängig von der Alarmanlage. Sie enthält vier Melodien, die abwechselnd gespielt werden. Drückt man den Klingelknopf während das Abspielen einer Melodie, so ignoriert das der Computer.

Während die Klingel-Melodie über eine einfache Lautsprechertreiberschaltung wiedergegeben wird, dient als Alarmausgang ein Relais, mit dem man z. B. eine Sirene antreiben kann, um die gegenüber der Klingel doch erheblich höhere nötige Lautstärke zu erzielen. Will man vor unangenehmen Überraschungen sicher sein, so sollte man dar-

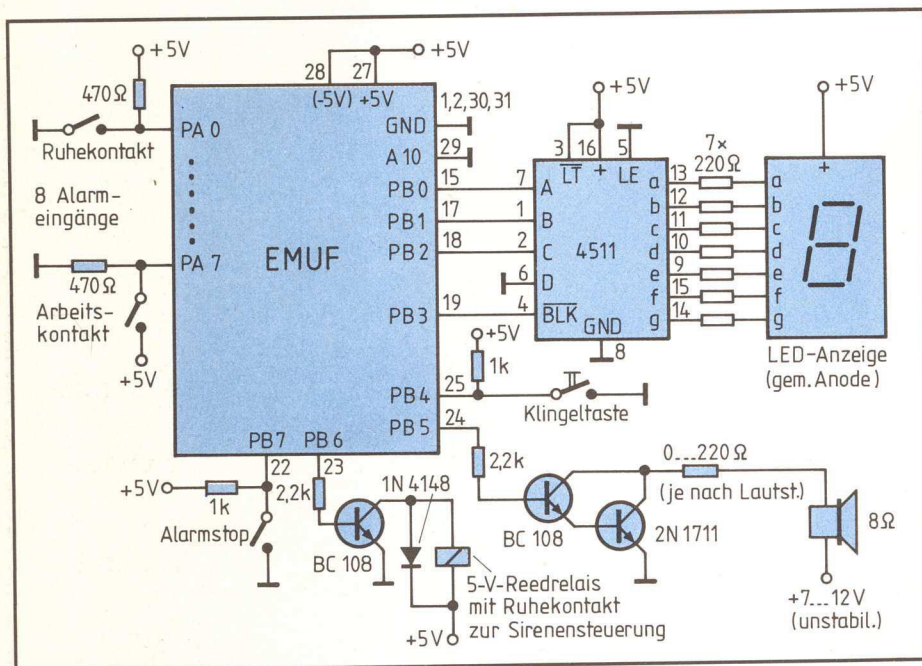


Bild 1. Verdrahtung des EMUF. An jeden der acht Alarmeingänge kann wahlweise ein Arbeits- oder Ruhekontakt angeschlossen werden

Bild 2. Assemblerlisting des Programms. Das Hauptprogramm dient als Türklingel, während die Alarmanlage vollständig als Interrupt-Programm aufgebaut ist

```

0000 ;EMUF-ALARM+MUSIK
0000 ;H.FEICHTINGER/1981
0000 ;EMUF: S."MC"2/1981
0000 ;PA0-7:ALARMEING.
0000 ;PB7=AL.STOP-EING.*
0000 ;PB6=ALARMAUSG.*
0000 ;PB5=MUSIKAUSG.
0000 ;PB4=KLINGELTASTE*
0000 ;PB3=BLANK DISPL.*
0000 ;PB0-2:DISPL.(7447)
0000 ;(ACTIV LOW = *)
0000 DUR =250 ;25SEC ALRM
0000 * =0
0000 SAVX * =*+1
0001 TI * =*+1
0002 LST * =*+1
0003 WORK * =*+6
0009 LIMIT * =*+3
000C VAL2 * =*+1
000D VAL1 * =*+1
000E TIMER * =*+1
000F XSAV * = $C00
0C00 MUS = $D00
0C00 PA = $800
0C00 PB = $802
0C00 PBD = $803
0C00 CLK = $81F
0C00 RESV = $FFC
0C00 IRQV = $FFE
0C00 RES A2FF LDX £$FF ;RESET
8602 STX LST ;NORMAL:
9A TXS ;PA0-7=H
A96F LDA £%01101111
8D0308 STA PBD
    
```

```

OC0A      A9D0   LDA  £%11010000
OC0C      8D0208 STA  PB
OC0F      8D1F08 STA  CLK      ;IRQ-START
OC12      58     CLI
OC13      D8     CLD
OC14 FRST  A205   LDX  £5      ;ERSTE
OC16 LP1   BDB20C LDA  INIT,X   ;MEL.
OC19      9503   STA  WORK,X
OC1B      CA     DEX
OC1C      10F8   BPL  LP1
OC1E DEB   A200   LDX  £0      ;KLINGEL-
OC20 WAIT  ADO208 LDA  PB      ;TASTE?
OC23      2910   AND  £%00010000
OC25      DOF7   BNE  DEB      ;NEIN
OC27      CA     DEX
OC28      DOF6   BNE  WAIT
OC2A GO    A000   LDY  £0      ;MUSIK-
OC2C      B107   LDA  (WORK+4),Y
OC2E      C9FF   CMP  £$FF     ;PROGR.
OC30      FOE2   BEQ  FRST
OC32      E607   INC  WORK+4   ;AEHNL.
OC34      D002   BNE  *+4     ;JIM BUTTER-
OC36      E608   INC  WORK+5   ;FIELD'S
OC38      C9FA   CMP  £$FA     ;MUSICBOX
OC3A      FOE4   BEQ  WAIT
OC3C NEXT  900F   BCC  NOTE   ;AUS
OC3E      E9FB   SBC  £$FB     ;DEM
OC40      AA     TAX      ;FIRST
OC41      B107   LDA  (WORK+4),Y
OC43      E607   INC  WORK+4   ;BOOK
OC45      D002   BNE  *+4     ;OF KIM
OC47      E608   INC  WORK+5
OC49      9503   STA  WORK,X
OC4B      B0DD   BCS  GO
OC4D NOTE  A603   LDX  WORK
OC4F      860A   STX  LIMIT+1
OC51      A604   LDX  WORK+1
OC53      A8     TAY
OC54      3002   BMI  OVER
OC56      A201   LDX  £1
OC58 OVER  8609   STX  LIMIT
OC5A      297F   AND  £$7F
OC5C      850C   STA  VAL2
OC5E      FO02   BEQ  HUSH
OC60      850D   STA  VAL1
OC62 HUSH  A50C   LDA  VAL2
OC64      2506   AND  WORK+3
OC66      FO04   BEQ  ON
OC68      E60D   INC  VAL1
OC6A      C60C   DEC  VAL2
OC6C ON    A60C   LDX  VAL2
OC6E      78     SEI      ;WEGEN PB-
OC6F      ADO208 LDA  PB      ;PROBL.
OC72      0920   ORA  £%00100000
OC74      20880C JSR  SOUND   ;BEI IRQ
OC77      30B1   BMI  GO
OC79      A60D   LDX  VAL1
OC7B      78     SEI
OC7C      ADO208 LDA  PB
OC7F      29DF   AND  £%11011111
OC81      20880C JSR  SOUND
OC84      30A4   BMI  GO
OC86      10DA   BPL  HUSH
OC88 SOUND A405   LDY  WORK+2
OC8A      840E   STY  TIMER
OC8C      860F   STX  XSAV
OC8E SLOOP E000   CPX  £0
OC90      D008   BNE  CONT
OC92      A60F   LDX  XSAV
OC94      C60E   DEC  TIMER
OC96      DOF6   BNE  SLOOP
OC98      FO16   BEQ  SEX
OC9A CONT  8D0208 STA  PB
OC9D      CA     DEX
OC9E      C60B   DEC  LIMIT+2
OCA0      DOEC   BNE  SLOOP
OCA2      C60A   DEC  LIMIT+1
OCA4      DOE8   BNE  SLOOP
OCA6      A403   LDY  WORK
OCA8      840A   STY  LIMIT+1
OCAA      C609   DEC  LIMIT
OCAC      DOE0   BNE  SLOOP
OCAE      A9FF   LDA  £$FF
OCBO SEX   58     CLI
OCB1      60     RTS
OCB2 INIT  30     .BYT 48,2,1,$FF
OCB3      02
OCB4      01
OCB5      FF
OCB6      000D   .WOR MUS
OCB8      ;INTERRUPT-ROUTINE
OCB8      ;FUER ALARMANLAGE
OCB8 IRQ   48     PHA
OCB9      ADO208 LDA  PB      ;STOPTASTE?
OCBC      3009   BMI  IRQ0     ;NEIN
OCBE      0940   ORA  £%01000000
OCC0      29F7   AND  £%11110111
OCC2      8D0208 STA  PB      ;AL.+DISP.
OCC5      D030   BNE  IRQ2     ;AUS
OCC7 IRQ0  ADO008 LDA  PA
OCCA      4502   EOR  LST      ;AENDERG.?
OCCC      FO19   BEQ  IRQ1
OCCE      8600   STX  SAVX     ;JA
OCDO      AE0008 LDX  PA
OCD3      8602   STX  LST      ;X RETTEN
OCD5      A2FF   LDX  £$FF
OCD7 IRQ3  2A     ROL  A      ;DISPLAY-
OCD8      E8     INX      ;CODE
OCD9      90FC   BCC  IRQ3     ;ERZEUGEN
OCDB      8A     TXA
OCDC      A600   LDX  SAVX
OCDE      0998   ORA  £%10011000
OCEO      8D0208 STA  PB      ;AL.EIN
OCE3      A9FA   LDA  £DUR
OCE5      8501   STA  TI
OCE7 IRQ1  A501   LDA  TI      ;ZEIT
OCE9      FO0C   BEQ  IRQ2     ;VORBEI?
OCEB      C601   DEC  TI
OCED      D008   BNE  IRQ2
OCEF      ADO208 LDA  PB      ;JA
OCF2      0940   ORA  £%01000000
OCF4      8D0208 STA  PB      ;AL.AUS
OCF7 IRQ2  A964   LDA  £100
OCF9      8D1F08 STA  CLK
OCFC      68     PLA
OCFD      40     RTI
OCFE      *=RESV
OFFC      000C   .WOR RES
OFFE      *=IRQV
OFFE      B80C   .WOR IRQ
1000      .END
1000      ERRORS= 0000

```

auf achten, daß nicht durch die Verkabelung unerwünschte Störimpulse in den Computer gelangen und so zum „Aufhängen“ des Programms führen. Im schlimmsten Fall ertönt dann die Sirene dauernd und schaltet freiwillig nicht mehr ab. Bewährt haben sich z. B. abgeschirmte Leitungen zu den Alarmkontakten und zur Klingeltaste. Auch muß man aufpassen, daß diese Kontakte nicht ihrerseits zusätzlich mit anderen verbunden sind, z. B. mit metallischen Fensterrahmen u. ä., um Masseschleifen zum vermeiden. In besonders kritischen Fällen kann man natürlich durch den Einsatz von Optokopplern solche Probleme radikal vermeiden. Ein eindeutiges Anzeichen für einen „aufgehängten“ Computer ist das Nichtfunktionieren der Klingeltaste. Bild 2 zeigt das vollständige EMUF-Programm im Assemblercode, und Bild 3 gibt schließlich den Hex-Dump mit den Codierungen der vier Melodien an, die man bei der EPROM-Programmierung nicht vergessen darf.

Bild 3. Das ist der Datenblock für die vier Melodien, die die EMUF-Türklingel spielen kann

```
( )=OD00 FB 18 FE FF 44 51 E6 E6 66 5A 51 4C C4 C4 C4 D1
( ) OD10 BD BD BD 00 44 BD 00 44 3D 36 33 2D A8 80 80 33
( ) OD20 44 B3 80 80 44 51 C4 80 80 5A 51 E6 80 80 FA FE
( ) OD30 00 FB 28 5A 5A 51 48 5A 48 D1 5A 5A 51 48 DA E0
( ) OD40 5A 5A 51 48 44 48 51 5A 60 79 6C 60 DA DA FA FE
( ) OD50 FF 5A 5A 5A 5A 5A 5A 66 72 79 E6 E6 80 00 56 56
( ) OD60 56 56 56 56 5A 66 F2 80 80 4C 4B 4C 4C 4C 56
( ) OD70 5A 56 4C 00 C4 44 4C 56 5A 5A 56 5A 66 56 5A 66
( ) OD80 F2 80 FE 00 00 72 5A CC 72 5A CC 72 5A CC 80 B8
( ) OD90 80 4C 56 5A 56 5A E6 F2 80 FA FE 00 56 52 4D AF
( ) ODA0 4D AF 4D FC 06 AF FC 02 FE FF 2F 29 26 24 2F 29
( ) ODB0 A4 32 A9 FC 06 AF FC 02 FE 00 56 52 4D AF 4D AF
( ) ODC0 4D FC 06 AF FC 02 FE FF 39 40 44 39 2F A4 29 2F
( ) ODD0 39 A9 80 80 FE 00 56 52 4D AF 4D AF 4D FC 06 AF
( ) ODE0 FC 02 FE FF 2F 29 26 24 2F 29 A4 32 A9 AF 80 80
( ) ODF0 2F 29 24 2F 29 A4 2F 29 2F 24 2F 29 A4 2F 29 2F
( ) OEO0 24 2F 29 A4 32 A9 AF 80 80 FA FF FF FF FF FF FF
```

Codierung der Melodien

| Steuercode | Wirkung |
|------------|--|
| FB | Geschwindigkeit ändern: 18 = schnell, 60 = langsam, 30 = normal. |
| FC | Dauer „langer“ Noten: 2 = doppelt so lang wie „kurze“ Note. |
| FD | Oktave setzen: 2 = Bass, 4 = tiefer Bass (normal = 01). |
| FE | Instrument setzen: FF = Piano, 00 = Klarinette. |
| FA | Ende einer Melodie. |
| FF | Ende aller Melodien, Rücksetzen auf die erste. |

| Ton | Toncodierung | | | |
|-------|--------------|------|------------|------------|
| | tief | hoch | Hex (kurz) | Hex (lang) |
| A | 75 | F5 | 39 | B9 |
| A# | 6E | EE | 35 | B5 |
| B | 68 | E8 | 32 | B2 |
| C | 62 | E2 | 2F | AF |
| C# | 5C | DC | 2C | AC |
| D | 56 | D6 | 29 | A9 |
| D# | 52 | D2 | 26 | A6 |
| E | 4D | CD | 24 | A4 |
| F | 48 | C8 | 22 | A2 |
| F# | 44 | C4 | 20 | A0 |
| G | 40 | C0 | 1E | 9E |
| G# | 3C | BC | 1C | 9C |
| Pause | 00 | 80 | | |

D/A-Wandlung per Software

In vielen Anwendungsfällen ist es nötig, daß ein vom Mikrocomputer errechneter Wert als analoge Spannung ausgegeben wird, z. B. um die Frequenz und Laut-

stärke eines Tongenerators zu steuern oder die Helligkeit einer Beleuchtung zu regeln. Nicht immer kommt es dabei auf eine hohe Digital-Analog-Umsetzge-

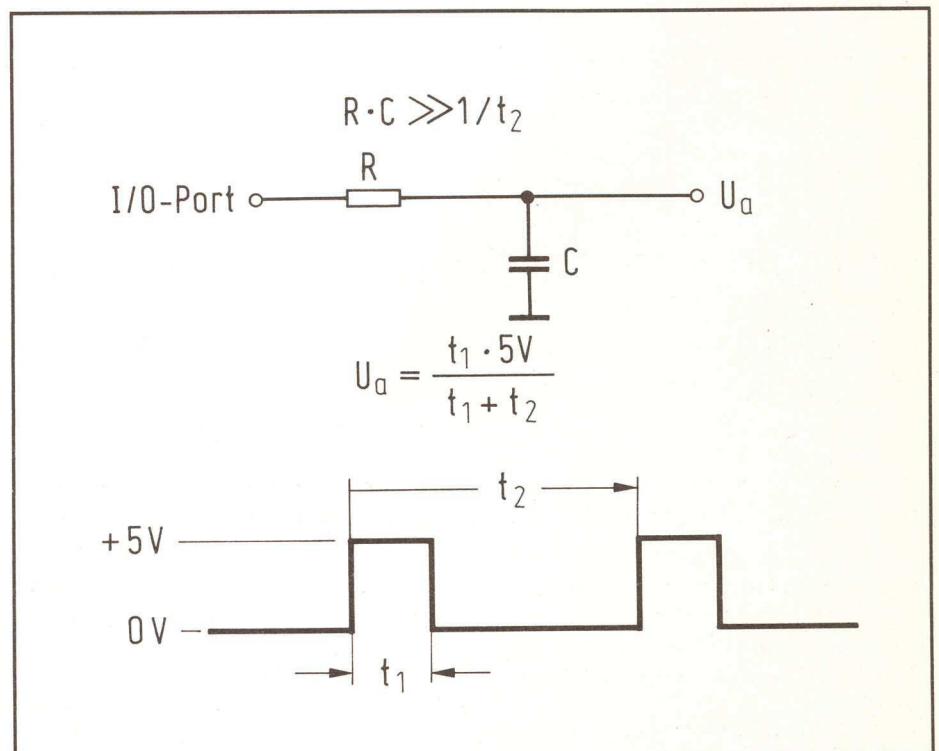


Bild 1. Prinzip der Software-D/A-Umsetzung

```

PASS 1
PASS 2
A406
0000
0000 ;D/A CONVERTER
0000 U =0 ;VALUE
0000 CL64 =$A49E ;(081E)
0000 PBD =$A002 ;(0803)
0000 PB =$A000 ;(0802)
0000 IRQV =$A404 ;(0FFE)
0000 *= $010C
010C 4C000C JMP START
010F *= $C00
OC00 START A901 LDA £1 ;START
OC02 8D9EA4 STA CL64 ;TIMER
OC05 A908 LDA £8 ;PB3=
OC07 8D02A0 STA PBD ;OUTP.
OC0A 58 CLI
OC0B 60 RTS
OC0C IRQ 48 PHA
OC0D A500 LDA U ;U=0?
OC0F FO13 BEQ HI ;YES
OC11 AD00A0 LDA PB
OC14 2908 AND £8 ;PB3=H?
OC16 DOOC BNE HI ;YES
OC18 AD00A0 LDA PB
OC1B 0908 ORA £8 ;NO, MAKE
OC1D 8D00A0 STA PB ;IT H
OC20 A500 LDA U ;T=T1
OC22 DO11 BNE TI ;JUMP
OC24 HI AD00A0 LDA PB ;MAKE
OC27 29F7 AND £$F7 ;PB3=L
OC29 8D00A0 STA PB
OC2C A900 LDA £0 ;T2=256
OC2E 38 SEC ;T=
OC2F E500 SBC U ;T2-T1
OC31 DO02 BNE TI
OC33 A9FF LDA £$FF ;AVOID
OC35 TI 8D9EA4 STA CL64 ;T=0
OC38 68 PLA
OC39 40 RTI
OC3A *=IRQV
A404 OCOC .WOR IRQ
A406 .END
A406 ERRORS= 0000
    
```

Bild 2. Assemblerlisting für den AIM-65; EMUF-Adressen in Klammern

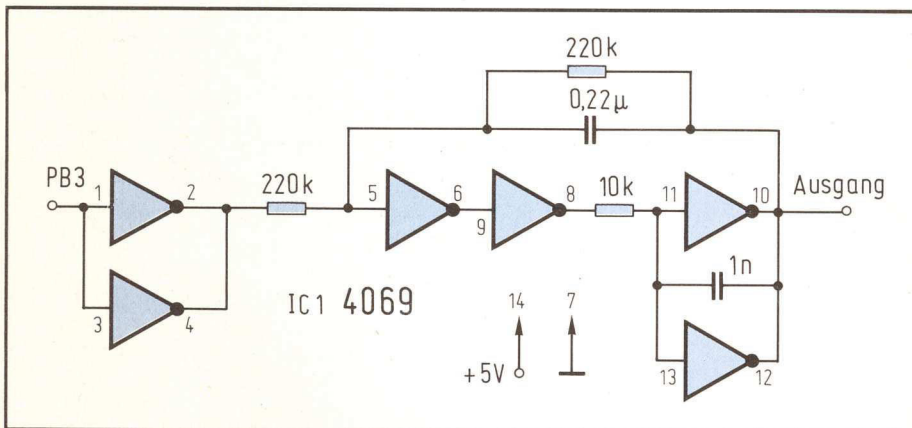


Bild 3. Verbesserte Tiefpaß-Schaltung mit niederohmigem Ausgang

schwindigkeit und auf hohe Genauigkeit an: Oft genügt es, wenn der Analogwert in etwa einer halben Sekunde stabil anliegt und auf zwei oder drei Prozent genau ist. In solchen Fällen kann man auf den relativ teuren D/A-Wandler-Baustein verzichten und dem Mikrocomputer selbst

einen Großteil des Problems aufhalsen. Die gewünschte Analogspannung gewinnt man über einen einfachen Tiefpaß, an dessen Eingang eine Rechteckspannung mit veränderlichem Tastverhältnis angelegt wird. Diese Rechteckspannung kann der Computer an einem I/O-Port per Interruptprogramm erzeugen.

Das parallel laufende Hauptprogramm wird dann von der D/A-Routine nicht behindert und übergibt dieser den auszugebenden Wert in einer Speicherzelle. Bild 1 veranschaulicht das Prinzip, und Bild 2 zeigt ein Assemblerlisting für den AIM-65 mit den EMUF-Adressen zum Vergleich in Klammern. Als Interruptquelle dient der Timer im Baustein 6532, der sowohl im AIM-65 als auch im EMUF vorhanden ist. Der Start des AIM-Programms erfolgt über den Vektor bei 010C mit der Funktionstaste F1. Hierbei wird lediglich der periodische Timer-Interrupt initialisiert, der bis zum Drücken der Reset-Taste weiterläuft und den Inhalt der Speicherzelle 0000 (die man jetzt mit dem Monitorprogramm-Befehl „/“ ändern kann) als Analogwert über PB3 ausgibt. Der Tiefpaß in Bild 1 ($R=100\text{ k}$, $C=0,47\text{ }\mu\text{F}$) ist leider nur für den Anschluß hochohmiger Meßgeräte geeignet; ferner liefert der I/O-Port auch nicht den vollen Spannungshub von 5 V. Deshalb zeigt Bild 3 eine verbesserte Schaltung mit einem Sechsfach-Inverter-IC, die ebenfalls nur Pfennige kostet, aber einen niederohmigen Ausgang besitzt. Die Dimensionierung ist für eine Rechteckfrequenz von rund 60 Hz ausgelegt, wie sie das Programm in Bild 2 liefert. Fe.

Programmierte EPROMs für den EMUF

Für eine Reihe der in diesem Heft beschriebenen EMUF-Applikationen, so z. B. für das V24-Interface, für das IEC-Bus-Strichcode-Interface und andere Anwendungen, sind von den Firmen r+r (Heidelberg), Wirth (Remshalden) und Elektronikladen (Detmold), die übrigens auch EMUF-Bausätze liefern, fertig programmierte EPROMs vom Typ 2716 erhältlich. Für diese EPROMs ist nur eine Versorgungsspannung von 5 V erforderlich. Bitte richten Sie Anfragen bezüglich Lieferbarkeit und Preisen direkt an diese Firmen. Sollten Sie selbst über eine Möglichkeit zum Programmieren von EPROMs verfügen und Ihre Bekannten dadurch mit EMUF-EPROMs versorgen können, so ist dies mit Quellenangabe von Hardware (EMUF: mc 1981, Heft 2) und des jeweiligen Programms gern gestattet. Fe.

Dipl.-Ing. Rolf-Dieter Klein

Mini-Datenlogger

Vielfach werden Kleinrechner zur Steuerung von Anlagen o. ä. eingesetzt. Mit Hilfe des EMUF ist es möglich, eine 32-Bit-Ausgabebeeinrichtung zu schaffen, die über den IEC-Bus an einen CBM-Rechner angeschlossen werden kann.

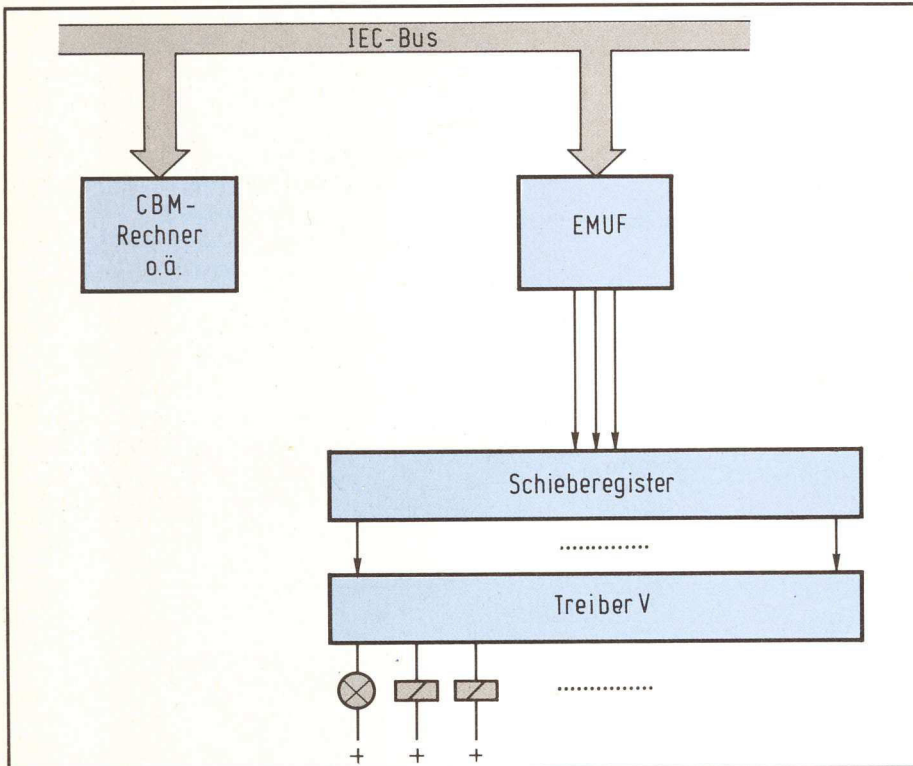


Bild 1. Anschlußbeispiel für den Mini-Logger

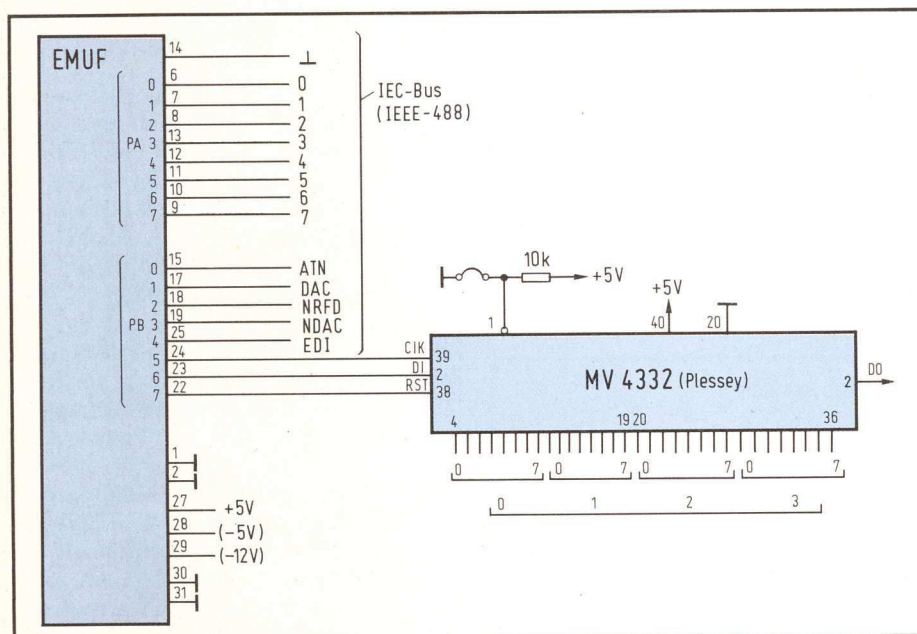


Bild 2. Die Steckerbelegung des EMUF

Da der EMUF zu wenig Anschlüsse besitzt, um eine ausreichende Anzahl von Steuerleitungen zur Verfügung zu stellen, muß er erweitert werden. Dies geschieht mit Hilfe eines Schieberegisters, in das seriell die Daten gesendet werden. Als Schieberegister wird hier eine interessante Version mit 32 Ausgängen verwendet, die ursprünglich für LCD-Anwendungen gedacht war. Bild 1 zeigt das Zusammenspiel mit den CBM. Das Schieberegister wird nur über drei Leitungen bedient; daher kann an dieser Stelle auch eine längere Leitung eingebaut werden, so daß die Steuersignalleitungen erst „vor Ort“ mit kurzen Leitungen verlegt werden müssen. Da das Schieberegister in CMOS-Technik aufgebaut ist, müssen zur Ansteuerung von Relais usw. Treiber dazugeschaltet werden. Als Schieberegister kann natürlich auch eine TTL-Lösung gewählt werden, dadurch ändert sich normalerweise nichts an der Software. Bild 2 zeigt die Beschaltung des EMUF. Dabei kann über eine Brücke noch die Polarität der Ausgangssignale des Schieberegisters eingestellt werden. Bild 3 zeigt die Software. Der IEC-Bus Teil ist bereits aus früheren mc-Heften bekannt. Die Primäradresse ist programmierbar; es wird die erste auf dem IEC-Bus liegende Adresse verwendet. An der entsprechenden Stelle im Programm ist Platz für zwei INC-Befehle, um ggf. die um zwei erhöhte Adresse programmieren zu können, falls die beiden früheren IEC-EMUFs [1, 2] ebenfalls am Bus liegen. Das Programm läßt sich auch leicht auf mehr als 32 Bits erweitern, wenn entsprechend mehr Zwischenspeicher verwendet werden. Bild 4 zeigt die Programmierung vom CBM aus. An den EMUF werden drei Parameter übergeben:

PRINT # 1, a, b, c

a gibt die logische Portnummer an. Die 32 Bits des Schieberegisters werden in vier Gruppen zu 8 Bits eingeteilt. Mit a wird die Nummer der Gruppe angegeben, mit b wird das Bit innerhalb der Gruppe angegeben, und c gibt den Wert an, den das Bit annehmen soll. Der Bereich von a geht von 0 bis 3, b kann Wert von 0 bis 7 annehmen und c kann 0 oder 1 sein. Ein Beispiel:

PRINT # 1, 2, 4, 1


```

00A6 48      PHA      ;RETTEN
00A7 AD0208 LDA      PHA
00A8 850A    PB      FLAG
00A9 A908    STA      %00001000
00AA 800208 STA      %00001000
00AB AD0208 LDA      LOPB:
00AC 2902    BEQ     LOPB
00AD F0F9    BEQ     LOPB
00AE A900    LDA      %00000000
00AF 800208 STA      %00000000
00B0 68      PLA
00B1 60      RTS

00BF AD0208 ; TALKON: LDA
00C0 2901    AND     %00000001
00C1 F0F9    BEQ     TALKON
00C2 A912    LDA      %00010010
00C3 800208 STA      %00010010
00C4 89FF    LDA      %11111111
00C5 800108 STA      %01010010
00C6 A952    LDA      %00010010
00C7 800308 STA      %00010010
00C8 A912    LDA      %00010010
00C9 800208 STA      %00010010
00CA 60      RTS

00CB A900    ; TALKOFF: LDA
00CC 800108 STA      %00000000
00CD A900    LDA      %00000000
00CE 800208 STA      %00000000
00CF A94C    LDA      %01001100
00D0 800308 STA      %00000000
00D1 A900    LDA      %00000000
00D2 800208 STA      %00000000
00D3 60      RTS

00D4 48      ; SEN01: PHA
00D5 201600 LDA      SEN01:
00D6 AD0208 STA      SEN01:
00D7 2904    BEQ     SEN01:
00D8 68      PLA
00D9 49FF    EOR     SEN01:
00DA 800008 STA      SEN01:
00DB A910    LDA      SEN01:
00DC 800208 STA      SEN01:
00DD 201600 LDA      SEN01:
00DE AD0208 STA      SEN01:
00DF 2908    BEQ     SEN01:
00E0 F0F6    BEQ     SEN01:
00E1 A912    LDA      SEN01:
00E2 800208 STA      SEN01:
00E3 60      RTS

00E4 60      ; CKRATN: RTS
00E5 60      ; SEN01: PHA
00E6 201600 LDA      SEN01:
00E7 AD0208 STA      SEN01:
00E8 F0F6    BEQ     SEN01:
00E9 68      PLA
00EA 49FF    EOR     SEN01:
00EB 800008 STA      SEN01:
00EC A910    LDA      SEN01:
00ED 800208 STA      SEN01:
00EE 201600 LDA      SEN01:
00EF AD0208 STA      SEN01:
00F0 2904    BEQ     SEN01:
00F1 F0F6    BEQ     SEN01:
00F2 A912    LDA      SEN01:
00F3 800208 STA      SEN01:
00F4 60      RTS

00F5 60      ; CONNSE: PHA
00F6 201600 LDA      CONNSE:
00F7 AD0208 STA      CONNSE:
00F8 2908    BEQ     CONNSE:
00F9 F0F6    BEQ     CONNSE:
00FA A912    LDA      CONNSE:
00FB 800208 STA      CONNSE:
00FC 60      RTS

00FD 60      ; CKRATN: RTS
00FE 60      ; SEN01: PHA
00FF 201600 LDA      SEN01:
0100 AD0208 STA      SEN01:
0101 2904    BEQ     SEN01:
0102 F0F6    BEQ     SEN01:
0103 A912    LDA      SEN01:
0104 800208 STA      SEN01:
0105 60      RTS

0106 60      ; MAIN: PHA
0107 209600 LDA      MAIN:
0108 8508    STA      %033 8508
0109 60      RTS
    
```

```

00B7 C920      CMP      #20      ;FALLS BLANK
00B9 F0EF      BEQ      LM3      ;IGORIEREN
00BB C92C      CMP      #2C      ;AUCH KOMMA
00BD F0EB      BEQ      LM3
00BF 2901      AND      #%00000001 ;NUR 0,1 ERLAUBT
00C1 D00D      BNE      ORM      ;WENN = 1 DANN ODERN
00C3 98        TYA      ;HOLE WERT
00C4 49FF      EOR      #%11111111 ;COMPLEMENTIEREN
00C6 3510      AND      VAL1,X    ;IN X IST 0..3
00C8 9510      STA      VAL1,X    ;NEUER WERT
00CA 20750C    JSR      BITSET   ;UND BELEGEN
00CD 4C300D    JMP      MAIN     ;FERTIG
00D0          ORM:
00D0 98        TYA
00D1 1510      ORA      VAL1,X    ;HIER ODERN NOETIG
00D3 9510      STA      VAL1,X    ;NEUER WERT
00D5 20750C    JSR      BITSET
00D8 4C300D    JMP      MAIN
;
;
;
0000          END

```

---- SYMBOL TABLE ----

| | | | | | |
|--------|------|--------|------|--------|------|
| B1 | 0C6D | LPPB | 0C63 | SKM1 | 0DA9 |
| BITSET | 0C75 | MAIN | 0030 | TALKOF | 0C08 |
| BYTA | 0C5D | MAIN1 | 0035 | TALKON | 0CBF |
| CKATN | 0D16 | MDE | 000E | TALPA | 004A |
| CNT | 000F | NULA | 0C35 | TIM1 | 0814 |
| CONSE | 0006 | ORM | 00D0 | TIM64 | 0816 |
| COUNT | 000C | PA | 0800 | TIM8 | 0815 |
| COUV | 0014 | PAD | 0801 | TIMFLC | 0817 |
| EINSA | 0C48 | PADR | 000D | TIMIN | 0816 |
| FLAG | 000A | PB | 0802 | TMP1 | 0015 |
| GETCHA | 0C90 | PBD | 0803 | VAL1 | 0010 |
| INIA | 0C25 | RESET | 0C00 | VAL2 | 0011 |
| LISTPA | 0D4D | S1 | 0C70 | VAL3 | 0012 |
| LM1 | 0D6E | SEND | 0CF1 | VAL4 | 0013 |
| LM2 | 0D86 | SEND1 | 0CF0 | XTEMP1 | 0000 |
| LM3 | 0DAA | SENE01 | 0D17 | ZCOU | 0009 |
| LOPA | 0C95 | SENE0I | 0D18 | ZEICH | 0008 |
| LOPB | 0CB1 | SK2 | 0D66 | ZOUT | 0008 |
| LPI | 0DA1 | SK3 | 0D6E | | |

Das Bit 4 der zweiten Gruppe wird auf 1 gesetzt, also Bit 20.

PRINT # 1, 1, 2, 0

Bit 2 der ersten Gruppe wird auf 0 gesetzt, damit Bit 10. Einen Nachteil hat

die Lösung mit dem Schieberegister MV 4332 natürlich: Während des Hineinschiebens gibt es kurzzeitig Störungen an den Ausgängen. Dies läßt sich mit einem Zwischenspeicher an den Ausgängen beseitigen. Wird das CLR-Signal freigelassen und statt dessen ein Latch-

Bild 4.
Programmierbeispiel für den CBM 8032

```

5  REM Test aller Bits
10 OPEN 1,8
20 FOR I=0 TO 3
30 FOR J=0 TO 7
40 FOR W=0 TO 1
50 PRINT#1,I,J,W
60 NEXT W
70 NEXT J
80 NEXT I

```

Signal für ein Schieberegister verwendet, so funktioniert das Ganze ohne Störungen. In der Software ist Platz an der entsprechenden Stelle für einen Unterprogrammssprung nach INIA, so daß der ursprüngliche CLR-Puls als Strobe nun nach der Ausgabe gegeben werden kann, und die Daten übernommen werden. Die Schaltung eignet sich ohne das zusätzliche Latch insbesondere für das Ein- und Ausschalten von Lampen, Relais usw. wo die kurzen Störungen nichts ausmachen.

Literatur

- [1] Klein, Rolf-Dieter: EMUF bringt Strichcode zum IEC-Bus. mc 1981, Heft 3.
- [2] Klein, Rolf-Dieter: V.24-Interface. mc 1981, Heft 4.

Wenn der EMUF streikt

Die Fehlersuche ist bei Mikrocomputern meist erheblich schwieriger als bei leichter durchschaubaren konventionellen Digitalschaltungen. Die folgende Checkliste wurde aufgestellt, um typische Probleme schneller auffinden zu können.

Bei selbstgeschriebenen Programmen:

- Wurde eventuell der Reset-Vektor (OFFC) oder, wenn vom Interrupt Gebrauch gemacht wird, der IRQ-Vektor (OFFE) vergessen?
- Wird der Stackpointer am Programm-anfang initialisiert (LDX #\$\$F, TXS)?

- Werden am Programm-anfang das Interrupt-Flag mit SEI/CLI und das Dezimalflag mit CLD/SED korrekt gesetzt?
- Werden am Programm-anfang die Portregister korrekt belegt (PAD/PBD bei 0801/0803)?

Bei Programmen im EPROM:

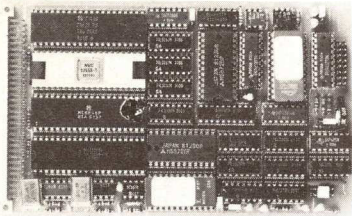
- Wird bei 2-KByte-EPROMs das richtige KByte angewählt (normalerweise untere Hälfte: Masse an Pin 29 der 32poligen Steckleiste)?
- Ist die Betriebsspannung zwischen 4,8 V und 5,2 V und der Stromver-

brauch zwischen 200 mA und 280 mA?

- Ist an Pin 6 des Bausteins 7400 ein 1-MHz-Taktsignal mit TTL-Pegel festzustellen?
- Arbeitet der Reset-Timer 555, d. h. geht beim Einschalten der Versorgungsspannung Pin 21 der 31poligen Steckleiste mit einer Verzögerung von etwa 0,3 s auf High-Pegel?

Noch ein Hinweis: Die im Bestückungsplan ganz links befindlichen vier Kondensatoren können bei Verwendung des EPROM-Typs 2716 entfallen. Fe.

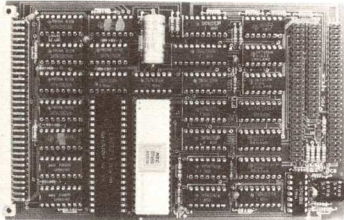
Die intelligente Terminalkarte



Video 4

- Videosystem mit eigener Z-80-CPU
- Bildformat: 80 Zeichen/Zeile, 24 Zeilen/Bild
- Vielfältige Funktionen, z. B. frei adressierbarer Cursor
- Beliebige Zeichensätze inkl. Grafik, z. B. auch APL
- Attribute: Invertieren, Blinken, Hellschrift u. Farbsteuersignale
- Tastatur u. Druckeranschluß – und vieles mehr

Die Verbindung vom Rechner zur Floppy



FDC 5-8

- Floppy-Controller für ECB- oder Elzet-Bus
- BASF-, Shugart- od. Philips-Laufwerke
- Single Side und Double Side
- Single Density und Double Density (IBM-Format)
- DMA Mode od. Polling Mode
- Z-80A-DMA-Controller auf der Karte

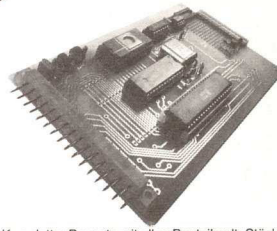
Computer Elektronik GEORG KRAUSE

Zum Römergrund 59 · 6501 Wörrstadt · Telefon (0 67 32) 41 78

r+r electronic

Versandanschrift:

Adlerstraße 55, 6900 Heidelberg 1, Tel. 0 62 21/78 15 00



6504 Computer-BS nach MC DM 89.-

Kompletter Bausatz mit allen Bauteilen lt. Stückliste (jedoch ohne EPROM), TI-Flachfassungen für alle ICs (auch EPROM) sowie 31pol. Stiftleiste.

Komplettpreis DM 89.- Platine einzeln DM 27.50

Unsere EMUF-Platine ist aus Epoxyd, durchkontaktiert mit Bestückungsaufdruck und mit Lötstoplack versehen.

Alle Programme aus EMUF-Sonderheft oder früheren MC-Ausgaben. In jedem EPROM befinden sich 2 Programme (ab 000H und 400H)

2716-MC 1 ... IEC/V 24 (MC 4/81) und IEC/BARCODE (MC 3/81) DM 19.50

2716-MC 2 ... Standardschnittstellen für Schreibmaschinen (seriell und parallel) aus Sonderheft DM 19.50

2716-MC 3 ... Türklingel/Alarmanlage aus Sonderheft DM 24.50

2716-MC 4 ... V 24-Schnittstellentester u. Interface für Praxis 30 aus Sonderheft DM 19.50

2716-MC 5 ... Funkfernempfänger und DCF-77 Decoder aus Sonderheft DM 19.50

2716-MC 6 ... Mini-Datenlogger und LCD-Display aus Sonderheft DM 19.50

2716-MC 7 ... Whisky-EMUF und Türklingel/Alarmanlage aus Sonderheft DM 19.50

Z80-Einplatinencomputer nach MC

Kompletter Bausatz mit allen Teilen lt. Stückliste, jed. ohne EPROM, TI-Flachfassungen für alle ICs (auch EPROM) inkl. Platine mit Bestückungsdruck und beidseitigem Lötstoplack.

Komplettpreis DM 225.-

Platine einzeln DM 79.-

Netzteilbausatz für Z80-Computer DM 59.-

inkl. sämtlicher Bauteile, auch Schrauben und Lötlötgel, und Platine mit Bestückungsdruck und Lötstoplack.

Netzteilplatte einzeln DM 22.50

Testprogramm in EPROM 2716 DM 29.50

Das EPROM besitzt 4 Testprogramme, mit denen alle Funktionen des Z80-Einplatinencomputers überprüft werden können. Das entsprechende

Testprogramm wird mit der Stellung des DIL-Schalters selektiert. Eine ausführliche Beschreibung wird mitgeliefert.

Aktuelle ICs

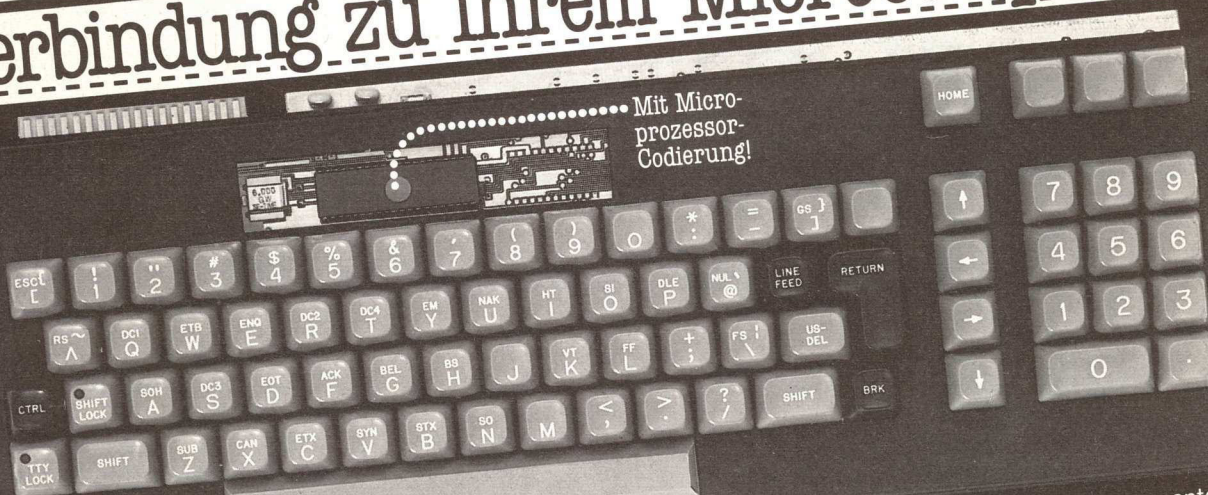
6504 22.50 8255, 8251 je 15.60

Z80-CPU 19.50 EPROM 2716 15.60

Ladenverkauf: Breslauerstr. 29, 6900 HD-Kirchheim
Geschäfts. Mo.-Fr. 9-13, 14-18, Sa. 9-13. Preise inkl. MwSt.
Versand per Nachnahme ab DM 30.- + Versandkosten
Sendungen ins Ausland nur per Vorausrechnung
Preisänderungen und Zwischenverkauf vorbehalten.

NEU

Die vielseitige Verbindung zu Ihrem Microcomputer



Hier ist sie:

Tastatur ET 110. Von Eurokey.
Vorzüglich geeignet für
Programmieranwendungen.

Technisch vielseitige Standardausstattung:

- +5V als einzige Versorgungsspannung
- TTL-Ausgänge
- Vergoldete Anschlüsse für Platinen-Stecker
- Wählbar durch Lötbrücke: N-Key oder 2-Key rollover, automatische Repeat-Funktion oder Repeat-Funktionstaste, gerade oder un-

gerade Parität, positiver oder negativer Ausgang. ● ASCII-Codierung mit Alpha-Lock (TTY-Lock) ● 2-farbig gespritzte Tastenkappen.

Und auf Wunsch:

- Kundenspezifische Software
- Andersfarbige Tastenkappen
- Spezielle Schriftzeichen und Symbole
- Verschiedene Federstärken.

Diese hochinteressante
Tastatur ist ab Lager
lieferbar. Auch in Stück-
zahlen.

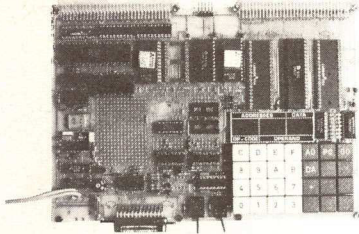
Ergänzende Informationen
direkt durch Eurokey.

eurokey

Tastaturen GmbH

Ravensburger Str. 128-134 · D-7981 Berg
Krs. Ravensburg · Tel. (07 51) 2 63 01
Telex 7-32 866

BETA 65



Der erste wirklich universell einsetzbare Single-Board-Computer!

BETA 65 ist ein äußerst preisgünstiges System für viele Anwendungen – vom Selbststudium (Lehrsystem) bis zur Prozeßsteuerung:

- mit dem am weitesten verbreiteten Prozessor **6502**
- bis zu **52 I/O-Leitungen** auf der Platine
- extrem leistungsfähiger Monitor (4 K)
- Hex-Assembler und -Editor, 2-K-RAM
- Kassetten-Interface und RS-232
- erweiterbar (u. a. mit BASIC)
- preisgünstig: **DM 598.-** (Bausatz DM 549.-) inkl. MwSt.

WOLFRAM FEISE MICROPROZESSORTECHNIK

Alte Zeche 2, D-3013 Barsinghausen 4
Postfach 15, Tel. (0 51 05) 6 29 27

EMUF-1-Platinen-Computer

Leerplatine DM 27.50/1, 77.-/3, 244.-/10 ● Bausatz incl. 31pol. Stifteleiste 89.-/1, 249.-/3 Fertiggerät, betriebsbereit getestet a. Anfrage ● Programmierte EPROMs 2716/5 V mit je 2 Programmen für Applikationen nach diesem Bauheft DM 22.50/1, 59.-/3 (verschiedene) 16 Einzeltasten DM 22.50, HEX-Tastenfeld a. A. ● LCD-Anzeige Siemens LCM 1010 Tagespreis anfragen! Schieberegister Plessey MV 4332 Tagespreis anfragen! Stifteleiste 31polig, vers. 5.-/1, 13.50/3, 38.-/10 Federleiste 31polig, vers. 6.50/1, 18.50/3, 49.-/10

6502 Regge-1-Karten-Computer

nach mc 2/82, S. 26
Leerplatine DM 73.-/1, 209.-/3, 657.-/10 ● Bausatz, komplett (ohne EPROM) 268.-, Teilsatz (ICs, Fassungen, Quarz, Platine) 235.-, Mengenrabatte. Fertiggerät, betriebsbereit getestet, auf Anfrage. Testprogramm + zusätzliche Funktionen: Kuckucksuhr, Morsezeichen, LED-Zeit Anzeige, mit kommentiert. Listing und HEX-Dump, EPROM 2716, DM 39.-/1 Federleiste 44polig, vergoldet 15.50/1, 39.-/3

6502/I-1, 1-Platinen-Computer (Industrie-System)

bietet die Möglichkeit zum Sandwich-Aufbau mit anderen Karten oder kann als Rückwandverdrahtung (Motherboard) für einen 19"-Teileinschub verwendet werden. I/O- und andere Karten werden dann senkrecht aufgesteckt. Dieser Computer bietet 60 I/O-Leitungen, der Bus ist herausgeführt. Die Adressen sind voll dekodiert. Bestückung: 2 x 6520, 1 x 6522, 6502, 2732/5 V, 1 K RAM, Quarz. Erweiterung auf Zusatzkarte möglich. **...Prospekt anfordern!...** Leerkarte DM 85.-/1, 212.-/3, 630.-/10, 4300.-/100 (incl. MwSt.) ● Bausatz DM 240.-/1, Fertiggerät, getestet DM 360.-. Relais-Karten, OPTO-Karten, ADW/DAW bitte anfragen.

Z 80-1-Platinen-Computer

Platinen, Bausätze, Fertiggeräte: bitte anfragen.

EPROM-Burner Bausätze + Fertiggeräte: bitte anfragen.

EPROM-Löschgerät L-UV-4

profess. Löschgerät für gleichzeitiges Löschen von 4 ICs in ca. 20 Minuten. Ozonfrei! Mit automatischer Abschaltung (10...30 min) 289.-. Ohne Abschaltung DM 239.-. Ersatzbrenner + Vorschaltgerät (f. Selbstbau) DM 149.-

Datencassetten

Spezial-Mechanik, 5fach geschraubt, Federdruck, mit BASF-Band TP 18 LH, C 10: 27.-/10 St., C 20: 29.-/10 St. Hohe Mengenrabatte ab 20 Stück.

17%

Bis 17% günstiger liefern wir jetzt **ATARI-Computer**, nur **DM 298,-** kostet bei uns der **grüne 12"-Monitor** ZVM 121 E. Konkurrenzlose Preise für OKI microline 80 = DM 1285,-, Seikosha GP 100 A = DM 1059,- und andere

Wir führen weitere Computer, Drucker, Peripherie und Software zu günstigen Konditionen. Die Preise enthalten die MwSt. – Bitte fragen Sie bei größerem Bedarf die günstigeren Preise an. – Händleranfragen erwünscht.

Wir liefern nicht nur Standard-Hard-/Software.

Wir bieten ...

...komplette Anlagen für die Meßwert erfassung an Maschinen/Anlagen, automatische Testeinrichtungen für Serienteile. Wir lösen spezifisch Ihre Probleme aus den Bereichen Messen · Prüfen · Steuern · Regeln · Überwachen. Vielleicht haben wir Ihr Problem in dieser oder ähnlicher Form bereits gelöst! – Fragen Sie an.

Wirth Elektronik D-7064 Remshalden
Tel. 0 71 51/7 12 26



Flache Tastaturen.

Eine ergonomisch begründete Forderung für Bildschirmarbeitsplätze. Voraus-

setzung: Einbautaster mit extrem niedriger Bauhöhe, also RAFI-System RS 74 und RS 76 mit zusätzlichen ergonomisch wichtigen Pluspunkten: Blendfreie, mattierte Tastenoberfläche, dunkle Zeichen auf hellem Grund, Fingernagelschutz.

Die RAFI-Systeme RS 74 und RS 76 stehen sowohl als Einzeltaster wie auch als Tastaturen zur Verfügung,

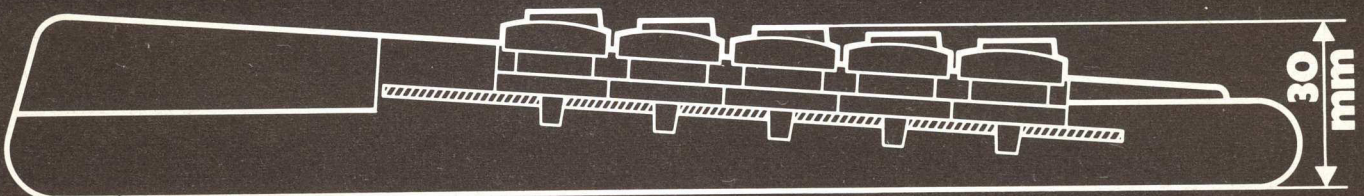
Den Flachen gehört die Zukunft

auch im formschönen Flachgehäuse. Sie entsprechen bereits heute einer Notwendigkeit der Zu-

kunft: Die Bauhöhe der Tastatur, gemessen an der Tastaturmittlereihe (C-Reihe nach DIN 2137) beträgt nicht mehr als 30 mm. RAFI – ein großes Leistungsspektrum. Von der Schaltungsentwicklung bis zum kompletten elektronischen System. Heute mehr denn je:

Planen Sie mit uns. Wir haben die Experten und die Erfahrung.

RAFI GmbH & Co. · Elektrotechnische Spezialfabrik
Postfach 2060 · 7980 Ravensburg · Telefon 07 51 / 890 · Telex 07 32 866



Herwig Feichtinger

EMUF morst Rufzeichen

Bei Funkamateuren besteht zum Teil die Notwendigkeit, das eigene Rufzeichen automatisch an bestimmte Aussendungen anzuhängen, z. B. bei Funkfern schreiben. Dabei kann die Betriebsart Morsetelegrafie auch von Inhabern der Lizenzklasse C verwendet werden.

Die Tabelle nennt die Belegung der I/O-Ports. Es können bis zu 16 unterschiedliche Texte gespeichert werden; welcher gerade gesendet wird, bestimmen die Pe-

gel an PB 0...3. Mit PB 4 lassen sich zwei Morsegeschwindigkeiten wählen. An PB 7 steht ein 800-Hz-Nf-Ausgangssignal zur Verfügung, und die Starttaste

kann man von PB 6 nach Masse legen. Besteht eine feste Verbindung von PB 6 nach Masse, so wird der Text ständig wiederholt. Die Wiederholzeit läßt sich mit PA 0...1 in vier Stufen wählen, hier etwa 0/0,5/2/8 Minuten.

An PB 5 steht ein Signal zur Verfügung, das stets während der Textaussendung auf High-Pegel liegt, z. B. zur automatischen Steuerung des Senders. Es setzt etwas vor dem ersten Ton ein, um eine

Programmlisting mit Beispieltexen. Das EPROM ist hier natürlich „kundenspezifisch“ ab 0D04

```

0000      ;MORSE GENERATOR
0000 FREQ      =10
0000 SPD1      =110
0000 SPD2      =70
0000 PNT      *=*+2
0002 TIL      *=*+1
0003 TIH      *=*+1
0004 CHR      *= $FFC
0FFC      000C  .WOR RES
0FFE PA      = $800
0FFE PB      = $802
0FFE PBD      = $803
0FFE CL64     = $816
0FFE CLK      = $817
0FFE      *= $C00
0C00 RES     A2FF  LDX  #$FF
0C02      9A    TXS
0C03      78    SEI
0C04      D8    CLD
0C05      A9A0  LDA  #$A0      ;PB7=AF
0C07      8D0308 STA PBD      ;PB5=TX
0C0A      A900  LDA  #0
0C0C      8D0208 STA PB       ;TX OFF
0C0F      F029  BEQ  START
0C11 WAIT    AD0008 LDA  PA
0C14      2903  AND  #3        ;PA0-1
0C16      AA    TAX           ;FOR REPEAT
0C17      BDC00C LDA  RPT,X    ;TIME
0C1A      8503  STA  TIH
0C1C      A901  LDA  #1
0C1E      8D0208 STA PB       ;TX OFF
0C21      8502  STA  TIL
0C23 WAIT0   A962  LDA  #9B
0C25      8D1708 STA  CLK
0C28 WAIT1   2C1708 BIT  CLK
0C2B      10FB  BPL  WAIT1
0C2D      2C0208 BIT  PB
0C30      7008  BVS  START
0C32      C602  DEC  TIL
0C34      D0E1  BNE  WAIT0
0C36      C603  DEC  TIH
0C38      D0E9  BNE  WAIT0
0C3A START   2C0208 BIT  PB      ;PB6KEY
0C3D      70FB  BVS  START
0C3F      A920  LDA  #$20      ;TX ON
0C41      8D0208 STA  PB
0C44      A904  LDA  #<TXT     ;SET PNT
0C46      8500  STA  PNT      ;TO BEGIN
0C48      A90D  LDA  #>TXT     ;OF TEXT
0C4A      8501  STA  PNT+1
0C4C      AD0208 LDA  PB       ;PB0-3
0C4F      290F  AND  #$F       ;TO SELECT
0C51      0980  ORA  #$80      ;TEXT
0C53      A000  LDY  #0
0C55 STXT    D100  CMP  (PNT),Y
0C57      08    PHP           ;SAVE Z FLAG
0C58      20930C JSR  INCP
0C5B      28    PLS
0C5C      D0F7  BNE  STXT
0C5E      A214  LDX  #20      ;DELAY
0C60      D002  BNE  SPC
0C62 NXT     A202  LDX  #2      ;PAUSE
0C64 SPC     18    CLC           ;BETW. CHR'S
0C65      209A0C JSR  ELEM      ;Y=0
0C68      B100  LDA  (PNT),Y
0C6A      30A5  BMI  WAIT
0C6C      20930C JSR  INCP
0C6F      38    SEC
0C70      E920  SBC  #$20
0C72      D004  BNE  CONU
0C74      A205  LDX  #5      ;IT'S A
0C76      D0EC  BNE  SPC      ;SPACE
0C78 CONU    AA    TAX
0C79      BDC40C LDA  CTAB,X
0C7C      8504  STA  CHR
0C7E SHFT    0604  ASL  CHR
0C80      F0E0  BEQ  NXT
0C82      A201  LDX  #1
0C84      9002  BCC  DOT
0C86      A203  LDX  #3      ;DASH
0C88 DOT     38    SEC
0C89      209A0C JSR  ELEM
0C8C      18    CLC
0C8D      EB    INX           ;1 ELEM
0C8E      209A0C JSR  ELEM      ;PAUSE
0C91      F0EB  BEQ  SHFT      ;ALW.
0C93      ;INCREMENT POINTER
0C93 INCP    E600  INC  PNT
0C95      D002  BNE  *+4
0C97      E601  INC  PNT+1
0C99      60    RTS
0C9A      ;SEND X ELEMENTS WITH
0C9A      ;AF OUTPUT IF C=1
0C9A ELEM    A06E  LDY  #SPD1
0C9C      A910  LDA  #$10
0C9E      2D0208 AND  PB       ;MASK PB4
0CA1      F002  BEQ  ELEM1     ;FOR SPEED
0CA3      A046  LDY  #SPD2
0CA5 ELEM1   2C1708 BIT  CLK
0CA8      10FB  BPL  ELEM1
0CAA      A90A  LDA  #FREQ
0CAC      8D1608 STA  CL64
0CAF      9008  BCC  QUIET
0CB1      AD0208 LDA  PB
0CB4      4908  EOR  #$80
0CB6      8D0208 STA  PB
0CB9 QUIET   88    DEY
0CBA      D0E9  BNE  ELEM1
0CBC      CA    DEX
0CBD      D0DB  BNE  ELEM

```

```

0CBF      60      RTS
0CC0      ;REPEAT TIMES 0-3
0CC0 RPT   14      .BYT 20,5,2,1
0CC1      05
0CC2      02
0CC3      01
0CC4      ;ASCII TO CW CONVERSION
0CC4 CTAB  80      .BYT $80,$36,$4A,1
0CC5      36
0CC6      4A
0CC7      01
0CC8      16      .BYT $16,$80,$44,$7A
0CC9      80
0CCA      44
0CCB      7A
0CCC      B6      .BYT $B6,$B6,$80,$80
0CCD      B6
0CCE      80
0CCF      80
0CD0      CE      .BYT $CE,$86,$56,$94
0CD1      86
0CD2      56
0CD3      94
0CD4      FC      .BYT $FC,$7C,$3C,$1C
0CD5      7C
0CD6      3C
0CD7      1C
0CD8      0C      .BYT $C,4,$B4,$C4
0CD9      04
0CDA      84
0CDB      C4
0CDC      E4      .BYT $E4,$F4,$56,$56
0CDD      F4
0CDE      56
0CDF      56
0CE0      AC      .BYT $AC,$8C,$54,$32
0CE1      8C
0CE2      54
0CE3      32
0CE4      4C      .BYT $4C,$60,$88,$A8
0CE5      60
0CE6      88
0CE7      A8
0CE8      90      .BYT $90,$40,$28,$D8
0CE9      40
0CEA      28
0CEB      D8
0CEC      08      .BYT 8,$20,$78,$B0
0CED      20
0CEE      78
0CEF      B0
0CF0      48      .BYT $48,$E0,$A0,$F0
0CF1      E0
0CF2      A0
0CF3      F0
0CF4      68      .BYT $68,$D8,$50,$10
0CF5      D8
0CF6      50
0CF7      10
0CF8      C0      .BYT $C0,$30,$18,$70
0CF9      30
0CFA      18
0CFB      70
0CFC      98      .BYT $98,$B8,$C8,$80
0CFD      B8
0CFE      C8
0CFF      80
0D00      80      .BYT $80,0,$80,$80
0D01      00
0D02      80
0D03      80
0D04      80
0D04      ;MORSE TEXT IN ASCII
0D04      ;(EXAMPLE FOR DC1YB / DK8CD)
0D04 TXT   8F      .BYT $8F,'DK8CD'
0D05      444B
0D0A      8E      .BYT $8E,'DK8CD/P',$8D
0D0B      444B
0D12      8D
0D13      444B .BYT 'DK8CD/M',$8C
0D1A      8C
0D1B      5445 .BYT 'TEST DE DK8CD'
0D28      2049 .BYT ' IN MUENCHEN',$8B
0D34      8B
0D35      5445 .BYT 'TEST DE DK8CD/P',$8A
0D44      8A
0D45      5445 .BYT 'TEST DE DK8CD/M',$89
0D54      89
0D55      444B .BYT 'DK8CD IN ATV',$88
0D61      88
0D62      5053 .BYT 'PSE QRX QRX DE DK8CD',$87
0D76      87
0D77      4351 .BYT 'CQ CQ CQ DE DK8CD',$86
0D88      86
0D89      5152 .BYT 'QRZ QRZ DE DK8CD',$85
0D99      85
0D9A      4443 .BYT 'DC1YB',$84
0D9F      84
0DA0      4443 .BYT 'DC1YB/P',$83
0DA7      83
0DAB      5445 .BYT 'TEST DE DC1YB',$82
0DB5      82
0DB6      5445 .BYT 'TEST THE QUICK'
0DC5      2042 .BYT ' BROWN FOX JUMPS'
0DD5      204F .BYT ' OVER THE LAZY'
0DE3      2044 .BYT ' DOG 1234567890'
0DF2      3C21 .BYT '<!"#< -=,?/>$'
0E00      4443 .BYT 'DC1YB IN MUENCHEN',$80
0E11      80
0E12      454D .BYT 'EMUF-CW U1.1'
0E1E      2044 .BYT ' DC1YB 8/1981'
0E2B      80 .BYT $80
0E2C      .END
0E2C      ERRORS= 0000

```

Tabelle: Portbelegung beim Morsezeichen-Generator

| Stift | Port | Belegung |
|-------|------|--|
| 1 | - | Masse, verbinden mit 2, 14, 29, 30, 31 |
| 6 | PA 0 | Wiederholzeit 8/2/0,5/0 Minuten |
| 7 | PA 1 | |
| 15 | PB 0 | Textauswahl (0...15) |
| 17 | PB 1 | |
| 18 | PB 2 | |
| 19 | PB 3 | |
| 22 | PB 7 | Nf-Ausgang ($U_{ss} = 5\text{ V}$), Ruhelage Low |
| 23 | PB 6 | Starttaste oder Schalter, aktiv Low |
| 24 | PB 5 | Steuersignal-Ausgang, aktiv High |
| 25 | PB 4 | Geschwindigkeit langsam/schnell |
| 27 | - | + 5 V, verbinden mit 28 |

Hinweis: Offene Eingänge sind automatisch auf High-Pegel.

eventuelle Einschaltverzögerung des Senders zu berücksichtigen. Die Texte sind im Programm (Bild) folgendermaßen codiert: Vor jedem Text steht ein Byte, dessen niederwertige vier Bits gleich dem Zustand von PB 0...3 sind, um den Text anzuwählen, und dessen höchstwertiges Bit High ist, um es als „Label“ von den nachfolgenden ASCII-Zeichen zu unterscheiden. Als Schlußkennzeichen steht am Ende aller Texte nochmals ein Byte hex 80. Das Programm enthält die erforderliche Konversionstabelle; sie entspricht der bereits in FUNKSCHAU 1980, Heft 9, für den AIM-65 publizierten.

Herwig Feichtinger

EMUF steuert Relaisfunkstelle

Relaisfunkstellen dienen Funkamateuren zum Vergrößern der Reichweite tragbarer und mobiler UKW- und UHF-Funkgeräte. Sie werden an exponierten Standorten aufgestellt, mit einem 1750-Hz-Ton eingeschaltet und müssen nach den postalischen Vorschriften ihr zugeteiltes Rufzeichen in Morsecode ausstrahlen. Der EMUF stellt die bisher wohl einfachste, preiswerteste und flexibelste Steuerung einer solchen Station dar.

Eine Relaisfunkstelle besteht grundsätzlich aus drei Funktionseinheiten: einem Empfänger (z. B. 145,0 MHz), einem mit dem Empfangssignal modulierten Sen-

der (z. B. 145,6 MHz) und einer Steuerung, die das postalisch vorgeschriebene Morse-Rufzeichen erzeugt, den 1750-Hz-Ton als Einschaltsignal erkennen muß

und den Sender ein- und ausschaltet. Man kann sich leicht vorstellen, daß die Realisation einer solchen Steuerung in herkömmlicher Digital- und Analogtechnik einen nicht unerheblichen Aufwand darstellt. Erbauer von Relaisfunkstellen wissen es daher zu schätzen, wenn sich dieser Aufwand auf eine einzige Europakarte mit wenigen Bauelementen reduziert.

Bild 1 gibt das typische Blockschaltbild einer Relaisfunkstelle wieder und zeigt auch die Beschaltung des EMUF. Er erhält das Nf-Ausgangssignal des Empfängers mit TTL-Pegel (z. B. über drei hintereinandergeschaltete CMOS-Inverter als Verstärker), damit er mit Hilfe der Autokorrelation [1] den 1750-Hz-Ton auswerten kann. Ferner benötigt er ein Signal von der Rauschsperrung des Empfängers, das ihm mitteilt, ob ein Signal empfangen wird (wenn ja: Low-Pegel an PB6).

Ausgangsseitig steuert der EMUF den Sender (ein: Low an PB1) und liefert das Morsesignal mit TTL-Pegel als Niederfrequenz an PB0.

Bild 2 zeigt das Assemblerlisting des EMUF-Programms, hier mit DC1YB als Beispielrufzeichen (auch längere Rufzeichen können unter CALL programmiert werden, da mit drei FF-Bytes hierfür Speicherplatz freigehalten wurde). Solange der Sender eingeschaltet ist, dient ein empfangener 1750-Hz-Tonruf zum Abrufen eines unter TEXT programmierbaren Morsetextes, der praktisch beliebig lang sein kann und dessen Codierung aus der Tabelle hervorgeht. Er muß mit einem Null-Byte abgeschlossen werden (vgl. [2]). Hört die gerade empfangene Station zu senden auf, so wird ein kurzer „Roger-Beep“ ausgestrahlt, hier als Morsezeichen K. Wenn nur ein normaler Roger-Ton gewünscht wird, muß der Morsecode für T unter dem Label K programmiert werden.

Das Morse-Rufzeichen wird sofort beim Einschalten des Relais sowie in regelmäßigen Abständen von etwa drei Minuten ausgesendet. Wird länger als etwa vier Sekunden kein Signal empfangen, so schaltet sich der Sender wieder aus. Eine Reihe von Parametern kann bei Bedarf im Programm geändert werden. Deren Adressen sind in Bild 2: 0C13 = Mindest-Tonrufdauer zum Einschalten; 0C2A = Rufzeichen-Wiederholzeit; 0C2E = Roger-Beep-Verzögerungszeit (z. B. FC = doppelte Zeit); 0C47 = Haltezeit bis zum Ausschalten des Senders; 0C56 = Mindest-Tonrufdauer zum Abrufen des vorprogrammierten Textes; 0C73 = Tonruf-Auswertefrequenz (hier 1750 Hz); 0C99 = Tonruf-Auswertemp-

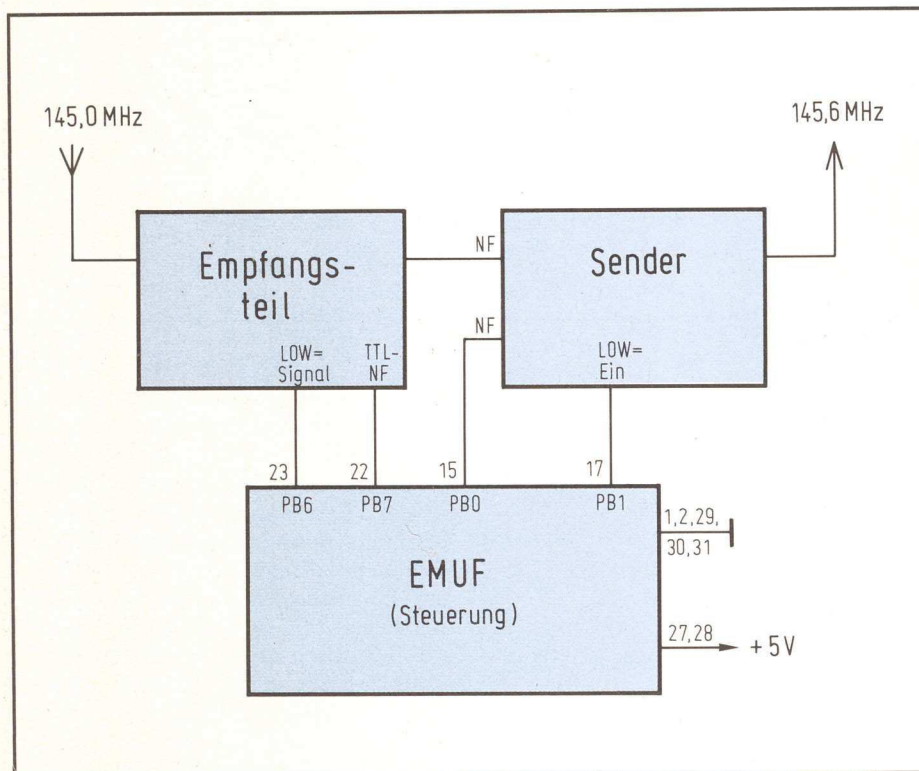


Bild 1. Der EMUF erhält ein Rauschsperr- und ein Nf-Signal vom Empfangsteil, schaltet den Sender ein und aus und erzeugt die Nf der Morse-Ausstrahlung

findlichkeit (z. B. 03 = empfindlicher); OCA7 = Zeit vom Einschalten des Senders bis zum Morse-Rufzeichen; OCD9 = Morse-Geschwindigkeit; OCDB = Morse-Tonfrequenz.

Da das Programm nur etwa ein Viertel KByte belegt, könnte man noch eine ganze Reihe zusätzlicher Funktionen einbauen; darauf wurde hier verzichtet, weil die Relais-Verantwortlichen meist sehr unterschiedliche Vorstellungen über solche Spezial-Features entwickeln. Der Verfasser ist gern bereit, die

abgedruckte Programmversion mit den jeweils gewünschten Rufzeichen und Texten für postalisch lizenzierte Relaisfunkstellen-Erbauer in ein EPROM 2716 zu brennen (Adresse wie mc-Redaktion).

Codierung der Morsezeichen

| | | | | | | | |
|---|----|---|----|---|----|---|----|
| 0 | FC | ? | 32 | J | 78 | T | C0 |
| 1 | 7C | A | 60 | K | B0 | U | 30 |
| 2 | 3C | B | 88 | L | 48 | V | 18 |
| 3 | 1C | C | A8 | M | E0 | W | 70 |
| 4 | 0C | D | 90 | N | A0 | X | 98 |

| | | | | | | | |
|---|----|---|----|---|----|---|----|
| 5 | 04 | E | 48 | O | F0 | Y | B8 |
| 6 | 84 | F | 20 | P | 68 | Z | C8 |
| 7 | C4 | G | D0 | Q | D8 | / | 94 |
| 8 | E4 | H | 08 | R | 50 | | |
| 9 | F4 | I | 20 | S | 10 | | |

Der Leerraum zwischen Worten wird mit hex FF codiert.

Literatur

- [1] Tonerkennung per Software (Autokorrelation). mc 1981, Heft 4.
- [2] EMUF als Morse-Rufzeichengeber. In diesem EMUF-Sonderheft.

```

0000 ;REPEATER (DC1YB)
0000 ;PB7=AF INPUT (TTL)
0000 ;PB6=SQUELCH INPUT
0000 ;PB1=TX OFF OUTPUT
0000 ;PBO=AF OUTPUT
0000 TIL *=*+1
0001 TIM *=*+1
0002 TIH *=*+1
0003 THLD *=*+1
0004 SUM *=*+1
0005 SPL1 *=*+1
0006 SPL2 *=*+1
0007 PB =$802
0007 PBD =$803
0007 *=$FFC
OFFC 000C .WOR RES
OFFE *= $COO
OC00 RES A2FF LDX £$FF
OC02 9A TXS
OC03 D8 CLD
OC04 78 SEI
OC05 A903 LDA £3
OC07 8D0308 STA PBD
OC0A ;TX OFF
OC0A OFF ADO208 LDA PB
OC0D 0902 ORA £2
OC0F 8D0208 STA PB
OC12 ;WAIT FOR 1750 HZ
OC12 A264 LDX £100
OC14 OFF1 20670C JSR AK
OC17 10F1 BPL OFF
OC19 CA DEX
OC1A DOF8 BNE OFF1
OC1C ;TX ON
OC1C ADO208 LDA PB
OC1F 29FD AND £$FD
OC21 8D0208 STA PB
OC24
OC24 ;SEND CW CALL
OC24 CA A200 LDX £0
OC26 20A40C JSR CW
OC29 A902 LDA £2
OC2B 8502 STA TIH
OC2D ON A9FE LDA £$FE
OC2F 8503 STA THLD
OC31 ;COUNT DOWN TIMER
OC31 ON3 C600 DEC TIL
OC33 D020 BNE ON2
OC35 C601 DEC TIM
OC37 D004 BNE ON1
OC39 C602 DEC TIH
OC3B FOE7 BEQ CA
OC3D ON1 2C0208 BIT PB
OC40 50EB BVC ON
OC42 E603 INC THLD
OC44 F008 BEQ BEEP
OC46 A90A LDA £10
OC48 C503 CMP THLD
OC4A 30BE BMI OFF
OC4C 1007 BPL ON2
OC4E ;SEND ROGER BEEP
OC4E
OC4E BEEP
A209 LDX £K-CALL
8606 STX SPL2
20AA0C JSR NXT
;TEXT REQUEST?
OC55 ON2 A264 LDX £100
OC57 REQ 20670C JSR AK
OC5A 10D5 BPL ON3
OC5C CA DEX
OC5D DOF8 BNE REQ
OC5F A20B LDX £TXT-CALL
OC61 20A40C JSR CW
OC64 4C310C JMP ON3
OC67
OC67 ;AUTOCORRELATION
OC67 ;1750HZ,A/X/Y SAVED
OC67 AK 48 PHA
OC68 98 TYA
OC69 48 PHA
OC6A 8A TXA
OC6B 48 PHA
OC6C A900 LDA £0
OC6E 8504 STA SUM
OC70 A010 LDY £16
OC72 AK1 A20A LDX £10
OC74 AK2 CA DEX
OC75 DOFD BNE AK2
OC77 ADO208 LDA PB
OC7A OA ASL A
OC7B 6605 ROR SPL1
OC7D 6606 ROR SPL2
OC7F 88 DEY
OC80 DOFO BNE AK1
OC82 A505 LDA SPL1
OC84 F010 BEQ AK5
OC86 C9FF CMP £$FF
OC88 FOOC BEQ AK5
OC8A 4506 EOR SPL2
OC8C A007 LDY £7
OC8E AK3 6A ROR A
OC8F B002 BCS AK4
OC91 C604 DEC SUM
OC93 AK4 88 DEY
OC94 10F8 BPL AK3
OC96 AK5 A504 LDA SUM
OC98 6905 ADC £5
OC9A 8504 STA SUM
OC9C 68 PLA
OC9D AA TAX
OC9E 68 PLA
OC9F A8 TAY
OCA0 68 PLA
OCA1 2404 BIT SUM
OCA3 RDY 60 RTS
OCA4 ;SEND CW CODE
OCA4 CW 8606 STX SPL2
OCA6 A21E LDX £30
OCA8 D002 BNE *+4
OCAA NXT A202 LDX £2
OCAC SPC 18 CLC
OCAD 20D80C JSR ELEM
OCBO A606 LDX SPL2
OCB2 E606 INC SPL2
OCB4 BDF30C LDA CALL,X
OCB7 FOEA BEQ RDY
OCB9 8505 STA SPL1
OCBB C9FF CMP £$FF
OCBD D004 BNE SHFT
OCBF A207 LDX £7
OCC1 D0E9 BNE SPC
OCC3 SHFT 0605 ASL SPL1
OCC5 FOE3 BEQ NXT
OCC7 A201 LDX £1
OCC9 9002 BCC DOT
OCCB A203 LDX £3
OCCD DOT 38 SEC
OCCE 20D80C JSR ELEM
OCD1 18 CLC
OCD2 E8 INX
OCD3 20D80C JSR ELEM
OCD6 FOEB BEQ SHFT
OCD8 ELEM A096 LDY £150
OCDA ELEM1 A932 LDA £50
OCDC 8504 STA SUM
OCDE ELEM2 C604 DEC SUM
OCE0 DOFC BNE ELEM2
OCE2 9008 BCC QUIET
OCE4 ADO208 LDA PB
OCE7 4901 EOR £1
OCE9 8D0208 STA PB
OCEC QUIET 88 DEY
OCED DOEB BNE ELEM1
OCF0 CA DEX
OCF1 DOE6 BNE ELEM
OCF2 60 RTS
OCF3 ;CW CALL 'DC1YB'
OCF3 CALL 90 .BYT $90,$A8
OCF4 A8
OCF5 7C .BYT $7C,$B8
OCF6 B8
OCF7 88 .BYT $88,0
OCF8 00
OCF9 FF .BYT $FF,$FF,$FF
OCFA FF
OCFB FF
OCFC ;ROGER BEEP 'K'
OCFC K BO .BYT $B0,0
OCFD 00
OCFE ;CW TEXT
OCFE ;'TEST DE DC1YB'
OCFE TXT CO .BYT $C0,$40
OCFF 40
OD00 10 .BYT $10,$C0
OD01 CO
OD02 FF .BYT $FF,$90
OD03 90
OD04 40 .BYT $40,$FF
OD05 FF
OD06 90 .BYT $90,$A8
OD07 A8
OD08 7C .BYT $7C,$B8
OD09 B8
OD0A 88 .BYT $88,0
OD0B 00
OD0C .END
OD0C ERRORS= 0000

```

Bild 2. Assemblerlisting des auf einem PC-100 entwickelten und mit einer Typenrad-Schreibmaschine per EMUF-Interface ausgedruckten Relaisfunkstellen-Steuerprogramms

Herwig Feichtinger

Der Whisky-EMUF

Die Besucher der Stuttgarter Ausstellung Hobby-Elektronik '81 konnten in unserem Stand eine besonders nützliche Anwendung des EMUF bestaunen, nämlich den Ausschank einer „hochohmigen“ Flüssigkeit an Besucher, die schnell genug im Kopf eine angezeigte Zufalls-Dualzahl (4 Bit) in eine Hexadezimal-Ziffer umrechnen konnten.

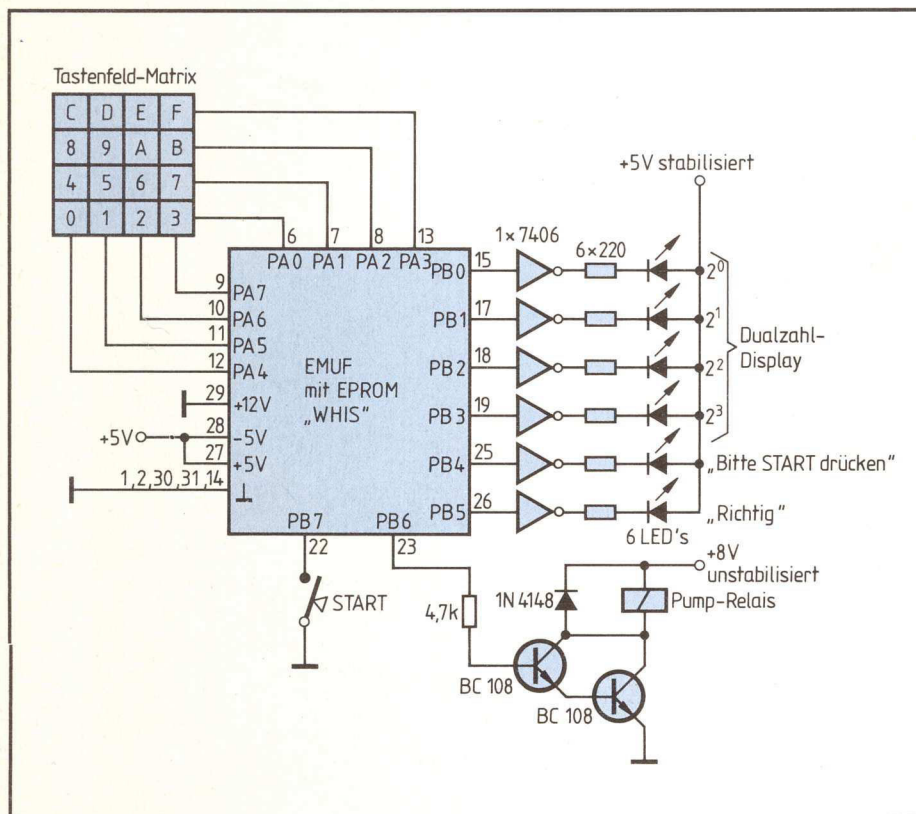


Bild 1. Schaltbild des Whisky-EMUF. Als Pumpe, die sinnvollerweise (wegen des hohen Strombedarfs) über ein kleines Relais angesteuert wird, eignen sich Ausführungen aus der Kfz-Scheibenwaschanlage oder aus dem Aquarium



Der Ablauf bei der Bedienung des Whisky-EMUF ist folgender: Der Benutzer drückt zunächst kurz eine Starttaste. Sofort erscheint auf vier LEDs eine binäre Zufallszahl. Das Hex-Äquivalent dieser Zahl (0...F) muß er nun möglichst schnell auf einer kleinen Tastatur mit 16 Tasten eintippen. War es die richtige Ziffer, so leuchtet eine LED zur Bestätigung auf. Außerdem wird ein kleines Glas Whisky mit Hilfe einer Pumpe ausgeschenkt, wenn die Beantwortung innerhalb einer bestimmten Zeit geschah. Falls er knapp unter der Zeitgrenze lag, bekommt der Besucher als „Trostpries“ die halbe Menge Whisky. Eine kleine Bosheit ist noch eingebaut. Abhängig vom mittleren Intelligenzquotienten der Besucher paßt sich die vorgegebene maximale Beantwortungszeit automatisch an: Kommen viele schlaue Leute, wird die Zeit langsam kürzer, und kommen nur Leute, die keine Ahnung von Dual- und Hexzahlen haben, wird sie wieder verlängert. Außerdem lassen sich natürlich manche Dualzahlen, wie 0001 oder 1111, besonders leicht erraten und erhalten deshalb einen „Malus“. Da für so etwas ein paar Monoflops und Gatter schon nicht mehr ausreichen würden, kam der Einplatinen-Computer EMUF zum Einsatz (Bild 1). Mit wenigen zusätzlichen Bauelementen und einem kleinen Netzteil übernimmt er die Tastenfeldabfrage sowie die Ansteuerung der LEDs und der Pumpe, wofür sich übrigens Ausführungen aus der Kfz-Scheibenwaschanlage ebenso eignen wie aus einem kleinen Aquarium. Bild 2 zeigt die EMUF-Software.

Bild 2. Das Programm wurde mit einer Olivetti-Typenrad-Schreibmaschine (ebenfalls mit EMUF als Interface) ausgedruckt. Typenradbedingung erscheint statt dem Doppelkreuz-Zeichen ein englisches Pfundzeichen

```

0001      PASS 1
1000      PASS 2
0000
0000      ;MC-WHISKY-EMUF
0000      ;V1.1/FE/26.7.1981
0000      *=0
0000 TIME  *=*+1
0001 KEY   *=*+1
0002 RND   *=*+1
0003 TI    *=$COO
0C00 PA    =$800
0C00 PAD   =$801
0C00 PB    =$802
0C00 PBD   =$803
0C00 CLK   =$817
0C00 RES   A90F LDA £$F ;INIT
0C02      8D0108 STA PAD ;PORTS
0C05      A97F LDA £$7F
0C07      8D0308 STA PBD
0C0A      A2FF LDX £$FF ;AND STACK
0C0C      9A TXS
0C0D      A946 LDA £70 ;DEFAULT
0C0F      8503 STA TI ;TIMING
0C11      D8 CLD
0C12      78 SEI
0C13 WAIT  E8 INX ;X-RANDOM
0C14      2C1708 BIT CLK
0C17      100F BPL WAIT1
0C19      A503 LDA TI
0C1B      8D1708 STA CLK
0C1E      AD0208 LDA PB
0C21      2910 AND £$10
0C23      4910 EOR £$10 ;FLASH
0C25      8D0208 STA PB ;START LED
0C28 WAIT1 2C0208 BIT PB ;START?
0C2B      30E6 BMI WAIT
0C2D      8A TXA ;YES
0C2E      290F AND £$F
0C30      8502 STA RND ;DISPLAY NR
0C32      8D0208 STA PB
0C35      A900 LDA £0 ;START TIMER
0C37      8500 STA TIME
0C39 LP    20BC0C JSR DLY
0C3C      E600 INC TIME
0C3E      30D3 BMI WAIT
0C40      20980C JSR SCAN
0C43      DOF4 BNE LP
0C45      A501 LDA KEY ;CORRECT
0C47      C502 CMP RND ;KEY?
0C49      D03B BNE ERR
0C4B      0920 ORA £$20 ;YES,
0C4D      8D0208 STA PB ;LED ON
0C50      A500 LDA TIME
0C52      A602 LDX RND
0C54      18 CLC
0C55      7DC70C ADC MAL,X
0C58      C91E CMP £30
0C5A      B02A BCS ERR ;NO WHISKY
0C5C OK    C91C CMP £28
0C5E      B00A BCS HALF
0C60      A503 LDA TI ;TOO EASY?!
0C62      E905 SBC £5 ;ADAPT TI
0C64      8503 STA TI
0C66      A214 LDX £20 ;FULL GLASS,
0C68      D002 BNE PMP ;CHEERS!
0C6A HALF  A20A LDA £10 ;HALF
0C6C PMP   AD0208 LDA PB
0C6F      0940 ORA £$40 ;PUMP IT
0C71      8D0208 STA PB ;FROM BOTTLE
0C74 PMP1  A962 LDA £98
0C76      20BEOC JSR DLYO
0C79      CA DEX
0C7A      DOF8 BNE PMP1
0C7C      AD0208 LDA PB
0C7F      292F AND £$2F ;PUMP OFF
0C81      8D0208 STA PB
0C84      D002 BNE PAU
0C86 ERR   E603 INC TI ;NOT EASY
0C88 PAU   A23C LDX £60 ;WAIT
0C8A PAU1  20BC0C JSR DLY ;A FEW
0C8D      CA DEX ;SECONDS
0C8E      DOFA BNE PAU1 ;LOOP IF
0C90      2C0208 BIT PB ;START KEY
0C93      10F3 BPL PAU ;STILL DOWN
0C95      4C130C JMP WAIT
0C98      ;SCAN KEYBOARD
0C98      ;HEX VALUE IN A,
0C98      ;Z=1 IF KEY PRESSED
0C98 SCAN  A900 LDA £0
0C9A      8501 STA KEY
0C9C      A203 LDX £3
0C9E      A90E LDA £$E
0CA0      8D0008 STA PA
0CA3 SCAN1 A004 LDY £4
0CA5      A910 LDA £$10
0CA7 SCAN2 2C0008 BIT PA
0CAA      FOOF BEQ FND
0CAC      E601 INC KEY
0CAE      0A ASL A
0CAF      88 DEY
0CBO      DOF5 BNE SCAN2
0CB2      0E0008 ASL PA
0CB5      EE0008 INC PA
0CB8      CA DEX
0CB9      10E8 BPL SCAN1
0CBB FND   60 RTS
0CBC      ;DELAY TI
0CBC DLY   A503 LDA TI
0CBE DLYO  8D1708 STA CLK
0CC1 DLY1  2C1708 BIT CLK
0CC4      10FB BPL DLY1
0CC6      60 RTS
0CC7      ;MALUS TAB. FOR 0-F
0CC7 MAL   05 .BYT 5,10,8,5
0CC8      0A
0CC9      08
0CCA      05
0CCB      08 .BYT 8,0,0,6
0CCC      00
0CCD      00
0CCE      06
0CCF      08 .BYT 8,2,0,0
0CD0      02
0CD1      00
0CD2      00
0CD3      02 .BYT 2,0,2,5
0CD4      00
0CD5      02
0CD6      05
0CD7      ;THAT'S ALL!
0CD7      FF .BYT $FF,$FF,$FF
0CD8      FF
0CD9      FF
0CDA      4D43 .BYT 'MC-EMUF/FE'
0CE4      FF .BYT $FF,$FF,$FF
0CE5      FF
0CE6      FF
0CE7      *=$FFC
0CFC      000C .WOR RES ;VECTOR
0CFF      000C .WOR RES
1000      .END
1000      ERRORS= 0000

```


Dipl. Ing. Rolf-Dieter Klein

Der LCD-EMUF

Oftmals kommt ein Datensichtgerät als Anzeige wegen des Preises nicht in Frage, andererseits genügt eine einfache Siebensegmentanzeige ebenfalls nicht, da alphanumerische Daten ausgegeben werden müssen. Für diesen Fall eignet sich der LCD-EMUF, mit dem eine 16stellige alphanumerische LCD-Anzeige betrieben werden kann. Der EMUF läßt sich dann wie eine Centronics-Parallelschnittstelle anschließen.

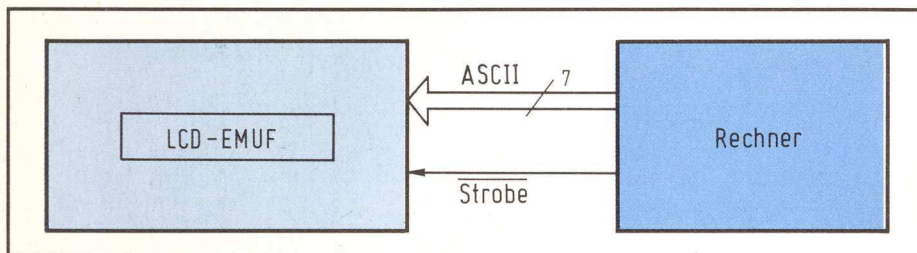


Bild 1. Anschluß des EMUF an einen Host-Rechner

Bild 1 zeigt die Verbindung des EMUFs mit einem Hostrechner. Als LCD-Anzeige wird eine Anzeigeeinheit LCM1010 von Siemens verwendet, die schon die Ansteuerlogik enthält, nicht aber die

Steuerung. Die Anzeige wird mit einem Takt versorgt, aus dem sich das ganze Timing ableitet. Die Periodendauer beträgt 4 ms. Bild 2 zeigt eine einfache Schaltung zur Gewinnung des Taktes.

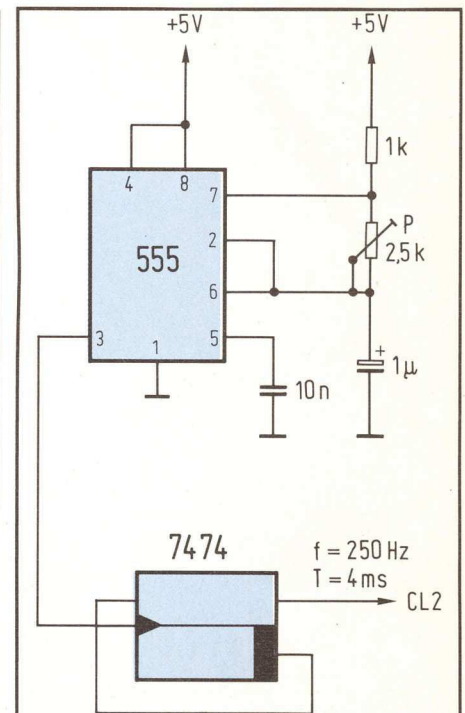


Bild 2. Takterzeugung für die Anzeige

Aus Sicherheits- und Zeitgründen sollte nicht der EMUF selbst diesen Takt erzeugen. An dem 2,5-kΩ-Regler kann die Frequenz abgeglichen werden. Das nachgeschaltete Flipflop sorgt für einen symmetrischen Takt, was unbedingt nötig ist. Bild 3 zeigt schließlich die Beschaltung des EMUF. Der Port PA dient der Dateneingabe. Port PB wird zur Ansteuerung der LCD-Anzeige verwendet. Die Flüssigkristallanzeige benötigt 5 V als Betriebsspannung und zusätzlich eine einstellbare Spannung, die mit dem 2,5-kΩ-Widerstand justiert werden kann. Damit kann der optimale Kontrast eingestellt werden.

Bild 4 zeigt das Taktschema zur Ansteuerung. Der Takt CL2 ist der Grundtakt. Mit ST1 werden die in einem Schieberegister der Anzeige befindlichen Werte in das Anzeigeregister übernommen. Die Daten werden vom EMUF mit Hilfe von CL1 und DI eingegeben. Dabei werden 5 mal 16 Datenbits einer Zeile plus 3 leere Takte plus 7 Takte mit der Information, welche Zeile als nächste angezeigt werden soll, übermittelt. Das Datenbit muß jeweils bei der fallenden Flanke von CL1 gültig sein. In der Zeit von 4 ms muß der EMUF die Zeichen aus dem internen Buffer entnehmen in die jeweilige Zeichenadresse des Zeichengenerators umrechnen und Bitweise ausgeben. Dies ist natürlich mit Zeitproblemen verbunden. Dennoch schafft es der EMUF gerade noch in den

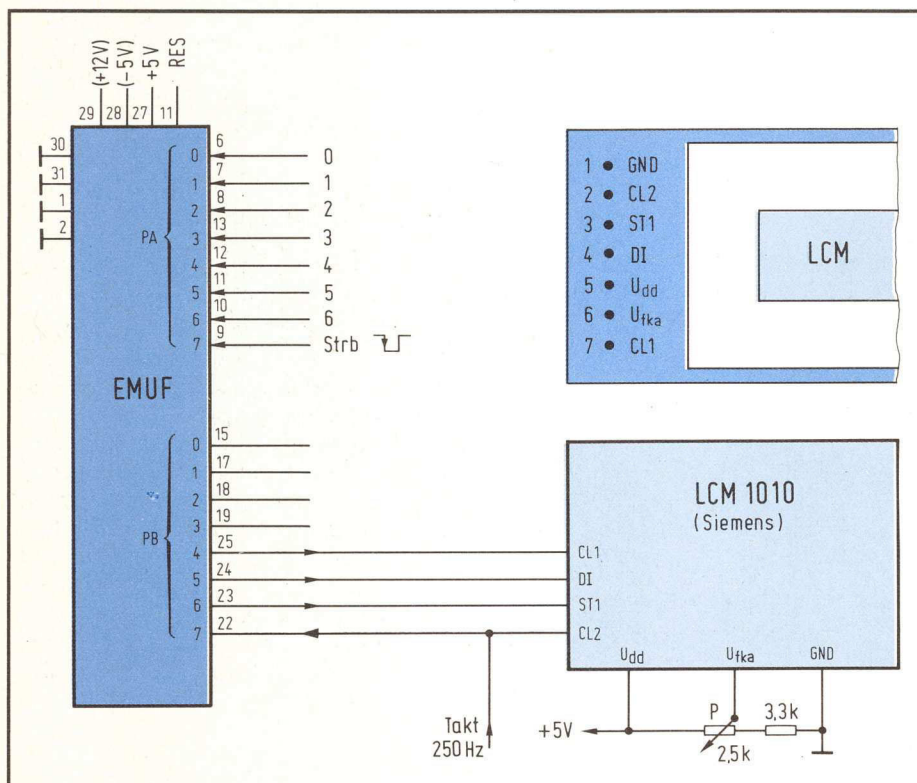


Bild 3. Anschlußbelegung des EMUF und der Flüssigkristall-Anzeige

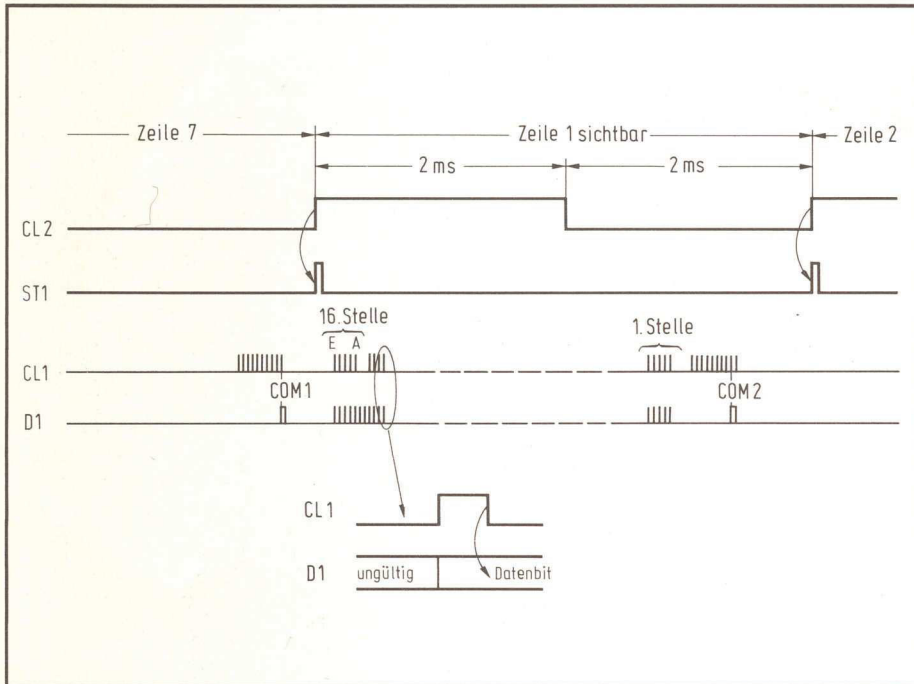


Bild 4. Impulsdiagramme zur LCD-Anzeige

```

OD80 00 00 00 00 00 00 00 00 00 04 04 04 04 00 04 00
OD90 0A 0A 0A 00 00 00 00 00 0A 0A 1F 0A 1F 0A 0A 00
ODAO 04 0F 14 0E 05 1E 04 00 19 19 02 04 08 13 13 00
ODBO 08 14 14 08 15 12 0D 00 04 04 04 00 00 00 00 00
ODCO 02 04 08 08 08 04 02 00 08 04 02 02 02 04 08 00
ODDO 00 15 0E 1F 0E 15 00 00 00 04 04 1F 04 04 00 00
ODE0 00 00 00 00 00 0C 04 08 00 00 00 00 1F 00 00 00
ODFO 00 00 00 00 00 0C 0C 00 01 01 02 04 08 10 10 00
OE00 0C 12 12 12 12 12 0C 00 04 0C 04 04 04 04 0E 00
OE10 0E 11 01 0E 10 10 1F 00 0E 11 01 06 01 11 0E 00
OE20 02 06 0A 12 1F 02 02 00 1F 10 1E 01 01 11 0E 00
OE30 06 08 10 1E 11 11 0E 00 1F 01 02 04 08 10 10 00
OE40 0E 11 11 0E 11 11 0E 00 0E 11 11 0F 01 02 0C 00
OE50 00 0C 0C 00 0C 0C 00 00 00 0C 0C 00 0C 04 08 00
OE60 01 02 04 08 04 02 01 00 00 00 1F 00 1F 00 00 00
OE70 10 08 04 02 04 08 10 00 0C 12 02 04 04 00 04 00
OE80 0E 11 01 0D 15 15 0E 00 04 0A 11 11 1F 11 11 00
OE90 1E 09 09 0E 09 09 1E 00 0E 11 10 10 10 11 0E 00
OEAO 1E 09 09 09 09 09 1E 00 1F 10 10 1C 10 10 1F 00
OEB0 1F 10 10 1C 10 10 10 00 0F 10 10 13 11 11 0F 00
OECO 11 11 11 1F 11 11 11 00 0E 04 04 04 04 04 0E 00
OEDO 01 01 01 01 01 11 0E 00 11 12 14 18 14 12 11 00
OEE0 10 10 10 10 10 10 1F 00 11 1B 15 15 11 11 11 00
OEF0 11 19 15 13 11 11 11 00 1F 11 11 11 11 11 1F 00
OF00 1E 11 11 1E 10 10 10 00 0E 11 11 11 15 12 0D 00
OF10 1E 11 11 1E 14 12 11 00 0E 11 08 04 02 11 0E 00
OF20 1F 04 04 04 04 04 04 00 11 11 11 11 11 11 0E 00
OF30 11 11 11 0A 0A 04 04 00 11 11 11 11 15 1B 11 00
OF40 11 11 0A 04 0A 11 11 00 11 11 0A 04 04 04 00 00
OF50 1F 01 02 04 08 10 1F 00 0E 08 08 08 08 08 0E 00
OF60 10 10 08 04 02 01 01 00 0E 02 02 02 02 02 0E 00
OF70 04 0E 15 04 04 04 04 00 00 00 00 00 00 00 1F 00
    
```

Bild 6. Listing des Zeichengenerators

4 ms. Der Abgleich des Taktes erfolgt daher mit Hilfe eines Oszilloskopes, so daß das Timing mit dem Bild übereinstimmt. Ist CL2 zu hoch, so schafft es der EMUF nicht mehr bei jeder steigenden Flanke von CL2, ST1 zu erzeugen und es bleibt eine Taktzeit aus. Dadurch sinkt die Multiplexrate und die Anzeige beginnt zu flimmern. Bild 5 zeigt den Aufbau der Zeichenmatrix. Es werden je Zeichen 5 mal 7 Punkte angezeigt. Der Zeichengenerator ist aber 8 mal 8 organisiert, um den Rechenaufwand klein zu halten. Bild 6 zeigt den Dump des Zeichengenerators. Er liegt im Adressbereich D80 bis F7F. Es können also 64 verschiedene ASCII-Zeichen dargestellt werden.

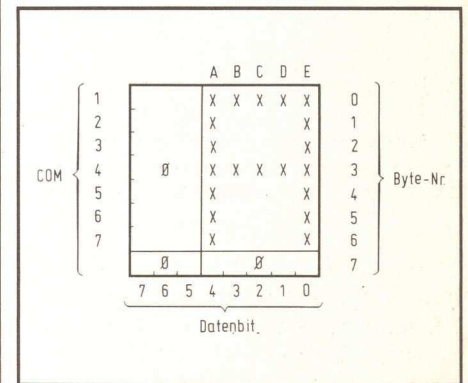


Bild 5. Aufbau eines Zeichens

Bild 7 zeigt das Listing. Die Eingabe der Daten erfolgt im ASCII-Code in einen 16-Zeichen-Buffer. Dies geschieht durch Interrupt. Das Interruptprogramm prüft als erstes drei Sonderzeichen ab:
 CR (0D): Schreibposition auf Anfang
 LF (0A): Löschen und auf Anfang
 BS (08): ein Zeichen nach links
 Alle anderen ASCII-Zeichen werden dann übersetzt, wobei Kleinbuchstaben in Großbuchstaben umgewandelt werden. Steuerzeichen werden ansonsten ignoriert. Das Hauptprogramm des EMUFs hat die Aufgabe, den Inhalt des Buffers zyklisch auszugeben. Bei der Inbetriebnahme des EMUF erscheint als erstes auf der Anzeige der Testtext, mit dem sich der Autor verewigte, nämlich „RDK“. Damit ist eine Funktionsprüfung ohne Rechneranschluß möglich. Ein Cursor läßt sich vom Rechner aus durch vorlaufendes Einschreiben der Sequenz Zeichen-BS-Unterstreichen-BS erreichen.

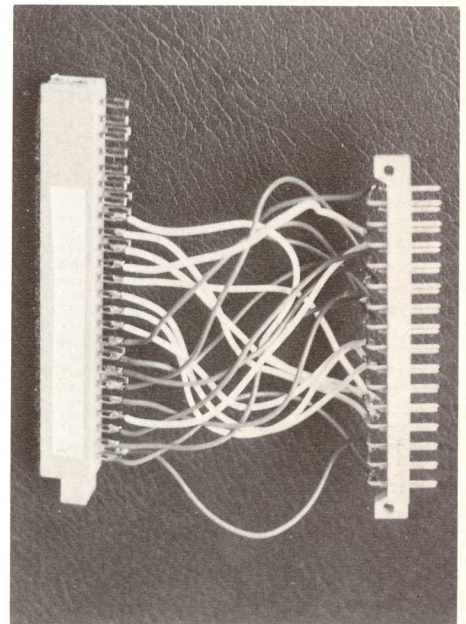
Literatur

LCM-1010-Kurzbeschreibung. Siemens AG, München.


```

00A7 A900      LDA      #LOW ZGE      ;ZEICHENGEN ADRESSE LSB
00A9 18        CLC
00AA 6511      ADC      ZADR
00AC 8511      STA      ZADR
00AE A900      LDA      #HIGH ZGE
00B0 6512      ADC      ZADR      ;8*ZEICHEN+ZGE->ZADR
00B2 8512      STA      ZADR
00B4 A905      LDA      #5
00B6 8514      STA      CNT      ;5 BITS
00B8 B111      LDA      (ZADR),Y
00BA 8513      STA      ZEICH1
00BC 6613      LP3:    ROR      ZEICH1
00BE B000      BCS      EINS      ;=1 DANN DUNKLER PUNKT
00C0 A910      LDA      #%00010000
00C2 800208    STA      PB
00C5 A900      LDA      #%00000000
00C7 800208    STA      PB
00CA 4CD80C    EINS:    JMP      CONT
00CC A930      LDA      #%00110000
00CE 800208    STA      PB
00D0 A920      LDA      #%00100000
00D2 800208    STA      PB
00D4 800208    JMP      CONT      ;ZEIT GLEICH
00D7 4CD80C    CONT:    DEC      CNT      ;UNTIL ALLE BITS
00DA C614      BNE      LP3
00DC D0DE      DEX
00DE CA        BNE      LP2      ;UNTIL ALLE ZEICHEN
00DF D0B5      LDA      #%00010000
00E1 A910      STA      PB      ;DREI TAKTE LEER
00E3 800208    LDA      #%00000000
00E6 A900      STA      PB
00E8 800208    LDA      #%00010000
00EB A910      STA      PB
00ED 800208    LDA      #%00000000
00F0 A900      STA      PB
00F2 800208    LDA      #%00010000
00F5 A910      STA      PB
00F7 800208    LDA      #%00000000
00FA A900      STA      PB
00FC 800208    LDA      #7
00FF A907      STA      CNT      ;RESTL 7 ADRESSBITS
0001 8514      LDA      COM
0003 A515      STA      ZEICH1
0005 8513      STA      ZEICH1      ;MASKE
0007 0613      LP4:    BCS      EINS1     ;=1 DANN DUNKLER PUNKT
0009 B000      LDA      #%00010000
000B A910      STA      PB
000D 800208    LDA      #%00000000
0010 A900      STA      PB
0012 800208    LDA      #7
0015 4C250D    JMP      CONT1
0018          EINS1:    LDA      #%00110000
001A 800208    STA      PB
001D A920      LDA      #%00100000
001F 800208    STA      PB
0022 4C250D    CONT1:  JMP      CONT1     ;ZEIT GLEICH
0025 C614      BNE      LP4
0027 D0DE      LDA      #0
0029 A900      STA      PB
002B 800208    AND      #10000000
002E AD0208    LW1:    BNE      LW1      ;VOR DEM CLOCK
0031 2980      AND      #10000000
0033 D0F9      BNE      LW1      ;WARTEN POS FLANKE CL2
0035 AD0208    LW2:    LDA      PB
0038 2980      AND      #10000000
003A F0F9      BEQ      LW2
003C A940      LDA      #10000000
003E 800208    STA      PB
0041 A900      LDA      #10000000
0043 800208    STA      PB
0046 0615      ASL      COM
0048 C8        INY
0049 C007      CPY      #7
004B D003      BNE      LP11
004D 4C8E0C    JMP      START
0050 4C940C    LP11:  JMP      LP1
0000          ;
          END
    
```

lungssystems ebenfalls an eine 31polige Steckleiste führen – das Entwicklungssystem tut so, als wäre es selbst der EMUF.



Verwendet man den AIM-65 oder PC-100 zur EMUF-Programmentwicklung, so kann man dazu ein einfaches Adapterkabel verwenden (Bild). Die Anschlüsse der AIM-Applikationsmessleiste werden mit der 31poligen Steckleiste wie folgt verbunden:

| Appl.-Conn. Pin | EMUF-Stecker Pin | Leitung |
|-----------------|------------------|---------|
| 1 | 1, 2 | Masse |
| A | 27 | +5 V |
| 2 | 13 | PA3 |
| 3 | 8 | PA2 |
| 4 | 7 | PA1 |
| 5 | 12 | PA4 |
| 6 | 11 | PA5 |
| 7 | 10 | PA6 |
| 8 | 9 | PA7 |
| 9 | 15 | PB0 |
| 10 | 17 | PB1 |
| 11 | 18 | PB2 |
| 12 | 19 | PB3 |
| 13 | 25 | PB4 |
| 14 | 6 | PA0 |
| 15 | 22 | PB7 |
| 16 | 24 | PB5 |
| 17 | 23 | PB6 |

Da die Ports des AIM-User-VIA 6522 benutzt werden, kann natürlich die (ohnehin selten verwendete) Interrupt-Möglichkeit an PA7 während der Entwicklungsphase nicht simuliert werden. Trotzdem spart der kleine Adapter eine Menge Zeit bei der Realisation neuer EMUF-Applikationen.

Fe.

EMUF-AIM-Adapterkabel

Der Einplatinencomputer EMUF verfügt über eine 31polige Steckleiste, über die er mit der jeweiligen Applikationsschaltung verbunden wird (es sei denn, diese sitzt ganz oder teilweise auf der EMUF-Lochrasterfläche). Während der Pro-

grammentwicklung ist es praktisch, einen Adapter zu haben, der das Entwicklungssystem als EMUF-Ersatz arbeiten läßt. Zu diesem Zweck muß man die I/O-Ports sowie die Masse- und Betriebsspannungsanschlüsse des Entwick-

Rudolf Hofer

Interface für Typenrad-Schreibmaschine

Seit einiger Zeit bietet das Kaufhaus Quelle die Typenrad-Schreibmaschine Olivetti P-30 unter dem etwas unhandlichen Namen „privileg electronic automatic correction schreibmaschine“ für knapp 1100 DM an. Anders als bei Kugelkopfmaschinen werden bei diesem Modell alle Funktionen elektronisch gesteuert. Das heißt, mit den Tasten werden lediglich Kontakte betätigt. Um einen Drucker zu erhalten, muß man lediglich die Tasten elektronisch nachbilden. Will man jedoch einen Drucker mit Standard-Schnittstelle, dann muß man schon etwas „Intelligenz“ hinzufügen – eine ideale Aufgabe für den EMUF.

Um Leitungen zu sparen, empfängt der EMUF die Zeichen seriell im ASCII-Code (7 Bit, ohne Paritätsbit). Fügt man Pegelwandler hinzu, dann erhält man eine V.24-Schnittstelle mit einer Rückmeldeleitung, die angibt, ob der Drucker gerade beschäftigt ist. Damit kann die Übertragungsgeschwindigkeit beliebig groß sein. Die Schaltung der gesamten Schnittstelle ist in Bild 1 dargestellt. Um die Schreibmaschine anzuschließen, muß man den Gehäusedeckel abnehmen. Tastatur und Prozessorkarte sind mit zwei Bandkabeln verbunden. Blickt man von vorne auf die Maschine, dann werden die Anschlüsse dieser Bandkabel von links nach rechts mit A0...A12 und B0...B12 bezeichnet.

Aus den Zeilen 8...12 des Listings (Bild 2) geht hervor, wie man an den Anschlüssen PB1 und PB2 des EMUF die Baudrate auswählt. Das Programm ist modular aufgebaut, so daß es im großen und ganzen leicht zu durchschauen ist. Auf eine detaillierte Beschreibung wird deshalb verzichtet, und der folgende Text ist in Form einer Bedienungsanleitung geschrieben.

Die Bedienung: komfortabel und doch einfach

Alle Funktionen der Schreibmaschine bleiben erhalten, auch wenn das Interface angeschlossen und eingeschaltet ist (siehe Bedienungsanleitung der Schreibmaschine). Beim Betrieb als Drucker darf die Shift-Lock-Funktion nicht eingeschaltet sein, und der Zeichensatz-Wahlschalter muß auf KBII stehen. Beim Einschalten der Schreibmaschine werden verschiedene Parameter voreingestellt. Das sind:

- linker und rechter Rand für DIN-A4-Format,
- Tabulatorstopp alle 10 Zeichen.

Auch das Interface wird beim Einschalten auf diese Werte eingestellt. Mit bestimmten Steuerzeichen, die im folgenden noch näher erläutert werden, können sie jedoch verändert werden. Das Interface sorgt automatisch dafür, daß die Schreibmaschine ebenfalls auf die entsprechenden Tabulatorpositionen eingestellt wird. Schaltet man die Ma-

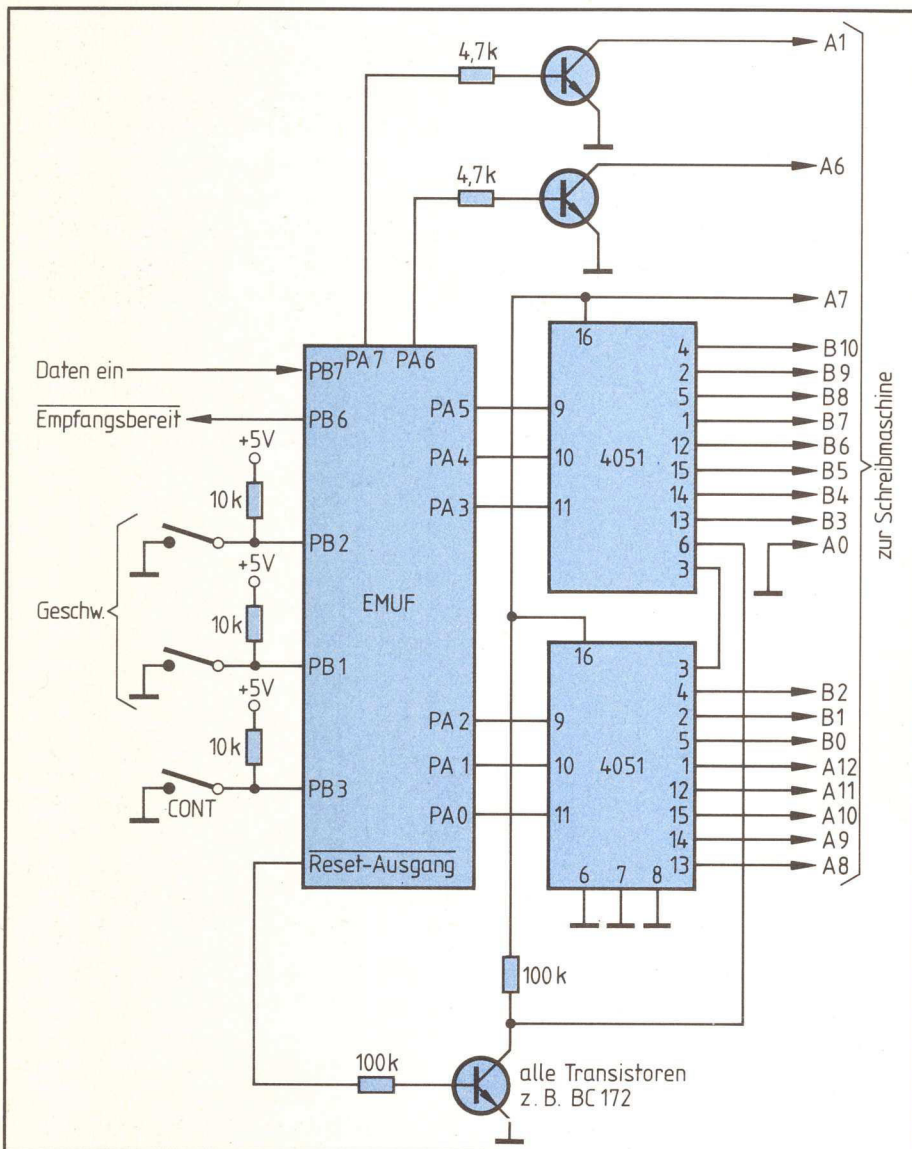


Bild 1. So wird die Tastaturmatrix der Olivetti-Schreibmaschine P30 mit der EMUF-Platine verbunden

Bild 2. Das EMUF-Programm zur Ansteuerung der Typenrad-Schreibmaschine. Prinzipiell ist natürlich auch eine Ansteuerung ohne EMUF möglich, indem man die entsprechenden Software-Teile in einen 6502-Tischcomputer implementiert

```

**END OF PASS 1
**END OF PASS 2

0800 1 ;V24-INTERFACE FUER PRIVILEG
0800 2 ;*****
0800 3 ;COPYRIGHT:
0800 4 ;RUDDOLF HOFER
0800 5 ;FRAUNHOFERSTR.22
0800 6 ;8260 MUEHLDORF/INN
0800 7 ;*****
0800 8 ;PB1/2 GESCHW. FUER V24
0800 9 ; 0 0 = 110 BD
0800 10 ; 1 0 = 300 BD
0800 11 ; 0 1 = 600 BD
0800 12 ; 1 1 = 1200 BD
0800 13 ;PB3 = ON LINE
0800 14 ;PB6=EMPFANGSBEREIT (0)
0800 15 ;PB7=DATENEINGANG
0800 16 ;=====
0800 17 ;PA = AUSGANG FUER SCHREIBMASCHINE
0800 18 ;*****
0800 19 ;
0800 20 ;RESET-VEKTOR FFC:00
0800 21 ;
0800 22 ; ORG #FFC
0800 23 ; HEX 000C
0800 24 ;
0800 25 ; ORG $C00
0800 26 ;*****
0800 27 PA EQU $800
0800 28 PB EQU $802
0800 29 DRG EQU $801
0800 30 DRB EQU $803
0800 31 WTIMER EQU $814
0800 32 RTIMER EQU $815
0800 33 FLAG EPZ 0
0800 34 XTEMP EPZ 1
0800 35 ZEIT1 EPZ 2
0800 36 ZEIT2 EPZ 3
0800 37 DIRFL EPZ 4
0800 38 NDEF EPZ 4
0800 39 LMAX EPZ $19
0800 40 CHMAX EPZ $1A
0800 41 LINCT EPZ $1B
0800 42 CHRCT EPZ $1C
0800 43 YTEMP EPZ $1D
0800 44 XTEM1 EPZ $1E
0800 45 YTEM1 EPZ $1F
0800 46 NEUD EPZ $20
0800 47 KFLAG EPZ $21
0800 48 TABUL EPZ $22
0800 49 LFD EQU $06
0800 50 CRD EQU $00
0800 51 DC1 EQU $11
0800 52 DC2 EQU $12
0800 53 PRESS EQU 140
0800 54 PAUSE EQU 173
0800 55 CLTD EQU $0B
0800 56 REPTD EQU $04

0C00 TSD EQU $10
0C00 TBD EQU $30
0C00 ;***** ;TABULATOR
0C00 ;INITIALISIERUNG
0C00 61 LDX #FF
0C00 62 TXS
0C00 63 CLD
0C00 64 SEI
0C00 65 STX DRG
0C00 66 LDA #0C
0C00 67 STA PA
0C00 68 LDA #40
0C00 69 STA DRB
0C00 70 LDA #0
0C00 71 STA NEUD
0C00 72 STA DIRFL
0C00 73 LDX #20
0C00 74 NUL STA NDEF,X
0C00 75 DEX
0C00 76 BNE NUL
0C00 77 STA CHRCT
0C00 78 STA LINCT
0C00 79 LDA #60
0C00 80 STA LMAX
0C00 81 LDA #60
0C00 82 STA CHMAX
0C00 83 LDX #11
0C00 84 LDA ITAB,X
0C00 85 STA TABUL,X
0C00 86 DEX
0C00 87 BPL NUL1
0C00 88 LDA #CRD
0C00 89 JSR ZAUS
0C00 90 LOOP JSR ZEIN
0C00 91 JSR ZAUS
0C00 92 JMP LOOP
0C00 93 ;*****
0C00 94 KTAB HEX 414243476A7172767702076A

0C00 95 KFILT LDX #0
0C00 96 PHA
0C00 97 AND #7F
0C00 98 VERGL CMP KTAB,X
0C00 99 BEQ FOUND
0C00 100 INX
0C00 101 CPX #12
0C00 102 BNE VERGL
0C00 103 CLC
0C00 104 PLA
0C00 105 RTS
0C00 106 FOUND SEC
0C00 107 PLA
0C00 108 RTS
0C00 109 ;=====
0C00 110 TABL JMP TABE
0C00 111 TBR JMP TABR

;C=1:DOFFELT
;BEL. Z. GEFUNDEN

```



```

OF12 113D39
OF15 3F292D
OF18 17
OF19 122321 433      HEX 1223211A251509220A0B1D3B3A13820C
OF1C 1A2515
OF1F 09220A
OF22 0B1D3B
OF25 3A1382
OF28 0C
OF29 434 ; #####
OF29 435 ; ZERO-PAGE-TABELLE
OF29 0A141E 436 ITAB HEX 0A141E283200
OF2C 283200
OF2F 000000 437      HEX 000000000000
OF32 000000
OF35 438      PAU
    
```

Tabelle 1. Zeichensatz (weitere Funktionen siehe Tab.3)

| HEX | DEZ | ZEICHEN | HEX | DEZ | ZEICHEN | HEX | DEZ | ZEICHEN |
|-----|-----|---------|-----|-----|---------|-----|-----|---------|
| 20 | 32 | | 40 | 64 | § | 60 | 96 | |
| 21 | 33 | ! | 41 | 65 | A | 61 | 97 | a |
| 22 | 34 | " | 42 | 66 | B | 62 | 98 | b |
| 23 | 35 | £ | 43 | 67 | C | 63 | 99 | c |
| 24 | 36 | \$ | 44 | 68 | D | 64 | 100 | d |
| 25 | 37 | % | 45 | 69 | E | 65 | 101 | e |
| 26 | 38 | & | 46 | 70 | F | 66 | 102 | f |
| 27 | 39 | ' | 47 | 71 | G | 67 | 103 | g |
| 28 | 40 | (| 48 | 72 | H | 68 | 104 | h |
| 29 | 41 |) | 49 | 73 | I | 69 | 105 | i |
| 2A | 42 | * | 4A | 74 | J | 6A | 106 | j |
| 2B | 43 | + | 4B | 75 | K | 6B | 107 | k |
| 2C | 44 | , | 4C | 76 | L | 6C | 108 | l |
| 2D | 45 | - | 4D | 77 | M | 6D | 109 | m |
| 2E | 46 | . | 4E | 78 | N | 6E | 110 | n |
| 2F | 47 | / | 4F | 79 | O | 6F | 111 | o |
| 30 | 48 | 0 | 50 | 80 | P | 70 | 112 | p |
| 31 | 49 | 1 | 51 | 81 | Q | 71 | 113 | q |
| 32 | 50 | 2 | 52 | 82 | R | 72 | 114 | r |
| 33 | 51 | 3 | 53 | 83 | S | 73 | 115 | s |
| 34 | 52 | 4 | 54 | 84 | T | 74 | 116 | t |
| 35 | 53 | 5 | 55 | 85 | U | 75 | 117 | u |
| 36 | 54 | 6 | 56 | 86 | V | 76 | 118 | v |
| 37 | 55 | 7 | 57 | 87 | W | 77 | 119 | w |
| 38 | 56 | 8 | 58 | 88 | X | 78 | 120 | x |
| 39 | 57 | 9 | 59 | 89 | Y | 79 | 121 | y |
| 3A | 58 | : | 5A | 90 | Z | 7A | 122 | z |
| 3B | 59 | ; | 5B | 91 | Ä | 7B | 123 | ä |
| 3C | 60 | (| 5C | 92 | Ö | 7C | 124 | ö |
| 3D | 61 | = | 5D | 93 | Ü | 7D | 125 | ü |
| 3E | 62 |) | 5E | 94 | — | 7E | 126 | ß |
| 3F | 63 | ? | 5F | 95 | µ | 7F | 127 | |

schine jedoch ab und das Interface nicht, dann stimmen die Tabulatorpositionen von Interface und Schreibmaschine nicht mehr überein.

In diesem Fall muß der Computer mit dem geeigneten Steuerzeichen die Tabulatorstopps neu einstellen. Soll breiteres Papier als DIN A4 verwendet werden, dann ist die Randbegrenzung vor Druckbeginn auf der Schreibmaschine per Hand einzustellen. Im Di-

rektmodus kann auch der Computer diese Einstellung ausführen.

Der Zeichensatz

Die vom Computer kommenden Codes werden vom Drucker gemäß *Tabelle 1* interpretiert. Bis auf wenige Ausnahmen entspricht das dem sog. ASCII-Zeichensatz. Um die Anpassung an verschiedene Computermodelle und unterschiedli-

che Programme zu erleichtern, können jedoch 10 Zeichen umdefiniert werden. Besonders nützlich ist das z. B. dann, wenn der Rechner statt CR,LF nur LF schickt. In diesem Fall definiert man einfach im Drucker, daß bei jedem eintreffenden LF die Funktion Carriage Return betätigt werden soll.

Zeichen- und Zeilenzahl einstellbar

Nach dem Einschalten ist der Drucker auf maximal 60 Zch./Zeile eingestellt. Vor dem 61. Zeichen wird automatisch die Zeilenschaltung betätigt (Zeilenabstand an der Schreibmaschine einstellen). Nach 60 Zeilen empfängt der Drucker keine Zeichen mehr, bis die CONT-Taste betätigt wurde. Dieser Wert kann per Steuerzeichen ebenso verändert werden wie die maximale Zahl der Zch./Zeile.

Steuerzeichen

Soll der Drucker alle eintreffenden Codes nach *Tabelle 1* interpretieren, maximal 60 Zch./Zeile zu Papier bringen und nach jeweils 60 Zeilen anhalten; und ist man außerdem damit zufrieden, daß der Tabulator fest an jeder zehnten Stelle gesetzt ist, dann benötigt man keine Steuerzeichen (außer Tabulatorsprung). Ist dies nicht der Fall, hat man folgende Möglichkeiten (*Tabelle 2*):

- Zahl der Zeichen/Zeile N einstellen (max. 110),
 - Zahl der Zeilen/Seite N einstellen,
 - Tabulatorpositionen N1, N2...N12 einstellen,
 - Druckeinheit zur nächsten Tabulatorposition,
 - ASCII-Code A so umdefinieren, daß Funktion D (*Tabelle 3*) der Schreibmaschine betätigt wird,
 - alle Parameter auf Anfangswert (wie nach dem Einschalten) zurücksetzen,
 - Funktion nach *Tabelle 3* betätigen.
- Die erste Einstellung ist denkbar einfach. Der Rechner schickt erst das Zeichen ESC (hex. 1B, dez. 27), dann den ASCII-Code für A (hex. 41, dez. 65) und danach eine Zahl zwischen 1 und 110 (hex. 1...6E). In BASIC sieht das folgendermaßen aus:

```
PRINT CHR$(27); "A"; CHR$(N);
```

wobei N entweder direkt als Zahl eingesetzt oder vorher definiert wird. Entsprechendes gilt für die Zahl der Zeilen/Seite. Achtung! Beide Parameter nur zu Beginn einer neuen Zeile verändern; Zeilen/Seite nur nach dem Einschalten oder nach ESC F. Bei der Tabulatoreinstel-

Tabelle 2. Steuerzeichen und ihre Bedeutung

| 1. Stz. | 2. Stz. | folg. Stz. | Funktion | Bemerkung |
|---------|---------|------------|---|---------------------|
| ESC | A | N | Zch./Zeile | N = Anzahl der Zch. |
| ESC | C | N | Zeilen/Seite | N = Anzahl der Zch. |
| ESC | D | N1...N12,0 | Tabulatorpos. | N beliebig bis 12 |
| ESC | B | A,D | Umdefinieren | A=ASCII/D=Druckerc. |
| ESC | E | | Parameter rücks. | |
| ESC | F | | Nächstes Z. im Direktmodus | |
| HT | | | Nächste Tab.-pos. HT = hex./dez. 09 | |
| FF | | | Drucker stoppt, weiter nach „CONT“ (FF = hex 0C/dez. 12) | |

Schalterstellung auf der Schreibmaschine unbedingt KB II

Tabelle 3. Druckerodes und zugehörige Tasten im Direktmodus

| ohne Shift | mit Shift | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------|-----------|------------|---|---|---|------------|---|----|---|
| Shift | Shift | 8 | 9 | A | B | C | D | E | F |
| 00...07 | 40...47 | ← | 9 | ^ | å | Rep. | 8 | CR | O |
| 08...0F | 48...4F | TAB CLEAR | v | x | y | | g | BL | c |
| 10...17 | 50...57 | TAB SET | i | p | ü | Shift Lock | u | ← | o |
| 18...1F | 58...5F | MAR. RIGHT | f | s | a | | z | | d |
| 20...27 | 60...67 | MAR. LEFT | r | w | q | | t | 7 | e |
| 28...2F | 68...6F | ↔ | m | . | - | ⊗ | n | b | , |
| 30...37 | 70...77 | TAB | 4 | 2 | 1 | | 6 | 5 | 3 |
| 38...3F | 78...7F | ⇔ | k | ö | ä | RELOC | j | h | 1 |

Bei Schalterstellung KB II können im Direktmodus nur die Symbole des 2. Zeichensatzes gedruckt werden

lung ist zu beachten, daß die Positionen in aufsteigender Reihenfolge einzugeben sind (N2 größer N1 usw.). Es können beliebig viele bis max. 12 eingestellt werden. Beispiel:

```
PRINT CHR$(27); "D"; CHR$(15);
CHR$(30); CHR$(0);
```

Die Null am Ende beendet die Steuerzeichenfolge, und das nächste Zeichen wird wieder normal interpretiert. Noch ein Beispiel für das Umdefinieren von Zeichen: Statt Å soll immer (gedruckt werden).

```
PRINT CHR$(27); "B"; CHR$(91);
CHR$(117);
```

Hierbei ist 91 das dez. Äquivalent des ASCII-Zeichens Å, und 117 ist der Druckercode von (in dez. Schreibweise.

Direktmodus für Sonderfunktionen

Nach der Steuerzeichenfolge ESC F wird das nächste Zeichen nach Tabelle 3 interpretiert. Damit sind auch sämtliche Funktionen der Schreibmaschine zugänglich, die nicht in Tabelle 2 aufgeführt sind, z. B. RELOC, Margin Left/Right, Wiederholtaste, Randüberschreitung und Shift Lock.

Literatur

- [1] Feichtinger, H.: Mädchen für alles. mc 1981, H. 2, S. 20...23.
- [2] Bedienungsanleitung zur Schreibmaschine Praxis-30, Olivetti.

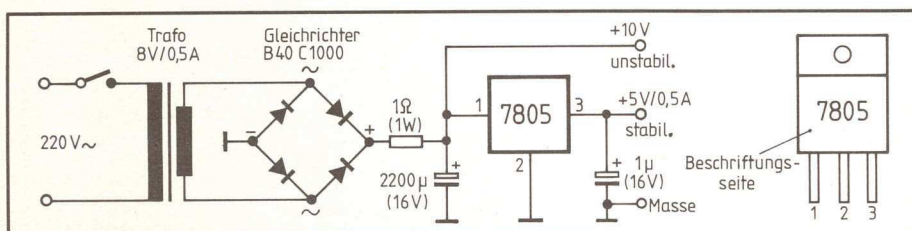
Ein Netzteil für den EMUF

Wenn man den Einplatinen-Computer EMUF mit einem EPROM-Typ bestückt, der mit nur einer Versorgungsspannung (+ 5 V) auskommt, so genügt für das ganze Gerät eine einfache Stromversorgung: Die EMUF-Platine selbst benötigt

rund 250 mA, und wenn man noch einige Leuchtdioden und zusätzliche ICs betreiben möchte, so dimensioniert man das Netzteil am besten für 5 V/0,5 A. Das Bild zeigt eine hierfür geeignete Schaltung. Das Regel-IC 7805 sollte man

auf eine Kühlfläche von wenigstens 5 × 5 cm² setzen; ein Rippenkühlkörper ist natürlich auch geeignet. Zusätzlich steht noch eine Ausgangsspannung von etwa 10 V unregelt zur Verfügung, z. B. um Relais, Lämpchen oder Lautsprecher-Treiberschaltungen zu versorgen (der EMUF darf an keinem seiner Anschlüsse diese Spannung erhalten!).

Es empfiehlt sich, alle peripheren Schaltungen, die direkt mit den I/O-Ports des EMUF verbunden sind, aus dem gleichen Netzteil zu versorgen, um zu vermeiden, daß sie Spannungen an die Ports liefern, während der EMUF noch keine Spannung erhält – dies könnte zu einer Beschädigung des 6532-Bausteins führen, da ein unzulässiger Ausgleichsstrom über die internen Schutzdioden fließt.



Einige wenige Bauelemente genügen, um den EMUF aus dem Netz mit Spannung zu versorgen. Statt des 7805 läßt sich ebensogut ein LM 309 als Regel-IC einsetzen

Herwig Feichtinger

Funkfern-schreib-Empfänger

In der folgenden Applikation dient der EMUF zum Empfang von Funkfern-schreib-Sendungen im Baudot-Code. Er benützt die Autokorrelation zum Decodieren der Töne und liefert ausgang-seitig sieben parallele ASCII-Bits nebst einem Strobe-Signal z. B. für ein Video-Interface oder einen Drucker.

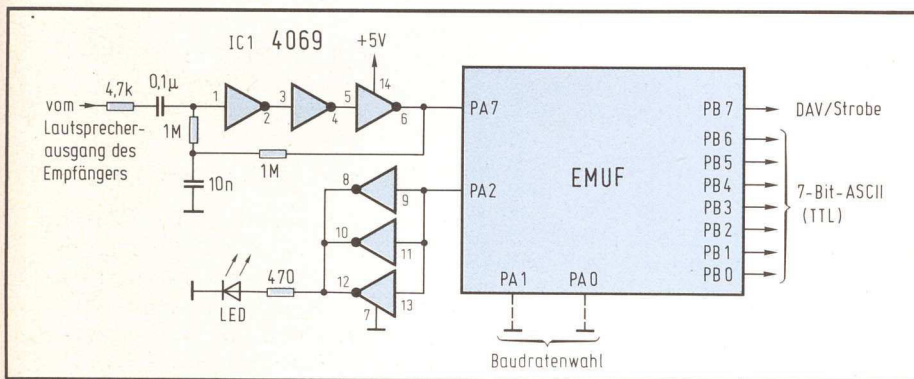


Bild 1. Externe Beschaltung des EMUF mit dem Funkfern-schreib-Empfangsprogramm im EPROM

Eine besondere Eigenschaft des hier vor-gestellten Funkfern-schreib-Decoders ist, daß kein Empfangskonverter im her-kömmlichen Sinn nötig ist, weil die Decodierung der Töne 2125 Hz für Mark (log. 1) und 1275 Hz für Space (log. 0) per Software geschieht [1]. Es ist deshalb lediglich erforderlich, am Port PA7 ein rechteckförmiges Nf-Signal mit TTL-Pe-gel anzuliefern; um den Rest kümmert

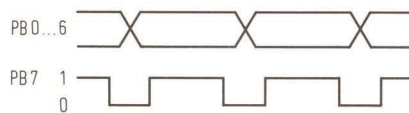


Bild 2. Zeitlicher Zusammenhang zwischen den ASCII-Datenausgängen PB 0...6 und dem DAV/Strobe-Ausgang PB 7

sich der EMUF. Die Empfindlichkeit der Decodierung entspricht derjenigen eines

guten PLL-Decoders; wenn vor dem Be-grenzer zusätzlich noch ein Doppelfilter geschaltet wird, das nur die beiden Fre-quenzen 1275 Hz und 2125 Hz mit einer Bandbreite von je etwa 100 Hz passieren läßt, wird die Empfindlichkeit eines Fil-terkonverters mühelos erreicht. Bild 1 zeigt die prinzipielle Beschaltung des EMUF.

In Bild 2 ist das Verhalten des Data-Valid/Strobe-Ausgangs PB7 in Bezug auf die ASCII-Daten an PB 0...6 zu sehen: Diese Leitung ist so lange auf High-Pe-gel, wie die Daten garantiert gültig sind. Sie kann als Strobe-Signal zur Daten-übernahme z. B. für ein Video-Interface, einen ASCII-Drucker mit Parallelschnitt-stelle oder das in diesem Heft beschrie-bene LCD-Interface verwendet werden. Bild 3 zeigt das vollständige Assembler-Listing.

Die in Bild 1 sichtbare Leuchtdiode dient zur korrekten Abstimmung des an-geschlossenen Empfangsgerätes. Wenn die Mark-Frequenz richtig eingestellt ist (Tonhöhen-Einstellung) bei Einseiten-band-Empfängern bzw. beim Telegrafie-Überlagerungsknopf), leuchtet sie auf; bei einem getasteten RTTY-Signal (RTTY = Radio Teletype, Funkfern-schreiben) blinkt sie im Takt der emp-fangenen Datenbits.

Auf die Möglichkeit der Mark/Space-Frequenzänderung oder Polaritätsum-

Bild 3. Assemblerlisting der EMUF-Software

```

0000 ;EMUF ALS RTTY-RX
0000 .OPT GEN ; F. TABELLE
0000 ;PB=ASCII-AUSGANG
0000 ;PB7=DAV/STROBE
0000 ;PA 0/1=BAUDRATE
0000 ;PA7=NF-EINGANG
0000 ;PA2=LED-AUSGANG
0000 PA = $800
0000 PAD = $801
0000 PB = $802
0000 PBD = $803
0000 BDR * = * + 1 ; BAUDRATE
0001 FLG * = * + 1 ; BU/ZI
0002 SUM * = * + 1 ; AUTOKORR.
0003 SPL1 * = * + 1 ; BIT-
0004 SPL2 * = $C00 ; MUSTER
0C00 RES A2FF LDX £$FF
0C02 8E0308 STX PBD ; PB=AUSG.
0C05 9A TXS
0C06 78 SEI
0C07 D8 CLD
0C08 E8 INX
0C09 8601 STX FLG ; BU-MODUS
0C0B A904 LDA £4
0C0D 8D0108 STA PAD
0C10 LP AD0008 LDA PA ; BAUDRATE
0C13 2903 AND £3 ; IN PA 0-1
0C15 AA TAX
0C16 BD100D LDA BDRT, X
0C19 8500 STA BDR
0C1B 20270C JSR ASC ; EMPFANG
0C1E 20BBOC JSR OUT ; AUSGABE
0C21 30ED BMI LP ; JUMP
0C23 ; BAUDOT ZU ASCII
0C23 LTR A900 LDA £0
0C25 FIG 8501 STA FLG
0C27 ASC 20430C JSR BDT ; EIN-
0C2A F0F7 BEQ LTR ; SPRUNG
    
```

| | | | | | | | | |
|------|--------|-----------------------|--------------------|------|------|--------|---------------------------------|------------|
| OC2C | C91F | CMP £\$1F | ;BU? | OCA5 | CHK | A007 | LDY £7 | ;8 BITS |
| OC2E | FOF3 | BEQ LTR | | OCA7 | CHK1 | 6A | ROR A | |
| OC30 | C91B | CMP £\$1B | ;ZI? | OCA8 | | B002 | BCS CHK2 | |
| OC32 | D004 | BNE *+6 | | OCAA | | C602 | DEC SUM | |
| OC34 | A920 | LDA £\$20 | | OCAC | CHK2 | 88 | DEY | |
| OC36 | DOED | BNE FIG | | OCAD | | 10F8 | BPL CHK1 | |
| OC38 | O501 | ORA FLG | | OCAF | ERR | A502 | LDA SUM | |
| OC3A | AA | TAX | ;CODE | OCB1 | | 6904 | ADC £4 | ;BEWERTUNG |
| OC3B | BDD10C | LDA TAB,X | ;WANDELN | OCB3 | | 8502 | STA SUM | ;ADDIEREN |
| OC3E | C940 | CMP £'§ | | OCB5 | | 68 | PLA | |
| OC40 | FOE1 | BEQ LTR | ;UNGUELTIG | OCB6 | | AA | TAX | ;X,Y,A |
| OC42 | 60 | RTS | | OCB7 | | 68 | PLA | ;RUECK- |
| OC43 | | ;BAUDOT-ZEICHEN LESEN | | OCB8 | | A8 | TAY | ;SPEICHERN |
| OC43 | BDT | A200 | LDX £0 | OCB9 | | 68 | PLA | |
| OC45 | | 8602 | STX SUM | OCBA | | 60 | RTS | |
| OC47 | | A600 | LDX BDR | OCBB | | | ;ASCII-AUSGABE | |
| OC49 | | CA | DEX | OCBB | OUT | 48 | PHA | |
| OC4A | BDTO | 20700C | JSR RD | OCBC | | AD0208 | LDA PB | ;ALTES |
| OC4D | | 2402 | BIT SUM | OCBF | | 297F | AND £\$7F | ;ZEICHEN |
| OC4F | | 30F2 | BMI BDT | OCC1 | | 8D0208 | STA PB | ;DAV=0 |
| OC51 | | CA | DEX | OCC4 | | 68 | PLA | |
| OC52 | | DOF6 | BNE BDTO | OCC5 | | 8D0208 | STA PB | ;NEUES |
| OC54 | | A205 | LDX £5 | OCC8 | | EA | NOP | ;ZEICHEN |
| OC56 | BDT1 | 20610C | JSR BITS | OCC9 | | EA | NOP | |
| OC59 | | 6A | ROR A | OCCA | | EA | NOP | |
| OC5A | | CA | DEX | OCCB | | 0980 | ORA £\$80 | |
| OC5B | | DOF9 | BNE BDT1 | OCCD | | 8D0208 | STA PB | ;DAV=1 |
| OC5D | | 4A | LSR A | OCDO | | 60 | RTS | |
| OC5E | | 4A | LSR A | OCDD | | | ;BAUDOT/ASCII-TAB. | |
| OC5F | | 4A | LSR A | OCDD | TAB | 4045 | .BYT '§E', \$A, 'A SIU', \$D | |
| OC60 | | 60 | RTS | OCD3 | | 0A | | |
| OC61 | | | ;BAUDOT-BIT LESEN | OCD4 | | 4120 | | |
| OC61 | BITS | A000 | LDY £0 | OCD6 | | 534955 | | |
| OC63 | | 8402 | STY SUM | OCD9 | | 0D | | |
| OC65 | | A400 | LDY BDR | OCDA | | 4452 | .BYT 'DRJNFCKTZLWHYPQOBG§' | |
| OC67 | BIT1 | 20700C | JSR RD | OCDC | | 4A4E | | |
| OC6A | | 88 | DEY | OCDE | | 4643 | | |
| OC6B | | DOFA | BNE BIT1 | OCEO | | 4B54 | | |
| OC6D | | 2602 | ROL SUM | OCE2 | | 5A4C | | |
| OC6F | | 60 | RTS | OCE4 | | 5748 | | |
| OC70 | | | ;AUTOKORRELATION | OCE6 | | 5950 | | |
| OC70 | | | ;T=1.1MS,F=2.19KHZ | OCE8 | | 514F | | |
| OC70 | RD | 48 | PHA | OCEA | | 424740 | | |
| OC71 | | 98 | TYA | OCED | | 4D58 | .BYT 'MXV§\$3', \$A, '- ', \$27 | |
| OC72 | | 48 | PHA | OCEF | | 5640 | | |
| OC73 | | 8A | TXA | OCF1 | | 4033 | | |
| OC74 | | 48 | PHA | OCF3 | | 0A | | |
| OC75 | | A010 | LDY £16 | OCF4 | | 2D20 | | |
| OC77 | RD1 | A207 | LDX £7 | OCF6 | | 27 | | |
| OC79 | RDO | CA | DEX | OCF7 | | 3837 | .BYT '87', \$D, '§4;' | |
| OC7A | | DOFD | BNE RDO | OCF9 | | 0D | | |
| OC7C | | AD0008 | LDA PA | OCFA | | 40343B | | |
| OC7F | | 0A | ASL A | OCFD | | 2C40 | .BYT ', §:(5+)2!6019?§\$. /=' | |
| OC80 | | 6603 | ROR SPL1 | OCFF | | 3A28 | | |
| OC82 | | 6604 | ROR SPL2 | ODO1 | | 352B | | |
| OC84 | | 88 | DEY | ODO3 | | 2932 | | |
| OC85 | | DOFO | BNE RD1 | ODO5 | | 2136 | | |
| OC87 | | AD0008 | LDA PA | ODO7 | | 3031 | | |
| OC8A | | 0904 | ORA £4 | ODO9 | | 393F | | |
| OC8C | | 8D0008 | STA PA | ODOB | | 4040 | | |
| OC8F | | A503 | LDA SPL1 | ODOD | | 2E2F3D | | |
| OC91 | | FO1C | BEQ ERR | OD10 | | | | |
| OC93 | | C9FF | CMP £\$FF | OD10 | | | ;BAUDRATEN-TABELLE | |
| OC95 | | FO18 | BEQ ERR | OD10 | BDRT | 14 | .BYT 20 | ;45 BD |
| OC97 | | 4504 | EOR SPL2 | OD11 | | 12 | .BYT 18 | ;50 BD |
| OC99 | | DOOA | BNE CHK | OD12 | | 0C | .BYT 12 | ;75 BD |
| OC9B | | A8 | TAY | OD13 | | 09 | .BYT 9 | ;100 BD |
| OC9C | | AD0008 | LDA PA | OD14 | | | *=\$FFC | ;RESET- |
| OC9F | | 29FB | AND £\$FB | OFFC | | 000C | .WOR RES | ;VEKTOR |
| OCA1 | | 8D0008 | STA PA | | | | | |
| OCA4 | | 98 | TYA | OFFE | | | ERRORS= 0000 | |

kehr wurde hier verzichtet, da dies normalerweise nicht nötig ist und nur wenige uneinsichtige Funkamateure noch mit vertauschtem Mark- und Space-Signal arbeiten [2]. Außerdem läßt sich die Vertauschung bei SSB-Empfängern auch durch Umschaltung auf das andere Seitenband (USB/LSB) erreichen. Die Space-Frequenz spielt dabei prinzipiell keine Rolle, weil nur 2125 Hz als Mark-Frequenz ausgewertet wird. Probleme

kann es nur bei zu geringem Shift, d. h. bei zu geringem Abstand zwischen Mark- und Space-Frequenzen geben. Die gewünschte Baudrate läßt sich durch entsprechende Beschaltung von PB 0 und PB 1 einstellen:

| | | | | |
|------|----|----|----|-----|
| Baud | 45 | 50 | 75 | 100 |
| PB 0 | 0 | 1 | 0 | 1 |
| PB 1 | 0 | 0 | 1 | 1 |

0 bedeutet hier Masse und 1 einen offenen Eingang oder eine Verbindung mit

+5 V. Da die Autokorrelations-Routine unabhängig von der Eingangsfrequenz eine nahezu konstante Laufzeit aufweist, konnte sie als Timing-Grundlage verwendet werden, so daß der 6532-Timer unbenutzt blieb.

Literatur

- [1] Tonerkennung per Software – Autokorrelation. mc 1981, Heft 4.
- [2] Amateurfunk-Sonderbetriebsarten: Funkfern schreiben. Funkschau 1980, Heft 23.

AIM steuert Typenrad-EMUF

Der Anschluß der Typenrad-Schreibmaschine P-30 an eine V24-Schnittstelle macht, wie schon beschrieben, keine Schwierigkeiten. Aber auch die Verbindung mit einem Computer, der „nur“ über einen 20-mA-Stromschleifen-Ausgang verfügt, ist leicht durchführbar.

Bild 1 zeigt ein kleines Hilfsprogramm für die Computer AIM-65 (Rockwell) und PC-100 (Siemens); mit der Funktionstaste F1 kann die Schreibmaschine eingeschaltet werden, so daß sie parallel zum Display mitläuft. F2 schaltet sie (natürlich stets vom Monitorprogramm aus) wieder aus, während F3 dem Monitorbefehl M entspricht – allerdings werden statt vier Bytes immer 16 Bytes pro Zeile ausgegeben, um die Papierbreite besser auszunützen.

Bild 2 zeigt die nötigen Hardware-Verbindungen zwischen AIM und EMUF: Nur vier Drähte genügen. Die Übertragung der Daten erfolgt mit 300 Baud, und PA6 dient beim AIM als Eingang für die Empfangsbereit-Meldung der Schreibmaschine. Das PC-100-Netzteil oder ein ausreichend dimensioniertes AIM-65-Netzteil ist leicht in der Lage, die zusätzlichen etwa 250 mA für den EMUF aufzubringen, so daß keine zusätzliche Stromversorgung nötig ist. Sollten beim Einschalten der Schreibmaschine unerwartet Probleme auftreten (z. B. die Schreibmaschinentastatur läßt keine Eingabe mehr zu), so ist folgende Reihenfolge einzuhalten:

```
( ) =OF90 A9 0C 8D 17 A4 A9 C2 8D 18 A4 A9 E8 A2 0F 8D 06
( ) OFA0 A4 8E 07 A4 60 A9 05 A2 EF D0 F3 20 AE EA B0 4D
( ) OFB0 A2 10 A0 00 20 3E E8 A9 1C 20 58 EB 20 46 EA C8
( ) OFC0 CA D0 F1 20 F0 E9 A9 3C 20 BC E9 20 93 E9 C9 20
( ) OFD0 F0 03 4C 8D E1 A0 10 20 CD E2 A9 3E 20 7A E9 20
( ) OFE0 3E E8 20 DB E2 4C B0 0F 2C 01 A0 70 FB 48 29 7F
( ) OFF0 20 A8 EE C9 OD D0 05 A9 0A 20 A8 EE 68 60 00 00
(M) =010C 4C 90 0F 4C
( ) 0110 A5 0F 4C AB
( ) 0114 0F
```

Bild 1. Hilfsprogramm zur Verbindung von AIM-65 oder PC-100 mit der V24-Schnittstelle des Schreibmaschinen-Interface

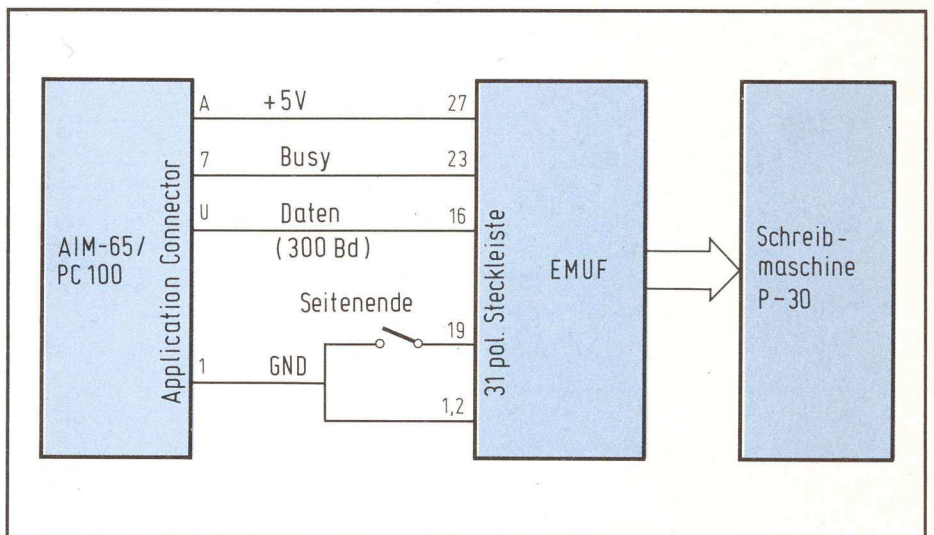


Bild 2. Nur vier Leitungen genügen, um den EMUF an den AIM-65/PC-100 anzuschließen und so die Typenrad-Schreibmaschine anzusteuern

1. Schreibmaschine und AIM-65, wenn nicht schon geschehen, einschalten.
2. Jetzt die Verbindung zwischen AIM und EMUF durch Zusammenstecken z. B. der 31-poligen Steckleiste herstellen.

Normalerweise ist diese Reihenfolge aber nicht erforderlich; nur in Einzelfällen verweigert der in der Schreibmaschine selbst befindliche F8-Prozessor den Einschaltreset.

Fe.

Rudolf Hofer

V.24-Schnittstellen- tester

Das Gerät testet V.24-Datenendgeräte auf ihre korrekte Funktion (speziell Drucker). Es simuliert eine Datenübertragungseinheit (Zentraleinheit, Modem) und gibt beim Auftreten von Fehlern Diagnosemeldungen ab. Alle wichtigen Parameter (Geschwindigkeit, mit/ohne Parität, gerade/ungerade Parität, 7/8 Datenbits) sind einstellbar.

Bild 1 zeigt den zeitlichen Ablauf, Bild 2 das Gesamtschaltbild. Es sind folgende Betriebsarten möglich:

1. Freilaufend ohne Quittung

Das Testgerät schickt laufend den Prüftext, ohne Quittungssignale zu beachten (keine Fehlermeldungen).

2. Freilaufend mit Quittung

Nach dem Einschalten zeigt LED 2 durch Blinken an, daß sie in Ordnung ist. Danach schickt das Testgerät den Prüftext, solange die EB-Leitung auf „1“ liegt. Ist die EB-Leitung länger als 0,25 s auf „0“, wird die Zeichenausgabe gestoppt, und LED 2 zeigt einen Fehler an (Tabelle 1).

3. Prozedur

Nach dem Einschalten zeigen beide LEDs durch Blinken an, daß sie in Ordnung sind. Danach schickt das Testgerät den Prüftext unter Beachtung aller im V.24-Standard festgelegten Quittungssignale. Bleibt eines der Signale aus, gibt

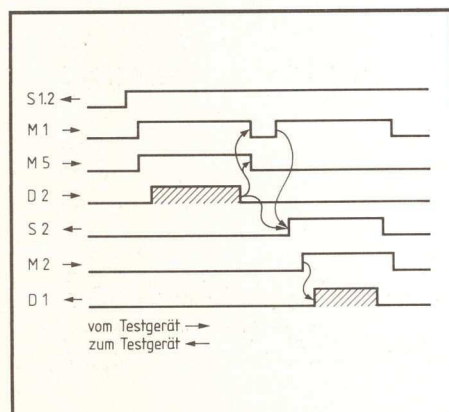


Bild 1. Zeitlicher Ablauf des V.24-Protokolls

es eine Fehlermeldung ab (siehe Tabelle 1). Wird ein Block mit NAK quittiert, blinkt LED 1 mehrmals, bevor der nächste Textblock geschickt wird. Bei allen

anderen Fehlermeldungen werden keine weiteren Daten ausgegeben.

In allen Betriebsarten beginnt der Test sofort nach dem Einschalten. Mit der Taste „Start“ kann aus jedem Zustand heraus ein neuer Test begonnen werden. Neue Parameter (siehe Tabelle 2) werden nur nach Drücken der Starttaste berücksichtigt. Bild 3 gibt das Assembler-Listing des EMUF-Programms wieder. Es beginnt bei der Adresse 6C00, die in 0C00 im EMUF dupliziert wird.

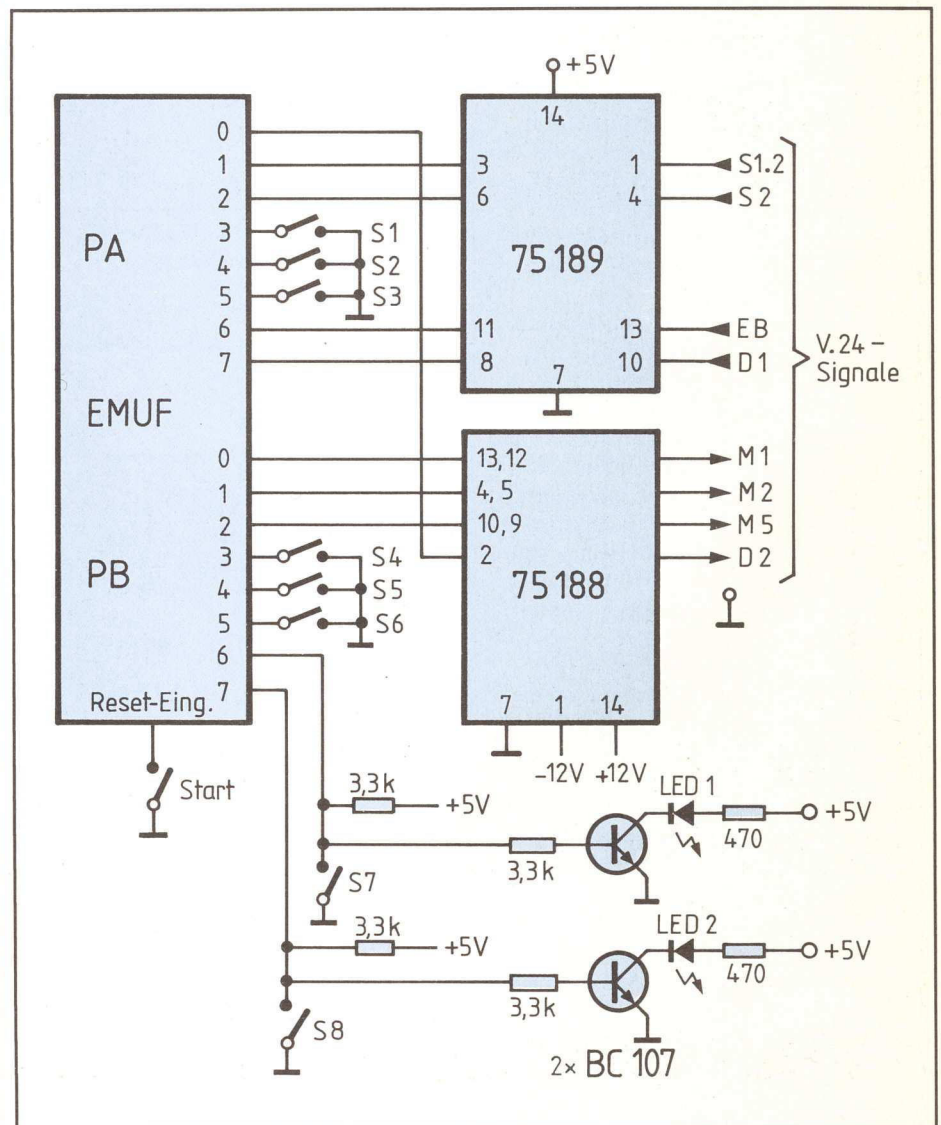


Bild 2. Schaltbild des V.24-Schnittstellentesters

Bild 3. Assemblerlisting der Software. Nachzutragen sind noch die Vektoren ab FFC: 00, 6C, C6, 6D für Reset und IRQ

```

0800 1 *****
0800 2 PAR EPZ 0
0800 3 BITS EPZ 1
0800 4 PARI EPZ 2
0800 5 ZEIT1 EPZ 3
0800 6 ZEIT2 EPZ 4
0800 7 XTEMP EPZ 5
0800 8 PTEMP EPZ 6
0800 9 BLMA0 EPZ 7
0800 1 BLMA1 EPZ 8
0800 1 FEHFL EPZ 9
0800 1 *****
0800 1 WTIMER EQU $814
0800 1 RTIMER EQU $815
0800 1 DDRA EQU $801
0800 1 DDRB EQU $803
0800 1 PA EQU $800
0800 1 PB EQU $802
0800 1 *****
0800 1 *****
0800 2 ORG $6C00
0800 2 LDX #0
0800 2 TXS
0800 2 JMP INIT
0800 2 SEND LDY #0
0800 2 NEUZ LDX BITS
0800 2 LDA PAR
0800 2 STA PTEMP
0800 2 JSR BEGA
0800 3 INY
0800 3 CPY #50
0800 3 BNE NEUZ
0800 3 RTS
0800 3 *****
0800 3 #UP AUSGABE EINES ZEICHENS
0800 3 BEGA JSR WAIT
0800 3 LDA #0
0800 3 STA PA
0800 3 LDA TEXTA,Y
0800 4 WIEDH PHA
0800 4 JSR WAIT
0800 4 PLA
0800 4 STA PA
0800 4 LSR
0800 4 BCC NOCRY
0800 4 INC PTEMP
0800 4 NOCRY DEX
0800 4 BNE WIEDH
0800 4 LDA PARI
0800 4 BEQ PRT
0800 5 LDA #1
0800 5 AND PTEMP
0800 5 PHA
0800 5 JSR WAIT
0800 5 PLA
0800 5 STA PA
0800 5 PRT LDA #1
0800 6C00 A2FF
0800 6C02 9A
0800 6C03 4C686C
0800 6C06 A000
0800 6C08 205C6C
0800 6C0B A601
0800 6C0D A500
0800 6C0F 8506
0800 6C11 201A6C
0800 6C14 C8
0800 6C15 C032
0800 6C17 D0F2
0800 6C19 60
0800 6C1A
0800 6C1A 20576C
0800 6C1D A900
0800 6C1F 800008
0800 6C22 B90E6E
0800 6C25 48
0800 6C26 20576C
0800 6C29 68
0800 6C2A 800008
0800 6C2D 4A
0800 6C2E 9002
0800 6C30 E606
0800 6C32 CA
0800 6C33 D0F0
0800 6C35 A502
0800 6C37 F00C
0800 6C39 A901
0800 6C3B 2506
0800 6C3D 48
0800 6C3E 20576C
0800 6C41 68
0800 6C42 800008
0800 6C45 A901
0800 6C47 48
0800 6C48 20576C
0800 6C4B 68
0800 6C4C 800008
0800 6C4F A901
0800 6C51 F003
0800 6C53 20576C
0800 6C56 60
0800 6C57
0800 6C57 2C1508
0800 6C5A 10FB
0800 6C5C A503
0800 6C5E 8605
0800 6C60 A604
0800 6C62 901408
0800 6C65 A605
0800 6C67 60
0800 6C68
0800 6C68 A2FF
0800 6C6A 78
0800 6C6B A901
0800 6C6D 800108
0800 6C70 800008
0800 6C73 A907
0800 6C75 800308
0800 6C78 A00208
0800 6C7B 0A
0800 6C7C 9006
0800 6C7E 0A
0800 6C7F 9055
0800 6C81 4C3C6D
0800 6C84 0A
0800 6C85 209A6C
0800 6C88 20066C
0800 6C8B 20286D
0800 6C8E 20286D
0800 6C91 20286D
0800 6C94 20286D
0800 6C97 4C886C
0800 6C9A
0800 6C9A A200
0800 6C9C 0A
0800 6C9D 9001
0800 6C9F E8
0800 6CA0 8602
0800 6CA2 A207
0800 6CA4 0A
0800 6CA5 9001
0800 6CA7 E8
0800 6CA8 8601
0800 6CAA A200
0800 6CAC 0A
0800 6CAD 9001
0800 6CAF E8
0800 6CB0 8600
0800 6C47 48 PHA
0800 6C48 20576C JSR WAIT
0800 6C4B 68 PLA
0800 6C4C 800008 STA PA
0800 6C4F A901 LDA #1
0800 6C51 F003 BEQ EINS
0800 6C53 20576C JSR WAIT
0800 6C56 60 RTS
0800 6C57 *****
0800 6C57 #UP WARTEN AUF TIMOUT
0800 6C57 2C1508 WAIT BIT RTIMER
0800 6C5A 10FB BPL WAIT
0800 6C5C A503 WAIT1 LDA ZEIT1
0800 6C5E 8605 WAIT12 STX XTEMP
0800 6C60 A604 LDX ZEIT2
0800 6C62 901408 STA WTIMER,X
0800 6C65 A605 LDX XTEMP
0800 6C67 60 RTS
0800 6C68 *****
0800 6C68 A2FF INIT LDX #0
0800 6C6A 78 SEI
0800 6C6B A901 LDA #1
0800 6C6D 800108 STA DDRA
0800 6C70 800008 STA PA
0800 6C73 A907 LDA #7
0800 6C75 800308 STA DDRB
0800 6C78 A00208 LDA PB
0800 6C7B 0A ASL
0800 6C7C 9006 BCC F0Q
0800 6C7E 0A ASL
0800 6C7F 9055 BCC FMQ
0800 6C81 4C3C6D JMP PROZ
0800 6C84 0A ASL
0800 6C85 209A6C JSR PARAM
0800 6C88 20066C JSR SEND
0800 6C8B 20286D JSR SEK/4
0800 6C8E 20286D JSR SEK/4
0800 6C91 20286D JSR SEK/4
0800 6C94 20286D JSR SEK/4
0800 6C97 4C886C JMP WFOQ
0800 6C9A *****
0800 6C9A #PARAMETER IN STEUERBLOCK
0800 6C9A A200 PARAM LDX #0
0800 6C9C 0A ASL
0800 6C9D 9001 BCC ZMEI
0800 6C9F E8 INX
0800 6CA0 8602 STX PARI
0800 6CA2 A207 LDX #7
0800 6CA4 0A ASL
0800 6CA5 9001 BCC SIEB
0800 6CA7 E8 INX
0800 6CA8 8601 STX BITS
0800 6CAA A200 LDX #0
0800 6CAC 0A ASL
0800 6CAD 9001 BCC GRD
0800 6CAF E8 INX
0800 6CB0 8600 STX PAR
0800 *****
0800 #STATUS B EINLESEN
0800 #1 BLOCK
0800 #2 STOPPBITS?
0800 #DATENBITS
0800 #PARITAEI

```



```

6CB2 A00008 1 LDA PA          #GESCHWINDIGKEIT
6CB5 4A      1 LSR
6CB6 4A      1 LSR
6CB7 4A      1 LSR
6CB8 2907   1 AND #7
6CBA AA     1 TAX
6CBB B0C66C 1 LDA TAB,X
6CBE 8503   1 STA ZEIT1
6CC0 B0CE6C 1 LDA TAB+8,X
6CC3 8504   1 STA ZEIT2
6CC5 60     1 RTS
6CC6        1 #TABELLE FUER GESCHWINDIGKEIT
6CC6 8C34C 1 TAB HEX 8C34C6233C15322
6CC9 6233C15322
6CCE 020201 1
6CD1 0101000000
6CD6        1 #*****
6CD6 209A6C 1 JSR PARAM
6CD9 A9C7   1 LDA #C7
6CD8 80308 1 STA DRB
6CDE A980   1 LDA #80
6CE0 8508   1 STA BLMA1
6CE2 A977   1 LDA #77
6CE4 8507   1 STA BLMA0
6CE6 200F60 1 JSR BLINK
6CE9 A000   1 NEUM
6CEB 205C6C 1 NEUZM
6CEE A601   1
6CF0 A500   1 LDA PAR
6CF2 8506   1 STA PTEMP
6CF4 201A6C 1 JSR BEGA
6CF7 A902   1 LDA #2
6CF9 8509   1 STA FEHFL
6CFB 203560 1 JSR FTIME
6CFE 2C0008 1 BIT PA
6D01 70FB   1 BVS EB
6D03 78     1 SEI
6D04 C8     1 INY
6D05 C031   1 CPY #49
6D07 D0E2   1 BNE NEUZM
6D09 202860 1 JSR SEK/4
6D0C 4CE96C 1 JMP NEUM
6D0F        1 #*****
6D0F A205   1 BLINK LDX #5
6D11 A0208 1 BL
6D14 0508   1 ORA BLMA1
6D16 800208 1 STA PB
6D19 202860 1 JSR SEK/4
6D1C 2507   1 AND BLMA0
6D1E 800208 1 STA PB
6D21 202860 1 JSR SEK/4
6D24 CA     1 DEX
6D25 D0EA   1 BNE BL
6D27 60     1 RTS
6D28        1 #*****
6D28 48     1 SEK/4
6D29 A9FF   1 LDA #$$F
#GESCHWINDIGKEIT
LDA PA
LSR
LSR
LSR
AND #7
TAX
LDA TAB,X
STA ZEIT1
LDA TAB+8,X
STA ZEIT2
RTS
#TABELLE FUER GESCHWINDIGKEIT
TAB HEX 8C34C6233C15322
HEX 0202010101000000
#*****
JSR PARAM
LDA #C7
STA DRB
LDA #80
STA BLMA1
LDA #77
STA BLMA0
JSR BLINK
NEUM
NEUZM
#*****
LDA PAR
STA PTEMP
JSR BEGA
LDA #2
STA FEHFL
JSR FTIME
BIT PA
BVS EB
SEI
INY
CPY #49
BNE NEUZM
JSR SEK/4
JMP NEUM
#*****
BLINK LDX #5
BL
ORA BLMA1
STA PB
JSR SEK/4
AND BLMA0
STA PB
JSR SEK/4
BNE BL
RTS
#*****
SEK/4
LDA #$$F
#2 LEDS AUSGANG
#M1,2,5=0
#BEIDE LEDS BLINKEN
#MASKE FUER S12
#S12=1?
#FEHLERCODE 2
#M1 UND M5
#M1,5=0
#S2=1?
#M2 (M1 UND M5)
#D1 EMPFANGEN
#ACK?
#NACK?

```

```

6E0A 20286D 2 JSR SEK/4
6E0D 60 RTS
6E0E 02 TEXTA HEX 02
6E0F D4C8C5 2 ASC "THE QUICK BROWN FOX
        JUMPS OVER THE LAZY DOG"
6E12 A0D1D5C9C3C8A0C2D2CF07CEA0C6CFD8A0CAD5CDD0D3
        A0CFD6C5D2A0D4C8C5A0CC1DAD9A0C4CFC7
6E3A 0A0D03 2 HEX 0A0D03
    
```

```

6DA1 F003 BEQ NACK1 #WEDER ACK NOCH NACK
6DA3 4CAC6D JMP KACKNA #NACK ANZEIGEN
6DA6 20BA6D 2 NACK1 #BLOCK WIDERHOLEN NACH NACK
6DA9 4C546D 2 #*****
6DAC ***** #LED2 BLINKT
6DAE 8508 KACKNA LDA #80
6DB0 A977 STA BLMA1
6DB2 8507 LDA #77
6DB4 200F6D STA BLMA0
6DB7 4C646D LOOP JSR BLINK
6DBA A940 FNACK LDA #40
6DBE A9B7 STA BLMA1
6DC0 8507 LDA #B7
6DC2 200F6D STA BLMA0
6DC5 60 JSR BLINK
6DC6 ***** RTS
6DC6 #INTERRUPT-ROUTINE *****
6DC6 EE1408 IRQ INC WTIMER
6DC9 A509 LDA FEH1
6DCB C901 CMP #1
6DD0 F009 BEQ FEH1
6DDC F009 CMP #2
6DDF F00B BEQ FEH2
6DD3 C903 CMP #3
6DD5 F00D BEQ FEH3
6DD7 40 RTI
6DD8 ***** #FEHLERBEHANDLUNG *****
6DD8 #FEHLERBEHANDLUNG
6DD8 A940 FEH1 LDA #40
6DDA 80D208 STA PB
6DDC 40 RTI
6DDE A980 FEH2 LDA #80
6DE0 80D208 STA PB
6DE3 40 RTI
6DE4 A9C0 FEH3 LDA #C0
6DE6 80D208 STA PB
6DE9 40 RTI
6DEA ***** #BEIDE LEDS PERMANENT *****
6DEA #EMPfang DER BESTAETIGUNG *****
6DEA 20356D EMPF JSR FTIME
6DED 2C0008 EMP BIT PA
6DF0 30FB BMI EMP
6DF2 78 SEI
6DF3 A209 LDX #9
6DF5 A910 LDA #10
6DF7 2D5E6C JSR WAIT12
6DFA A900 LDA #0
6DFC 48 PHA
6DFD 20576C JSR WAIT
6E00 68 PLA
6E01 0E0008 ASL PA
6E04 6A ROR
6E05 CA DEX
6E06 D0F4 BNE NBIT
6E08 291F AND #1F
    
```

Tabelle 1: Fehlermeldungen des Testers

| LED 1 | LED 2 | Prozedur | Freilaufend |
|-------|--------|----------------|-------------|
| 1 | 0 | S 1.2 fehlt | |
| 0 | 1 | S 2 fehlt | EB fehlt |
| 0 | blinkt | kein ACK/NACK | |
| 0 | 0 | NAK | |
| 1 | 1 | keine Quittung | |

Tabelle 2: Einstellen der Parameter

| Schalterstellungen | | 110 Bd | 300 Bd | 600 Bd | 1200 Bd | 2400 Bd | 4800 Bd | 9600 Bd |
|--------------------|--------------------------|---------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|----------|
| off | on | on | off | on | off | on | off | on |
| S 1 | Parität gerade | on | off | on | off | on | off | on |
| S 2 | 8 Datenbits | on | on | off | off | on | on | off |
| S 3 | 7 Datenbits | on | on | on | on | off | off | off |
| S 4 | Parität ungerade | on | on | on | on | off | off | off |
| S 5 | 8 Datenbits mit Parität | on | on | on | on | off | off | off |
| S 6 | 7 Datenbits ohne Parität | on | on | on | on | off | off | off |
| S 7 | | x | on | on | off | off | off | off |
| S 8 | | on | off | off | off | off | off | off |
| | | Freilaufend ohne Quittung | Freilaufend mit Quittung | Freilaufend mit Quittung | Freilaufend mit Quittung | Freilaufend mit Quittung | Freilaufend mit Quittung | Prozedur |

Michael Richter

EMUF als DCF-77-Decoder

DCF 77 heißt ein Sender, der von der Physikalisch-Technischen Bundesanstalt in Braunschweig (PTB) auf der Langwellenfrequenz 77,5 kHz betrieben wird. Er sendet, binär mit Impulsen amplitudenmoduliert, Uhrzeit und Datum atomuhrgenau aus. Diese Uhrzeit wird in ganz Deutschland als Normalzeit verwendet. Der Einplatinen-Computer EMUF dient hier als Decoder für die empfangenen Impulse.

Außer dem EMUF selbst wird in dieser Applikation ein geeigneter Empfänger für 77,5 kHz benötigt. Ein solcher Empfänger kann z. B. – wenn auch ohne Demodulator – von der Fa. Völkner (Braunschweig) bezogen werden; eine Schaltung dafür wurde auch in FUNKSCHAU 1979, Heft 14, Seite 841, beschrieben. An dieser Stelle wollen wir uns daher auf die Beschreibung des Digitalteils beschränken.

Bild 1 zeigt die externe Beschaltung der EMUF-Platine, sie ist in erster Linie für die Ansteuerung der sechs Siebensegment-Displays erforderlich, auf denen die Uhrzeit in Stunden, Minuten und Sekunden erscheint.

Das Programm, in Bild 2 hexadezimal aufgelistet (es wurde ohne Assembler geschrieben), hat leicht in einem KByte Platz. Die Tabelle zur Siebensegment-Codierung steht ab der Adresse 0F0F. Bei Adresse 0E1D beginnt die Initialisierung. Die Entwicklung erfolgte ursprünglich im Adressbereich ab 2C00, was für den EMUF jedoch belanglos ist. Rund zwei Minuten nach dem Einschalten sollte die Uhrzeit im Display erscheinen. Auf der EMUF-Rasterfläche läßt sich zusätzlich eine Empfangs-Ausfallanzeige aufbauen, die nach ein bis zwei Minuten aufleuchtet, wenn kein ausreichendes Signal vorhanden ist; dieser Schaltungsteil darf aber auch weggelassen werden (Bild 3). Statt des 10- μ F-Tantal-Kondensators empfiehlt sich allerdings eine bipolare oder MKL-Ausführung, da an ihm eine Spannung wechselnder Polarität anliegt, wenn das Monoflop kippt.

Es sei gleich vermerkt, daß sich das Programm noch geringfügig verkürzen ließe; in Anbetracht des ausreichenden Speicherplatzes wurde aber hierauf verzichtet. Das mit ausgesendete Parity-Bit wird hier mit ausgewertet und dient der Erkennung von Übertragungsfehlern zu verhindern, daß Unsinn angezeigt wird. Ist ein Fehler erkannt worden, läuft die Uhr mit „eigenem Takt“ weiter, bis wieder eine gültige Zeitinformation decodiert werden kann.

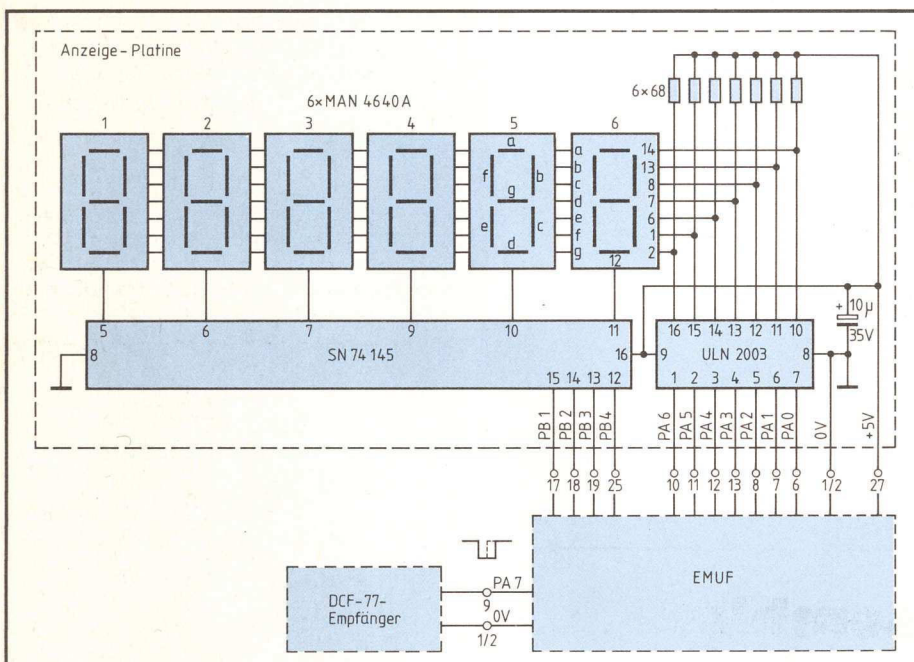


Bild 1. Schaltung zur Ansteuerung eines gemultiplexten sechsstelligen Siebensegment-Displays; der entsprechende Programmteil steht im Adressbereich 0D8E...0DF8

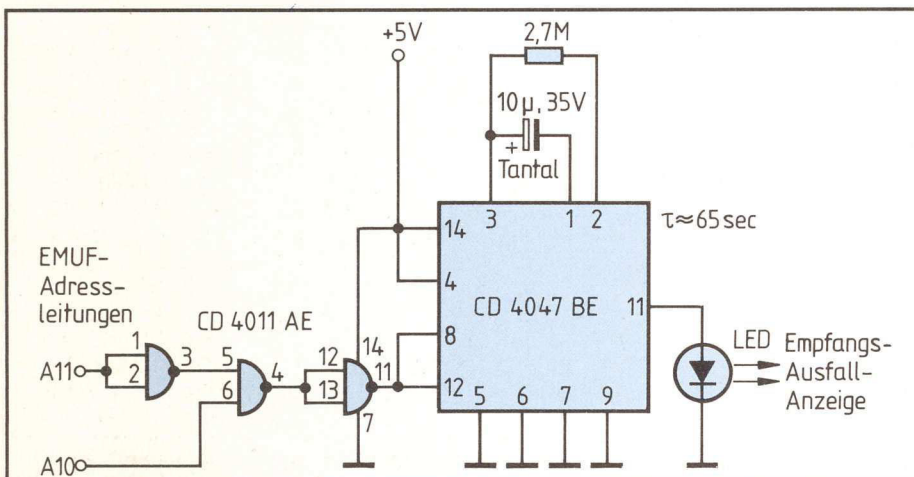


Bild 3. Eine kleine Zusatzschaltung zur Erkennung eines Empfangsausfalls hat noch auf der freien Lochrasterfläche der EMUF-Platine Platz

Bild 2. Hex-Dump des DCF-77-Empfangsprogramms. Da die Software „zu Fuß“ entwickelt wurde, steht diesmal kein Assemblerlisting zur Verfügung

```
( )=OC00 A9 EF 8D 3E 20 A9 2C 8D 3F 20 A2 FF 9A 8D 05 28
( ) OC10 58 20 E5 2C 00 EA EA A2 00 A0 00 2C 15 28 50 4D
( ) OC20 A2 00 E8 C8 20 35 2D AD 00 28 10 EB E8 C8 20 35
( ) OC30 2D C0 30 F0 07 AD 00 28 10 F3 30 F0 E0 24 A9 00
( ) OC40 6A A4 00 C0 20 30 17 F0 79 AA 10 02 E6 04 A6 03
( ) OC50 C0 28 F0 75 C0 35 F0 6F 18 15 05 6A 95 05 20 41
( ) OC60 2D F8 18 98 69 01 A8 84 00 D8 4C 17 2C 20 35 2D
( ) OC70 E8 E0 30 D0 A6 A5 00 C9 59 D0 62 20 41 2D F8 38
( ) OC80 A5 05 F0 2A E5 08 C9 01 D0 4B A5 06 E5 09 D0 45
( ) OC90 D8 85 0B A5 05 85 08 85 0C A5 06 85 09 85 0D A2
( ) OCA0 07 20 E7 2C A9 09 85 0A 8D 00 24 4C 17 2C A5 08
( ) OCBO 49 59 D0 21 A5 06 D0 02 A9 24 E5 09 49 01 D0 15
( ) OCC0 F0 CE AA F0 18 D0 97 56 05 46 04 B0 10 A9 00 85
( ) OCD0 04 E6 03 D0 89 A5 05 85 08 A5 06 85 09 A2 07 20
( ) OCE0 E7 2C 4C 17 2C A2 09 A9 00 95 00 CA 10 FB 60 48
( ) OCF0 98 48 8A 48 C6 0A F0 07 A9 7A 8D 1F 28 D0 21 A9
( ) OD00 03 8D 1E 28 A2 00 F8 20 2B 2D D0 0F 20 26 2D D0
( ) OD10 0A 20 26 2D C9 24 D0 03 20 26 2D D8 A9 09 85 0A
( ) OD20 68 AA 68 A8 68 40 A9 00 95 0B E8 18 B5 0B 69 01
( ) OD30 95 0B C9 60 60 86 01 84 02 20 8E 2D A6 01 A4 02
( ) OD40 60 A2 B0 20 35 2D CA D0 FA 60 FF FF FF FF FF FF

( )=OD80 FF FF FF FF FF FF FF FF FF FF FF FF FF FF A9 7F
( ) OD90 8D 01 28 A2 08 A4 16 A5 0D 20 CC 2D 88 F0 0D A5
( ) ODA0 0C 20 CC 2D 88 F0 05 A5 0B 20 CC 2D A9 00 8D 01
( ) ODB0 28 A0 03 A2 00 A9 FF 8E 02 28 E8 E8 2D 00 28 88
( ) ODC0 D0 F5 A0 06 8C 02 28 09 80 49 FF 60 48 84 1C 4A
( ) ODD0 4A 4A 4A 20 DF 2D 68 29 0F 20 DF 2D A4 1C 60 A8
( ) ODE0 B9 0F 2F 8D 00 28 8E 02 28 A0 7F 88 10 FD 8C 00
( ) ODF0 28 A0 06 8C 02 28 E8 E8 60 FF FF FF FF FF FF
( ) OEO0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
( ) OE10 FF FF FF FF FF FF FF FF FF FF FF FF FF A9 1E 8D
( ) OE20 03 28 A9 03 85 16 A2 FF 9A D8 78 4C 00 2C FF FF
( ) OE30 FF FF 6C 3E 20 00 2C FF FF FF FF FF 6C 6E 20 FF

( )=OF00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 40
( ) OF10 79 24 30 19 12 02 78 00 10 08 03 46 21 06 0E FF

( )=OFF0 FF FF FF FF FF FF FF FF FF FF FF FF FF 1D 2E 32 2E
```

Formatierte Assemblerlistings mit dem AIM-65

Die Mikrocomputer AIM-65 und PC-100 besitzen einen recht brauchbaren symbolischen Assembler, der dank einiger besonderer Editor-Funktionen sogar eine Art Makro-Verarbeitung zuläßt. Allerdings ist der von ihm auf dem 20stelligen Thermodrucker gelieferte Kontrollausdruck etwas unübersichtlich formatiert. Das hier vorgestellte Hilfsprogramm übernimmt eine Umformatierung, so daß ein Ausdruck entsteht, wie er in diesem Heft mehrmals als Assemblerlisting zu sehen ist.

Bild 1 zeigt einen kurzen Ausschnitt aus dem „normalen“ AIM-65-Assemblerformat. Und in Bild 2 ist als Hex-Dump das Umformatier-Programm zu sehen, das zusammen mit dem EMUF-Interface für die Typenrad-Schreibmaschine P-30 betrieben werden kann, da auch die

Handshake-Prozedur eingehalten wird (Busy-Leitung). Die Verbindung zwischen den beiden Geräten ist in diesem Heft unter dem Titel „AIM steuert Typenrad-EMUF“ beschrieben. Die Zeichenausgabe erfolgt stets über die V24-Schnittstelle des AIM (bzw. TTY-Schnittstelle) mit 300 Baud.

Die Bedienung des Assemblers ändert sich nur insofern, daß nach LIST-OUT die Taste U gedrückt werden muß. Typisch ist der Ablauf z. B. wie folgt nach Drücken von N:

```
ASSEMBLER
FROM=E00 TO=F00
IN=M
LIST?Y
LIST-OUT=U
OBJ?Y
OBJ-OUT=X
```

In diesem Fall wurde auf die Erzeugung von Objektcode verzichtet (X=Dummy), um zu verhindern, daß das Objektprogramm eventuell das ab 0F00 stehende Formatierprogramm überschreibt. Fe.

Bild 1. Dieses Format liefert normalerweise der AIM-65-Assembler auf seinem 20stelligen Thermodrucker. Das Programm in Bild 2 formatiert es so um, wie das in den anderen Listings dieses Hefts zu sehen ist

```
==0000
*=$C20
==OC20 DELY
A228 LDX £40
;DELAY
==OC22 LOOP
CA DEX
;CA.0.2MS
DOFD BNE LOOP
60 RTS
;READY
.END
```

```
(M)=010A 0B 0F (User-Vektor)
( )=OF00 A9 0C 8D 17 A4 A9 C2 8D 18 A4 60 90 F3 86 D4 A2
( ) OF10 1E 86 D6 68 85 D3 29 7F C9 0D D0 05 A2 01 86 D2
( ) OF20 60 C9 3D D0 1A A6 D2 F0 0A 20 D0 0F A2 01 86 D1
( ) OF30 4C 9F 0F A6 D1 D0 03 4C 9C 0F 46 D1 4C 9F 0F A6
( ) OF40 D2 F0 3F C9 3B D0 28 46 D2 A2 01 86 D5 4C 9F 0F
( ) OF50 A6 D0 E0 1D 90 05 8A 69 02 85 D6 A6 D0 E4 D6 F0
( ) OF60 06 20 9A 0F 4C 5B 0F A9 3B 20 9C 0F 4C 7A 0F A6
( ) OF70 D0 E0 0B B0 0A 46 D2 20 C6 0F A5 D3 4C 9C 0F 20
( ) OF80 AE 0F A5 D3 A6 D5 F0 0E 46 D5 C9 2E D0 C2 20 AE
( ) OF90 0F A9 3B 4C 9C 0F C9 7D D0 02 A9 20 20 A2 0F A6
( ) OFA0 D4 60 A6 D0 E0 4F B0 03 20 F8 0F E6 D0 60 20 D0
( ) OFB0 0F 20 B8 0F 20 C6 0F 60 A5 33 20 E1 0F A5 32 20
( ) OFC0 E1 0F 20 9A 0F 60 20 9A 0F A6 D0 E0 0B D0 F7 60
( ) OFD0 A9 0D 20 F8 0F A9 0A EA EA EA A9 00 85 D0 85 D2
( ) OFE0 60 48 4A 4A 4A 4A 20 EC 0F 68 29 0F 18 69 30 C9
( ) OFF0 3A 90 02 69 06 4C 9C 0F 2C 01 A0 70 FB 4C A8 EE
```

Bild 2. Hex-Dump des Umformatierprogramms. Es handelt sich um eine abgeänderte Version eines Programms von H. Steder (1979). Soll ein Kommentarfeld nicht hinter dem Sourcecode, sondern gleich nach dem Adressenfeld im Listing erscheinen, so ist im Editor nach dem Strichpunkt ein Punkt einzugeben (also ; statt); das Programm erkennt dies automatisch

Rolf-Dieter Klein

V24-Interface

Viele CBM-Besitzer haben sich sicher schon lange eine Schaltung gewünscht, mit der sie z. B. einen V24-Drucker an den CBM über den IEC-Bus anschließen können. Durch eine Softwarelösung mit dem EMUF ist dies nun Wirklichkeit geworden. Dabei sind IEC-Primäradresse und Baudrate über den IEC-Bus programmierbar.

In Bild 1 ist der prinzipielle Anschluß gezeigt. Das EMUF-Interface ist mit dem IEC-Bus gekoppelt. Der Drucker wird über einen Pegelwandler an den seriellen Ausgang angeschlossen. Eine Rückmeldeleitung BUSY oder CTS (Clear To Send) erlaubt es, die serielle Übertragung zu stoppen, wenn z. B. der Drucker gerade beschäftigt ist. Bild 2 zeigt die genaue Anschlußbelegung des EMUF. Der IEC-BUS wird dabei genauso wie bei dem Bar-Code-Leser [1] angeschlossen. Es bleiben dann noch drei Leitungen übrig, die für die serielle Schnittstelle verwendet werden können. An PB7 wird die Rückmeldung angeschlossen. Der EMUF gibt nur dann Daten aus, wenn der CTS-Eingang (22) auf Low-Pegel liegt. Es ist eine Transistorstufe eingezeichnet, so daß der Eingang CTS auch mit $\pm 12V$ -Pegeln versorgt werden kann. Das Interface ist frei, wenn dort ein High-Pegel anliegt. An PB6 (23) erscheinen die seriellen Daten. Dabei liegt an diesem Pin noch ein negiertes Signal an, hinter einer weiteren Transistorstufe erscheint dann ein TTL-kompatibles Signal, mit einem High als Ruhepegel. PB5 schließlich wird verwendet, um eine wahlweise Umrechnung von dem CBM-Zeichencode in ASCII zu erreichen. Ist der Eingang auf einem Low-Pegel, also der Schalter geschlossen, so werden die Daten direkt übernommen. Damit ist auch ein Anschluß z. B. an HP-Rechner möglich, oder die Übertragung binärer Daten mit dem CBM.

Bild 3 zeigt eine Schaltung zur Pegelumsetzung von TTL (open coll.) auf V24-Pegel ($\pm 12V$). Die Schaltung wird direkt an den Transistor-Ausgang der EMUF-Schaltung angeschlossen. Am Ausgang des Pegelumsetzers ist der Ruhepegel auf $-12V$. Damit können Standard-V24-Geräte betrieben werden. Bild 4 zeigt das Programm-Listing. Die IEC-Routinen entsprechen denen aus [1]. Neu sind die Serial-Routinen. Der Ablauf ist dabei wie folgt: Nach dem Einschalten des EMUF wird dieser auf 1200-Baud eingestellt. Es werden dann die Zeichen CR und LF zu Testzwecken ausgegeben. Die erste Primäradresse, die auf dem IEC-Bus erscheint, wird genommen und um eins erhöht. Diese neue Adresse ist die Geräteadresse, die in Zukunft verwendet wird. Damit ist der

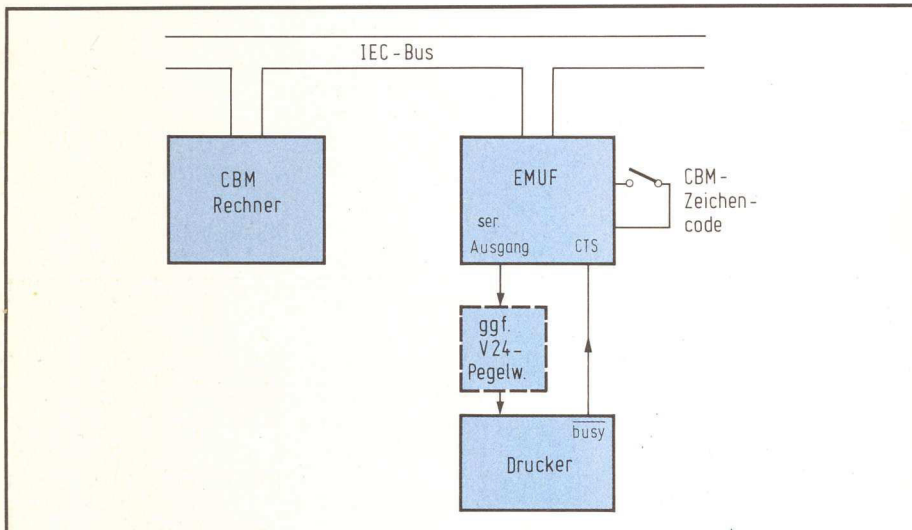


Bild 1. Anschluß eines V24-Druckers an einen CBM-Rechner

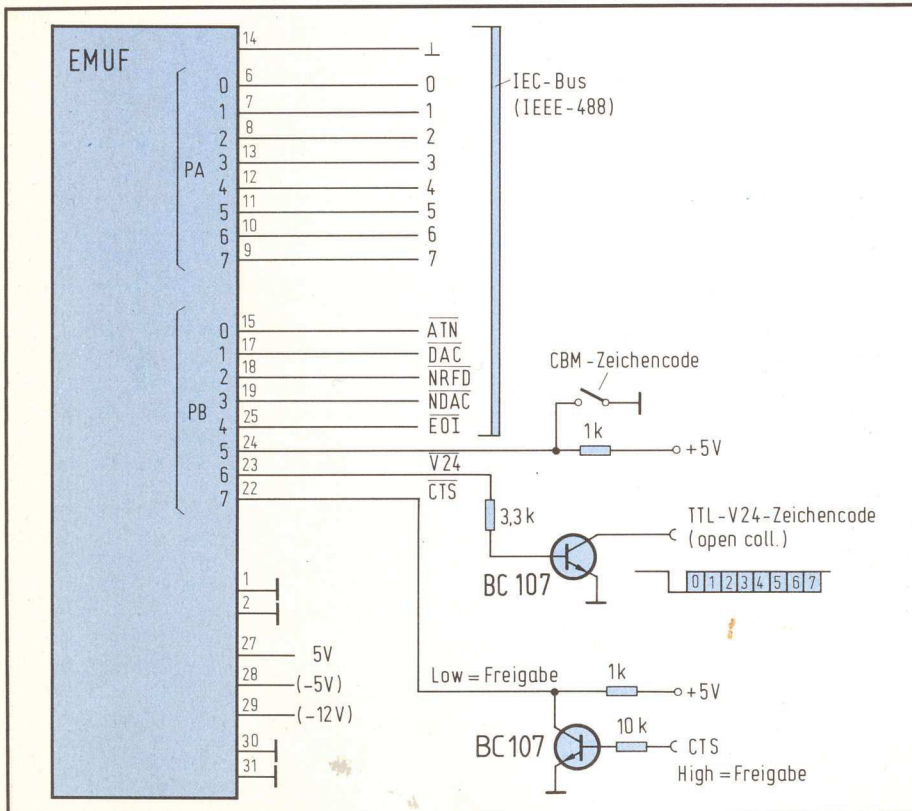


Bild 2. Die Steckerbelegung des EMUFs

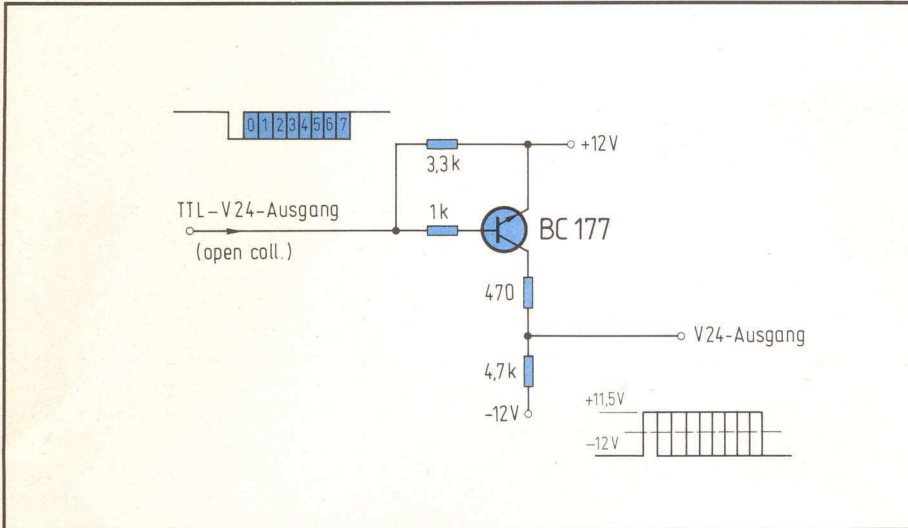


Bild 3. V24-Pegelwandler

EMUF auch mit dem Bar-Code-Leser in [1] an einem Bus betreibbar, da dieser die unveränderte erste Adresse nimmt. Über die Sekundäradresse läßt sich die Baudrate einstellen. Wird keine Sekundäradresse angegeben, so wird die zuletzt eingestellte Rate genommen. Nach dem Einschalten ist dies 1200 Baud. Die Sekundäradressen haben folgende Bedeutung:

- 0 110 Baud
- 1 300 Baud
- 2 600 Baud
- 3 1200 Baud
- 4 2400 Baud
- 5 4800 Baud
- 6 9600 Baud

Bild 4. Programmlisting des IEC/V24-Interface

```

;          EMUF   IECV24 INTERFACE 810814
;          ROLF-DIETER KLEIN
;
0050          WIDTH 80
;PORT A
; 7 .. 0  IEC DATABUS
;PORT B
; 7      6      5      4      3      2      1      0
; -CTS  -V24  PET   -EOI  -NDAC -NRDF  -DAC   -ATN
; -CTS = LOW DANN FREI
; -V24 RUHEPEGEL = LOW
; PET = 0 DANN PETUMWANDLUNG
;
0800          PA      EQU   $800
0801          PAD      EQU   $801
0802          PB      EQU   $802
0803          PBD      EQU   $803
;
0814          TIM1     EQU   $814
0815          TIM8     EQU   $815
0816          TIM64    EQU   $816
0816          TIMIN    EQU   $816
0817          TIMFLG   EQU   $817
;
0000          XTEMP1   EQU   $0
0008          ZOUT     EQU   8
0009          ZCOU     EQU   9
000A          FLAG     EQU   $A      ;IEC MERKER ATN ..
000B          ZEICH    EQU   $B      ;ZWSPEICHER
000C          COUNT    EQU   $C      ;ZWSPEICHER
000D          PADR     EQU   $D      ;FIRST TIME
000E          MDE      EQU   $E      ;MODE 0,1,2
000F          CNT      EQU   $F      ;ZEITSCHL.
;
;
; INIT ROUTINE
0FFC          ORG      $FFC
0FFC 000C     DW      $C00
;
;
0C00          ORG      $C00      ;START
0C00 A2FF     RESET:  LDX      #$FF      ;STACKPOINTER
0C02 9A       LDA      #Z00000000      ;ALL INPUT
0C03 A900     LDA      #Z01001100      ;SET UP
0C05 8D0108   STA      PAD
0C08 A94C     LDA      #Z01001100      ;SET UP
0C0A 8D0308   STA      PBD
0C0D A900     LDA      #Z00000000      ;NOT READY NOT ACCEPT
0C0F D8       CLD
0C10 78       SEI

```


| ---- SYMBOL TABLE ---- | | | | | |
|------------------------|------|--------|------|--------|------|
| BAUD | 0C27 | MDE | 000E | TALPA | 0D5E |
| CKATN | 0D2A | OKSK | 0C2C | TIM1 | 0814 |
| CNT | 000F | PA | 0800 | TIM64 | 0816 |
| CONSE | 0D1A | PAD | 0801 | TIM8 | 0815 |
| CONV | 0C51 | PADR | 080D | TIMFLG | 0817 |
| CONV1 | 0C69 | PB | 0802 | TIMIN | 0816 |
| CONV2 | 0C5E | PBD | 0803 | V241 | 0C8B |
| COUNT | 080C | PETASC | 0C47 | V24LP | 0C6B |
| DATEN | 0D97 | RESET | 0C00 | V24OUT | 0C6A |
| FINA | 0DAE | SEND | 0D05 | VL1 | 0C98 |
| FLAG | 080A | SEND1 | 0D04 | VLO | 0C79 |
| GETCHA | 0CA4 | SENE01 | 0D2B | VLOP | 0C7C |
| LISTPA | 0D61 | SENE0I | 0D2C | VSK | 0C94 |
| LOPA | 0CA9 | SK2 | 0D78 | XTEMP1 | 0900 |
| LOPB | 0CC5 | SK3 | 0D80 | ZCOU | 0809 |
| LOPMAI | 0D92 | TABBAU | 0C39 | ZEICH | 080B |
| MAIN | 0D44 | TALKOF | 0CEF | ZOUT | 0808 |
| MAIN1 | 0D49 | TALKON | 0C03 | | |


```

10 OPEN1,8,3
20 PRINT#1,"IEC/V24-INTERFACE";CHR$(10)
30 CLOSE1
40 OPEN128,9,3
50 PRINT#128,"TEXT"
60 CLOSE128
READY.
    
```

Bild 5. Programmierbeispiel für den CBM 8032

In Bild 5 ist ein Programmierbeispiel für den CBM-Rechner dargestellt. Die ersten Anweisungen in Zeile 10 und 20 definieren die Primäradresse im EMUF. Die Druckanweisung in Zeile 20 wird bereits ausgeführt, sie soll eigentlich nur mindestens ein Zeichen auf den IEC-Bus bringen. In Zeile 40 wird ein Kanal 128 eröffnet, der nun die neue Primäradresse beinhaltet. Mit der Sekundäradresse 3 wird eine Baudrate von 1200 Baud (auch Voreinstellung) programmiert. Der Kanal 128 besagt beim CBM 8032, daß bei der Print-Anweisung Zeilen mit CR (Wagenrücklauf) gefolgt von einem LF (Zeilenvorschub) ausgegeben werden. Wird Kanal 1 verwendet, so erfolgt am Ende der Zeile nur ein CR.

Literatur

[1] Rolf-Dieter Klein. EMUF bringt Strichcode zum IEC-Bus. mc 1981, Heft 3.

EMUF mit erweiterter Adressierung

Ein kleiner Schönheitsfehler des EMUF ist die Tatsache, daß nur 1 KByte EPROM adressiert werden kann. Viele Anwender möchten aber den 2716 als Programmspeicher einsetzen und die vollen 2 KByte benutzen. Dieses ist mit einer kleinen Änderung möglich. Die Änderung erfordert das Auftrennen von vier Leiterbahnen und das Einsetzen von drei Drahtbrücken und ist deshalb auf der vorhandenen Platine leicht zu verwirklichen.

Es sind folgende Verdrahtungsänderungen nötig:

1. Leitung A11 (6504, Pin 16) – \overline{RS} (6532, Pin 36) auf der Lötseite bei dem Lötspot neben Pin 1 des 7400 trennen und das zu \overline{RS} (6532, Pin 36) führende Ende mit einer Drahtbrücke mit A9 am besten an Pin 22 des 2716 verbinden.
2. Leitung A10 (6504, Pin 15) – Pin 10 (NAND im 7400) auf der Lötseite kurz vor Pin 14 (7400, +5 V) trennen, Pin

10 (7400) mit Pin 14 (7400, +5 V) verbinden und das von A10 (6504, Pin 15) kommende Ende über eine Drahtbrücke mit dem Anschluß für A10 am 2716, Pin 19 verbinden.

3. CS1 (6532, Pin 38) auf der Bestückungsseite von der vorbeiführenden +5-V-Leitung abtrennen und auf der

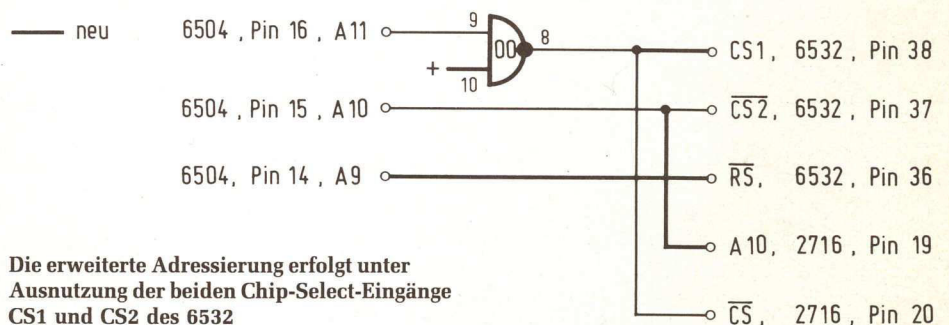
Lötseite mit einer Drahtbrücke mit Pin 8 des 7400 verbinden.

4. Unterbrechen der Leitung Pin 29 (Steckerleiste) – Pin 19 (2716), da beim 2716 am Pin 19 die Adreßleitung A10 liegt und die beim 2708 notwendige +12-V-Versorgung entfällt.

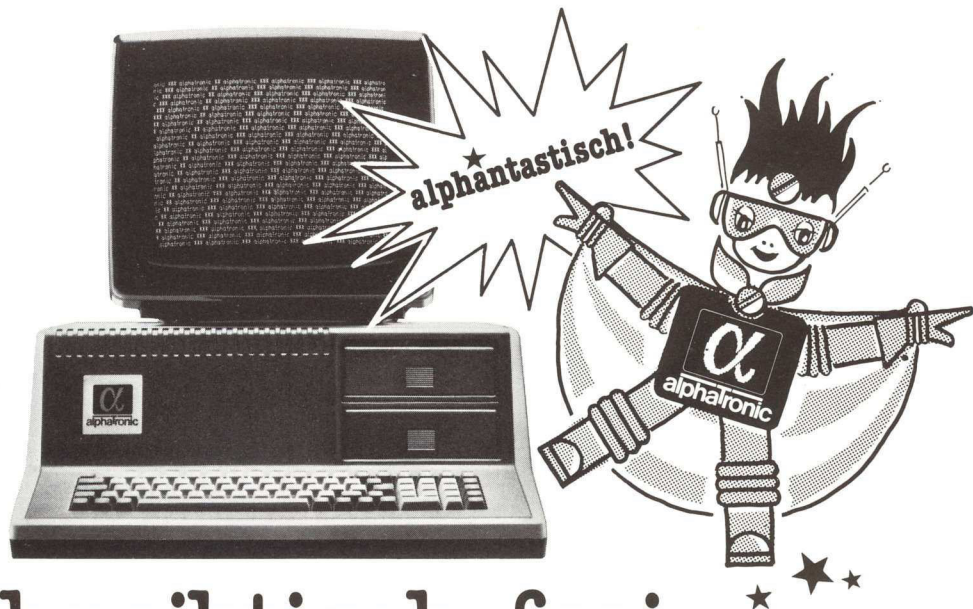
Michael Berger

Tabelle: Die neue Adressenbelegung

| A ₁₁ | A ₁₀ | A ₉ | |
|-----------------|-----------------|----------------|---|
| 0 | 0 | 0 | Zeropage und Stack gemeinsam in den 128 Byte RAM im 6532: 0000-007F Zeropage 0180-01FF Stack |
| 0 | 0 | 1 | E/A und Timer im 6532: 0200-021F |
| 0 | 1 | | frei (1K) |
| 1 | | | 2K Programm im 2716 (0800-0FFF) |



Laßt den Großen ihre Computersysteme!



Schreibtisch frei für die pfiffige Computer-Idee: alphontronic * Micro

Aktuelles aus der alphontronic Micro-Bar:

Standard-Pakete

Textverarbeitung
Auftragsabwicklung
Fakturierung
Finanzbuchhaltung
Mahnwesen
Lagerbuchhaltung
Lohn- und Gehaltsabrechnung
Kostenplanung
Personalplanung
Budgetkontrolle
Kalkulationen

Spezialanwendungen

Rechnerunterstützte
Maschinenfähigkeits-
untersuchung

Branchenpakete

Architekten
Ärzte
Dachdecker
Druckereien
Elektrohandwerk
Energieberatung
Fahrschulen
Fensterbau
Glasgroßhandel
Hausverwaltung
Heizungsplanung
Hotel
Malerei
Restaurant
Speditionen
Textileinzelhandel
(kurzfristige Erfolgsrechnung)
Versicherungsagenturen

Informationsabruf:

alphontronic Micro von TRIUMPH-ADLER, die Schreibtischlösung für Kleinbetriebe und Selbständige, die sich jeder leisten kann.

Text- und Computersystem in einem. Mit deutscher Schreibmaschinen-DIN-Tastatur, Norm-Zehnerblock, Bildschirm und Tischdrucker, Programm-Disketten, die einfach wie Schallplatten gehandhabt werden. Griffbereit an der alphontronic Micro-Bar.

Ich möchte wissen, wie pfiffig der alphontronic Micro wirklich ist. Bitte senden Sie mir weitere Informationen. Am meisten interessiert mich folgende Anwendung:

mc-emuf 1.3.82/alpha

Name

Titel/Stellung

Straße

Tel.

PLZ/Ort

TRIUMPH-ADLER Aktiengesellschaft
für Büro- und Informationstechnik
Witschelstraße 71 · D 8500 Nürnberg
Tel. (09 11) 32 05-1 · Telex 6-26 155

TA TRIUMPH-ADLER

NEU

HACKERCORNER

Angebot des Monats: Solange Vorrat reicht. 232 The Best of Creative Comp., Vol. 1 29,80 233 The Best of Creative Comp., Vol. 2 29,80 8020 Dr. Dobbs Sammelband, Vol. 1, auszug. Computerinform., 350 S. A4 DM 29,80

8056 My Computer likes me 9,80 8058 Interface Datenbuch 19,80 X1 Soundchip AY-38912 49,00 420 Schach f. CBM + PET 2000/3000 79,00 4812 Editor/Assembler CBM 3016/32 169,00 426 Textverarbeitung CBM/PET 96,00 4826 Gunfight PET/CBM 19,80

TRS-80 / Video Genie

Best.-Nr. Bezeichnung Preis/DM 5087 PACKER 149,00 5088 Z-80 Disassembler in Masch.-Spr. 99,00 5090 PRINT TO LPRINT TO PRINT 49,00 5091 Echtzeituhr für TRS-80 59,00

Geschäftsprogramme

5005 General Ledger-Hustl. 1 (C) 69,00 5006 General Ledger-Hustl. 2 (C) 89,00 5007 Checking Accounts (C) 79,00 5013 Lagerverwaltung + Inventur (C) 49,00 5014 Adressverwaltung (Cassette) 49,00 5025 Editor/Assembler 89,00 5034 Commerzielle Programme (C) 89,00 5037 Rechnungsschreibprogr. (D) 874,00 5038 Mailing List (D) 99,00 5039 Textverarbeitungspr., Text 81 (D) 99,00 5040 Inventurprogramm auf Diskette 298,00 5063 Textverarbeitung (Cassette) 49,00 5072 Advanced Statist. (C) 99,00 5073 Advanced Statist. (D) 99,00 5100 TEXED (Texteditor) (D) 198,00 5101 Adressverwaltung (Diskette) 149,00 5102 Ladenkasse (Cassette) 99,00

NEU

Spiele und Unterhaltung

5028 Snake Eggs (C) 49,00 5029 ANDROID NIM (C) 49,00 5030 LIFETWO (C) 49,00 5031 CUBES (C) 39,00 5032 42 Programme (C) 79,00 5045 TRS-80 Spiele (dt.) (C) 29,80 5048 TRS-80 Opera (C) 49,00 5049 SCRAMBLE (C) 49,00 5050 BEEWARY (C) 49,00 5051 CHALLENGE (C) 49,00 5052 Great Race (C) 49,00 5053 Owl Tree (C) 49,00 5055 Lying Chimps (C) 49,00 5062 AIR Traffic Controller (C) 24,80 5066 Spielprogramm Level I (C) 24,80 5068 Brettspiele (C) 24,80 5069 Weltraumspiele (C) 24,80 5070 Adventure Land (C) 59,00 5074 Pirate Adventure (C) 59,00 5080 Sargon Schach (D) 129,00 5081 Sargon Schach (C) 99,00

Nützliche Utilities

5041 EMU 02 (6502 Emulator) (C) 99,00 5042 JN LOCO PAC (Irelocate) (C) 49,00 5043 Super STEP (Single-step) (C) 49,00 5044 Super TLEGS (C) 49,00

Bücher für TRS-80, ZX-80, Video Genie etc.

111 Progr. m. TRS-80 und Z-80 29,80 119 Progr. i. Masch.-Spr., Z-80 49,00 155 The First Book of TRS-80 29,80 208 TRS-80 User Journal 14,80 245 Microsoft BASIC Decoded 89,00 246 BASIC Faster and Better 129,00 250 TRS-80 Beginners Programs 29,80 251 TRS-80 Sargon Chess Book 49,00 252 Z-80 Referenz-Karte 5,00 272 Z80 + 8080 Assembly Lang. Progr. 39,00 8029 Z-80 Assemblerhandbuch 29,80

TAB-Books

574 Beginner's Guide to Computer Pr. 39,00 752 Computer Programming Handbook 45,00 785 Microprocessor/Microprogramming 35,00 952 Microcomp. Progr. f. Hobbyist 39,00 1000 57 Practical Programs in BASIC 35,00 1015 Beginner's Guide to Microproc 29,80 1055 The BASIC Cookbook 24,80 1071 Complete Handbook of Robotics 29,80 1085 24 Ready to Run Progr. in BASIC 24,80 1088 Illustrated Dictionary of Microc. 35,00 1095 Programs in Basic f. Electr. Eng. 19,80 1070 Digital Interfacing 39,00 1141 How to Build your own working Robot PET 29,80 1076 Artificial Intelligence 29,80 1111 How to Design, Build + Program your own working Computer System 29,80 1099 How to Build your own work. 16 Bit Microc. 14,80 1062 The A to Z Book of Comp. Games 29,80 1053 Microprocessor Cookbook 24,80 1045 The Programmers Guide to LISP 24,80 1050 The most pop. Subrout. in BASIC 24,80 1169 The Giant Book of Comp. Projects f. 8080/6502 39,00 1187 The Fortran Cookbook 29,80 1203 Handb. of Microproc. Appl. 29,80 1205 PASCAL 35,00 1236 Fiberoptics 29,80 1271 Microcomp. Interfacing 35,00 1275 33 Chall. Comp. Games 29,80 1228 34 More Tested Ready-to-Run Pr. 35,00 311 Dragon Byte Disk Expansion Book 29,80 1341 How to Design and Build 59,00 274 The 8086 Primer 49,00 1191 Robot Intelligence with Exp. 49,00 1195 67 Ready to Run Progr. i. Basic 29,80 1276 Computer Graphics with 29 Progr. 39,00 1200 How to build your own working 49,00 1209 The MC 6809 Cookbook 29,80

ELCOMP. Fachzeitschrift f. Microcomputer

Einzelpreis 5,00 DM Jahresbezugspreis 69,00 DM Zurückliegende Hefte: Sept. 1978, Sept. 1979 (außer Nr. 2 und 4 1979) 37,00 DM Jahrgang 1981 (außer Nr. 2) 42,00 DM

VC-20

★★★ NEU ★ NEU ★ NEU ★★★

Best.-Nr. 478 VC-20 Games-Paket 3 aufregende Spiele (VIC-Trap, Bounce out, Seawolf). In Farbgrafik mit Ton (Grundversion). Mit engl. Beschr. 99,- DM Best.-Nr. 493 Haushaltsfinanzen mit VC-20 Dieses Paket besteht aus vier Progr. (Grundversion, Engl. Beschr. 179,- DM Best.-Nr. 4827 VC-Mona Ein einfacher Masch.-Monitor f. Grundversion. Durchforsten Sie ROM u. RAM. Zellen ansehen, ändern. 19,80 DM Best.-Nr. 4828 Spielesammlung für VC-20 Lustige u. unterhaltsame Spiele 49,- DM Best.-Nr. 4840 Logic Games Code Breaker u. Code Maker (C) 79,- DM Best.-Nr. 4841 Recreational / Educational I Hangman und Hangmath 69,- DM Best.-Nr. 4842 Monster Maze + Hurdler Sie werden begeistert sein. 69,- DM Best.-Nr. 4843 16k Speichererweiterung 16k-RAM od. EPROM 2716. Leiterplatte m. ausf. Bauanl. (ohne Bauteile) 149,- DM Best.-Nr. 4844 Universal Experimentierpl. Zum Aufbau eigener I/O u. Erw. 129,- DM Best.-Nr. 4845 Joystick für VC-20 Bauelemente m. Grundsoftware 149,- DM Best.-Nr. 4846 Schalterinterface f. VC-20 Schalten Sie Netzverbinder wie Radio, TV, etc. m. Ihrem Computer per Progr. 199,- DM 4847 Stecker für USER PORT 19,80 DM 4848 Stecker f. Erweiterungsport 19,80 DM 4860 3K RAM Expander f. Progr. u. ROM-Entwickler (fertig) 269,- DM 4861 RS232 Kommunikationsinterface (ohne Terminalprogr. (fertig)) 299,- DM 4862 Terminalprogr. f. Kommunikationsinterface 129,- DM 4863 8K RAM-/ROM-Platine (ohne Teile) 149,- DM 4864 BASIC Programmiers UTILITY ROM 199,- DM 4865 Allen Blitz VC-20 99,- DM 4866 Amokläufer VC-20 99,- DM 6210 Endlospapier für Ihren VC-20 Drucker, Kiste m. 1.000 Blatt 79,- DM 6211 Adressaufkleber, selbstklebend, per Karton f. VC-20 Drucker 199,- DM 141 Programmier-HB f. VC-20 29,80 DM

SINCLAIR ZX 81

Achtung - Sinclair ZX 81 Besitzer und solche, die es werden wollen!

Programmier-Handbuch für ZX 81, v. E. Flügel Endlich ein deutsches Programmier-Handbuch für den Sinclair ZX-81. Viele Tricks, Tips, Hinweise, Programmieren in Maschinenspr. mit ZX 81, Hardware-Erweiterung, lustige Spielprogramme zum eintippen. Best.-Nr. 140 29,80 DM

15 Programme für den ZX81 auf Cassette, 1

Best.-Nr. 2397 49,- DM

15 Programme für den ZX81 auf Cassette, 2

Best.-Nr. 2398 49,- DM

Schachprogr. incl. Schach-Uhrprogr. f. ZX 81

Best.-Nr. 2399 149,- DM

Adapterplatine f. ext. Experimente

Best.-Nr. 2400 39,- DM

Weitere interessante Bücher für den ZX 81 Besitzer

Z80 Assembler Handbuch Erklärung der Maschinenbefehle Best.-Nr. 8029 x 29,80 DM

Best.-Nr. 252 Z80 Referenzkarte 5,- DM

Programmieren in Maschinenspr. mit Z80 Best.-Nr. 119 49,- DM

BASIC-Handbuch Einführung in BASIC Best.-Nr. 113 19,80 DM

Elektronik Fachbücher

1 Transistor-Berechn. u. Bauanl. HB 29,80 2 TBB, Band 2 19,80 3 Elektr. i. Auto m. HB f. Polizei-Radar 9,80 4 IC-Handbuch (TTL, CMOS, Linear) 19,80 5 IC-Datenbuch 9,80 6 IC-Schaltungssammlung 19,80 7 Elektronikschnellkurs zum Basteln 19,80 8 IC-Bauanleitungs-Handbuch 19,80 9 Feldeffekttransistoren 9,80 10 Elektronik und Radio, IV 19,80 11 IC-NF-Verstärker 9,80 12 Beispiele integrierter Schaltungen 19,80 13 Hobby-Elektronik-Handbuch 9,80 14 IC-Vergleichsliste, TTL, CMOS (neu) 29,80 15 Optoelektronik-Handbuch 19,80 16 CMOS, Teil 1 19,80 17 CMOS, Teil 2 19,80 18 CMOS, Teil 3 19,80 19 IC-Experimentier-Handbuch 19,80 20 Operationsverstärker 19,80 21 Digital-Grundkurs 19,80 22 Elektronik-Grundkurs 9,80

HOFACKER

Ing. W. Hofacker GmbH, Tegernseerstr. 18, 8150 Holzkirchen, Tel. (08024) 73 31

Lieferung durch den Fach- und Buchhandel od. per Nachnahme od. Vorkasse, Postcheck-Kto. Mchn 15 994-807 od. Eurocheck, Eurocard, Preise inkl. MwSt., zuzgl. Porto u. NN-Gebühr. Unverbindliche Preisempfehlung. Angebot freibleibend. Zwischenverkauf vorbehalten.

ATARI 400 / 800

7001 16k BASIC Texteditor (C) 69,00 7002 dto. (D) 89,00 7003 3-D Computer-Grafik (C) 139,00 7004 dto. (D) 159,00 7005 Roter Baron, Luftkampf (C) 79,00 7007 Submarine Minefield (C) 49,00 7008 Down the Trench (8, 16, 24k) (C) 79,00 7009 Panzerkrieg-Battling (8k) (C) 49,00 7010 WUMPLUS Adventure 16k (C) 69,00 7011 WUMPLUS Adventure 24k (C) 79,00 7012 Schnuppertape (8/16k) (C) 49,00 7015 Direct Sound Output Cable 89,00 7019 Einfache Spiele in BASIC (C) 19,80 7020 Rechnungen schreiben (C) 99,00 7021 Adressverw. f. ATARI 800 (C) 99,00 7022 ATMONA-1 (Ma. Monitor) (C) 49,00 7023 Progr. i. Maschinensprache (C) 49,00 7024 Trivia Unlimited 24k (C) 49,00 7025 Trivia Unlimited 24k (D) 69,00 7026 Outdoor Games (C) 49,00 7028 Haunted House (C) 49,00 7029 Nr. 7026 + 7028 zusammen (D) 79,00 7037 Hail to the Chief 40k (D) 99,00 7038 Hail to the Chief 32k (C) 99,00 7209 First Book of ATARI 79,00 7040 Stecker (Game Connectors) (W) 19,80 7041 EPROM-Programmiergerät 2716/2732 Platine + Anleitung 349,00 7049 Supertracer (C) 149,00 7098 Editor/Assembler 32 od. 48k 199,00 7099 MACRO Assembler 48k 299,00

Adventure Spiel: Die aufregenden Abenteuer-spiele von Crystalware sind jetzt auch bei uns erhältlich. Für ATARI 800 auf Diskette mit ausf. engl. Anleitung. Für ATARI 800 m. ausf. engl. Anleitung. 7200 Quest for Power (D) 199,00 7201 Oregon Trail (D) 189,00 7202 Forgotten Island (D) 198,00 7203 Bermuda Triangle (D) 198,00 7204 Galactic Expedition (D) 198,00 7205 Waterloo II (D) 249,00 7206 The Crypt (D) 199,00 7207 Gunfight (C) 79,00



Druckerinterface f. Centronics kompatible Schnittstelle (EPSON, ITOH etc.) Platine mit Teilen u. komfortabler Software (Bildschirmdruck, einstellbare Zeilenlänge) Best.-Nr. 7208 179,00 7209 Morstrainer f. ATARI 400/800 149,00

APPLE II

6118 Schach - SARGON II (D) 119,00 6119 Super FORTH (D) 169,00 6151 Applesoft Compiler 699,00 6126 Dateiverwaltung (D) 199,00 6127 Adressverwaltung (D) 199,00 6128 Super Invaders (D) 99,00 6130 Utilities I (D) 99,00 6131 Utilities II (D) 99,00 6132 Statistik (D) 99,00 6133 Inventory (D) 69,00 6134 Invoicing (D) 79,00 6135 Dictionary (D) 49,00 6136 Game Package (D) 69,00 6140 Artikelverwaltung (D) 199,00 6141 Lagerbestand (D) 149,00 6142 SUPER APPLE™ BASIC (D) 199,00 6150 Adressverw. i. PASCAL (D) 199,00

Erweiterungsplatinen

für APPLE II und 6502 allgemein 604 Universal Experimentierpl. 59,00 605 Ein-/Ausgabe Experimentierpl. 129,00 606 Bus Expansion ELCOMP-1 129,00 607 EPROM Burner 2716 149,00 608 Musik Platine f. 8912 89,00 609 EPROM/RAM (4 x 2716 od. 4802) 59,00 610 A/D-Wandler 12 Bit (ADC 1210) 149,00 611 6502 Rechnerkopplung 249,00 612 32k RAM-Karte Dynamisch 169,00 615 16k RAM/EPROM Karte 149,00 625 S-44 Universal Experimentierpl. 189,00

HAYDEN Books

253 Computer controlled Robot 35,00 254 The S-100 Handbook 49,00 255 BASIC BASIC 39,00 256 Stimulating Simulations 19,80 257 BASIC Comp. Progr. in Science and Engineering 39,00 258 APL-An Introduction 39,00 259 Creative Progr. for Fun and Profit 39,00 260 BASIC Comp. Progr. f. Business, I 39,00 261 BASIC Comp. Progr. f. Business, 2 39,00 262 Homecomputer can make you rich 19,80 263 Sixty Challeng. Problems 19,80 264 The complete 1802 Cookbook 19,80 265 Musical Applications for Micros 79,00 266 Advanced BASIC Appl. 39,00 267 How to profit from your Microc. 39,00 268 Pascal with Style 39,00 269 Cobol with Style 39,00 270 BASIC with Style 39,00 271 BASIC FORTRAN 45,00 272 Z80 and 8080 Assembly Language Programming 39,00 273 Beat the ODDS: Microcomputer Simulations of Casino Games 39,00

NEUHEITEN

32 ATARI BASIC Handb. (400 S.) 29,80 35 Der freundliche Computer 29,80 114 Der Microcomputer i. Kleinbetr. 39,80 116 16 Bit Microcomputer (400 S.) 29,80 120 Anwenderpr. TRS-80/Video Genie 29,80 122 BASIC für Fortgeschrittene 39,00 130 Programme für CBM 19,80 132 CP/M Handbuch 19,80 137 FORTH Handbuch + Einführung 39,00 139 BASIC für blutige Laien 19,80 140 Programmier-HB für ZX81 29,80 141 Programmier für VC-20 29,80

ELCOMP Books in English

150 Care a. Feeding of the Comm. PET 19,80 151 8K Microsoft Basic Ref. Manual 19,80 152 Expansion Handb. f. 6502 u. 6800 19,80 153 Microcomputer Appl. Notes (Intell) 29,80 154 Complex Sound Gen. w. Microc. 19,80 155 The First Book of 80 US (TRS-80) 29,80 156 Small Business Programs 29,80 157 The First Book of Ohio Scientific 19,80 158 The Second Book of OHIO 19,80 159 The Third Book of OHIO 19,80 160 The Fourth Book of OHIO 29,80 161 The Fifth Book of OHIO 19,80 162 ATARI Games in BASIC 19,80 163 Tpe Periph. Handb. 29,80 164 ATARI Progr. Learning by Using 19,80

BASIC Bücher

113 BASIC Handbuch für Anfänger 19,80 121 Microsoft BASIC HB 29,80 122 BASIC für Fortgeschrittene 39,00 31 57 Praktische BASIC Programme 39,00 8057 Computer Games in BASIC 39,00 160 The Fourth Book of OHIO 29,80 255 BASIC/BASIC 39,00 256 Stimulating Simulations 19,80 257 BASIC Computer Programs in Science and Engineering 39,00 260 BASIC Computer Programs 39,00 156 Small Business Programs 29,80 266 Advanced BASIC Applications 39,00 151 Microsoft BASIC 19,80 270 BASIC with Style 39,00

University Software

Application Programs in Microsoft BASIC. 5 Bände mit 105 sehr guten Programmen in Spiralbindung zum Gesamtpreis von 543,00 8600 Small Business 199,00 8601 Education u. Scientific 139,00 8602 Fun u. Games, Volume 1 59,00 8603 Fun u. Games, Volume 2 59,00 8604 Home u. Economics 99,00

Riesensammelmagazin

8050 BASIC Software, Volume I 99,00 8051 BASIC Software, Volume II 99,00 8052 BASIC Software, Volume III 149,00 8053 BASIC Software, Volume IV 39,00 8054 BASIC Software, Volume V 39,00 8048 BASIC Software, Volume VI 199,00 8049 BASIC Software, Volume VII 159,00 8021 BASIC Software, Volume I-V 425,00

6502 Bücher

8042 6500 Software Manual 19,80 8043 6500 Hardware Manual 19,80 109 6502 Microcomputer Progr. 29,80 110 Programmierhandbuch PET 29,80 118 Programmieren in Maschinensprache mit dem 6502, für Apple, VC-20, PET, AIM, ATARI, Ohio (240 Seiten, neue überarbeitete Auflage) 49,00 150 Care and Feeding of the PET 19,80 152 Expansion Handbuch 6502 19,80 34 TINY BASIC Handbuch 19,80 1169 The Giant Book of Comp. Projects 39,00 157 The First Book of OHIO 19,80 158 The Second Book of OHIO 19,80 160 The Fourth Book of OHIO 29,80

Zubehör

600 1 Diskettenhülle f. 2 Disketten 2,30 601 Redysoft-Plastikordner, DIN A4 19,80 602 ELCOMP-Plastikordner, DIN A4 19,80 603 ELCOMP-Sammellordner 14,80 604 Ordner m. 20 Diskettenhüllen 69,00 605 ELCOMP-Plastikordner, DIN A5 9,80 10 MAXELL Disketten 5 1/4" soft. 99,00 Leercassetten - C 10- 8089 1 Cassette 3,50 8100 10 Cassetten 29,80 8096 100 Cassetten 249,00

SONDERANGEBOTE

Für den MICROCOMPUTER-Freund Sonderangebote - solange der Vorrat reicht 350 10 Creative Computing Hefte gem. 29,80 351 20 Creative Computing Hefte gem. 42,00 352 7 Byte Magazine Hefte gemischt 22,50 353 AIM-Manual, 6502 Hardware Manual, Softwareman., 2 Programmierkarten, Schaltplan, zus. 79,00 354 10 Dr. Dobbs Hefte gemischt 49,00 355 4 6502 User Notes Hefte 29,00 356 8084 Microcomputer Handbuch 5,00 Katalog gegen 2,- DM Vorkasse anfordern!