

AS EMUF- SONDERHEFT.

139 05,
m

ATINEN-MIKROCOMPUTER FÜR
RSELLE FESTPROGRAMM-ANWENDU

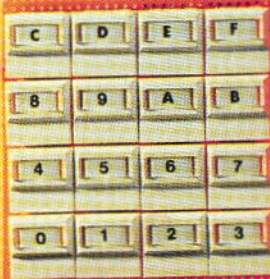
**Beweisen
Sie
dem „EMUF“
Ihr Geschick
und
Ihr logisches
Verständnis**

**Der
Schnelle
wird vom
„EMUF“
belohnt!**

Zufalls-Dualzahlen
8 4 2 1

➔ Richtig!

➔ Start ➔

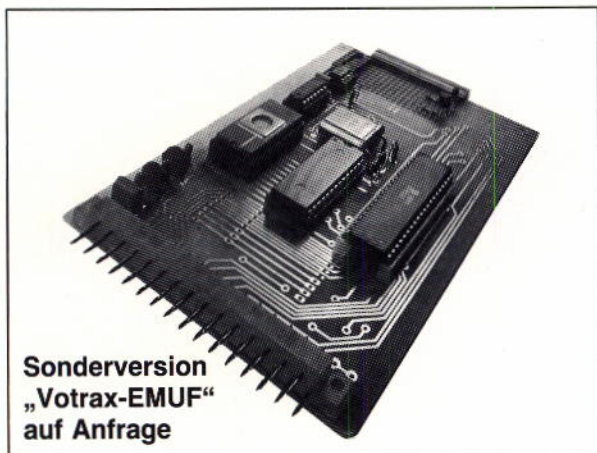


Hex-Tastenfeld

3., erweitz

mc-EMUF

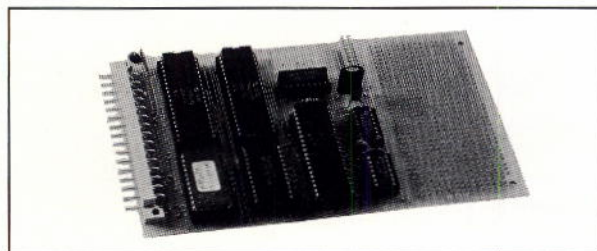
Europakartencomputer



Sonderversion
„Votrax-EMUF“
auf Anfrage

mc-Z80-EMUF

Europakartencomputer



ELZET 80

Europakartencomputer



**Wir machen Computer!
Aus Europakarten!
In Deutschland!**

– Der kleine Prozessor –

Der kleine Prozessor für Interfaces, Adapter, Ablaufsteuerungen und die vielen kleinen Probleme, die Sie bisher mit festverdrahteter Logik lösten. Vorgestellt von der „mc“ in Heft 2/1981. Flexibel, leicht zu programmieren und vor allen Dingen:

BILLIG

DM 89.–

(Bausatz mit durchkontaktierter Epoxy-Platine, glanzverzinnt, mit Bestückungsaufdruck, inkl. aller Bauteile, wie Quarz, ICs, Rs, Cs, IC-Fassungen, Stiftleisten, aussch. EPROM)

Geprüfte Fertigungskarte des EMUF, bestückt wie oben:

DM 119.–

Zwei EMUF-Applikationen im 2716-EPROM – DM 24.80 pro EPROM:

Türklingel + Alarmanlage (K1)/Whisky-EMUF (K2)*
Interface f. Typenrad-Schreibm. (K1)/V24-Schnittstellentester (K2)
Funkfernseh-Empfänger (K1)/DCF-77-Decoder (K2)
Mini-Datenlogger (K1)/LCD-EMUF (K2)
Interf. Barcode-IEC-Bus (K1)/Interf. IEC-Bus-V24 (K2)
Standardschnittst. f. Schreibm.-Drucker: Seriell (K1)/Parallel (K2)

* K1: Pin 29 der 31pol. Steckleiste auf Masse; K2: Pin 29 auf +5 V. Alle Applikationen stammen aus diesem Sonderheft.

Andere Applikationen auf Anfrage

– EMUF für Z80-Freunde –

Aus mc Heft 4/1983: Ein Grundkonzept für einen beliebigen Prozessor. Der 8080-softwarekompatible Prozessor Z80 hat viele Freunde durch das CP/M-Floppy-Betriebssystem und die darunter laufenden Programme.

Der Z80-EMUF ist ausbaufähig bis zu 32 E/A-Leitungen (2 x PIO), bis zu 8 kByte EPROM und 2 kByte RAM. Wie der 6504-EMUF ist er geeignet für kleine Interfaces und Ablaufsteuerungen. Das Wrapfeld läßt viel Platz für Erweiterungen.

Bausatz 4 MHz mit durchkontaktierter Epoxy-Platine, inkl. aller Bauteile, wie Quarz, Rs, Cs, IC-Fassungen, Steckerleisten, der Z80A CPU, einer Z80A PIO, 2 kByte RAM, jedoch ohne EProm: nur DM 99.–

Z80 EMUF als geprüfte Fertigungskarte:

DM 129.–

Softwareunterstützt durch ELZET 80 – PROCESS BASIC!

**Fordern Sie unsere Liste über
andere Z80-Einplatinencomputer an!**

– Das modulare System –

Seit fünf Jahren im Einsatz in Industrie und Forschung: ein busorientiertes Europakartensystem (ECB-kompatibel) aus über 50 Baugruppen.

Neben verschiedenen CPU's, FDC's, Videodisplays usw. auch Speicherkarten von 16K bis 256K und die Großauswahl an E/A-Karten: Arithmetikprozessor, Relais, 24-V=-Teiler, Optokoppler-E/A, RS232, IEC und Centronics-Schnittstellen in vielen Variationen, Steuerungsüberwachung, A/D- und D/A-Wandler, Zähler/Zeitgeber, Mini-Digitalrecorder, Rasterfeld-Experimentierkarten mit nur Busanschluß oder E/A-Ports und Zähler, versch. Busplatinen...

Ein umfangreicher Katalog liegt für Sie abrufbereit!

ELEKTRONIKLADEN

Giesler & Danne GmbH & Co. KG

Wilhelm-Mellies-Straße 88, D-4930 Detmold 18,
Telefon (0 52 32) 81 31, Telex 9 31 473 laden

ELZET 80 gibt es auch bei unseren Händlern:

ESmed, Berlin
0 30/3 63 61 73

GMS, Frankfurt
06 11/78 87 52

SCHWEIZ
Bernhard Elektronik
Aarauer Straße 20
CH-5734 REINACH AG
0 64/71 69 44

Allmos, Stuttgart
07 11/7 54 18 55

PTL, München
0 89/16 99 77

Alle angegebenen Preise beinhalten die gesetzliche Mehrwertsteuer.

Vorwort

Ein Produkt setzt sich durch, könnte die Überschrift auch lauten. Was ist das Erfolgsgeheimnis des „Einplatinen-Computers für universelle Festprogramm-Anwendung“, wie der EMUF ausgeschrieben heißt?

Drei Dinge sind es, die dafür sorgten, daß der EMUF heute mehrtausendfach verbreitet ist. Und daß aus einem Kunstwort, das 1981 keiner kannte, ein Markenname geworden ist.

Erstens: Der Preis stimmt. Das heißt, daß der EMUF für viele Applikationen in Frage kommt, bei denen sich vorher der Einsatz eines Einplatinen-Computers nicht lohnte. Im Handel kostet der EMUF nämlich weniger als 100 DM.

Zweitens: Für die Entwicklung von Programmen für den EMUF braucht man kein teures Entwicklungssystem (wie etwa für manche Einchip-Computer). Ein gewöhnlicher Tischcomputer mit Möglichkeit zum EPROM-Programmieren genügt, sofern er über einen Prozessor der 65XX-Familie verfügt – und solche Tischcomputer sind sehr verbreitet.

Drittens: Den EMUF kann man auch einsetzen, ohne selbst etwas von Software-Entwicklung zu verstehen. Denn für zahlreiche Applikationen sind fertig-programmierte EPROMs erhältlich – und das wieder sehr preiswert, denn in einem 2-KByte-EPROM können gewöhnlich zwei EMUF-Programme untergebracht werden. Da der EMUF ein erprobtes, vieltausendfach bewährtes Gerät ist, sind Nachbauprobleme praktisch ausgeschlossen, auch wenn man nicht über einen 10 000-DM-Logikanalysator, sondern nur über ein 40-DM-Vielfachmeßgerät verfügt. Und, wie dieses Heft zeigt, liegt auch eine sehr ausführliche Dokumentation mit vielen Beispielprogrammen vor, die eine wesentliche Hilfe auch bei der Realisation eigener, individueller Anwendungen darstellt.

Das vorliegende EMUF-Sonderheft erscheint nun in der dritten Auflage, von einigen kleinen „Ungereimtheiten“ bereinigt und mit neuen EMUF-Anwendungen sowie einem Grundlagenbeitrag über die Verwirklichung eigener Applikationen ergänzt.

*Ihre
Redaktion*

Mikro-computer ohne Ballast

Ein Mikrocomputer-Anleitungsbuch für Anfänger mit Assemblerprogrammen für die CPU 6502. Von Martinus Bernardus Immerzeel. 232 Seiten, 125 Abbildungen, 46 Tabellen. Lwstr.-geb. 44 DM. Franzis-Verlag, Postfach 37 01 20, 8000 München 37. ISBN 3-7723-6981-2

Oft genug liest man in Buchbesprechungen, daß ein bestimmtes Werk eine Lücke schließe. Hier trifft dies mit Sicherheit zu: Eine wirklich einführnde Literatur speziell über die Assemblerprogrammierung, verständlich, komplett und vor allem deutschsprachig, mußte man bisher mit der Lupe suchen. Das vorliegende Buch ist kein reines 6502-Buch, es geht kurz und bündig auch auf andere verbreitete Mikroprozessor-Typen ein. Der Schwerpunkt ist aber eindeutig die 6502-Assembler-Programmierung, da diese CPU in sehr vielen, vor allem preiswerten, Mikrocomputer-Systemen zu finden und auch mit neueren Typen wie 6509 und 6511 software-kompatibel ist. Ein paar Stichpunkte aus dem Inhalt: Digitales, binäres, hexadezimalen Rechnen. BCD-System. Logische Funktionen. Das Programm. Der Mikrocomputer. Befehle. Adressierungsarten. Einfache Programme. Die CPU 6800. Die CPU 8080A. Programmiersprachen. Der Leser sollte keine langen Kochrezept-Programme erwarten, dafür gibt es genügend andere Literatur. Hier steht das Selbst-Verstehen, das Selbst-Programmieren im Vordergrund. Daß man dann auch fremde Programme versteht, ist ein unvermeidlicher Nebeneffekt.

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. Löhr, 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.



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 ausgeschenkt. 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 42

Vorwort	3
mc-bücher	4
mc-grundlagen	
Glossarium	6
Von der Idee zum Programm	8
So entsteht eine EMUF-Applikation	17
EMUF-Programmiertips	18
mc-hard	
Mädchen für alles	10
Über die EMUF-Chips	14
AIM-65 emuliert EMUF	16
AIM schießt EMUF	23
Programmierte EPROMs für den EMUF	31
Wenn der EMUF streikt	35
Ein Netzteil für den EMUF	48
EMUF mit erweiterter Adressierung	62
EMUF-AIM-Adapterkabel	65
Kompatible EMUF-Erweiterung	73
Erweiterter EMUF	80
mc-applikation	
Standard-Schnittstellen für Schreibmaschinen-Drucker V24 / Centr.	20
EMUF bringt Strichcode zum IEC-Bus	IEC 24
Türklingel und Alarmanlage	28
Mini-Datenlogger	IEC 32
EMUF morst Rufzeichen	36
EMUF steuert Relaisfunkstelle	38
Der Whisky-EMUF	40
Interface für Typenradschreibmaschine	± V24 42
Funkfernsehempfänger	49
V.24-Schnittstellentester	52
EMUF als DCF-77-Decoder	56
V.24-Interface	IEC 58
Der Eier-EMUF	64
Selbstlernende Haus-Heizungsregelung	66
EMUF lernt sprechen	70
TRS-80 steuert Typenrad-EMUF	74
IEC-Centronics-Interface	IEC 77
EMUF als Schaltuhr, Thermometer, Stoppuhr und Frequenzmesser	81
Eine ungewöhnliche Anwendung	86
EMUF im Senegal	89
mc-soft	
D/A-Wandlung per Software	30
AIM steuert Typenrad-EMUF	51
Formatierte Assemblerlistings mit dem AIM-65	57
Interrupts mit dem 6532-Timer	88

Impressum: 3., überarbeitete und erweiterte Auflage 1983, Franzis-Verlag GmbH, Karlstraße 37-41, 8000 München 2. 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: Franzis-Druck GmbH, Karlstraße 35, 8000 München 2. Printed in Germany. Imprimé en Allemagne. ZV-Art.-Nr. 86041. F/ZV/783/800/8'.

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 ultravioletter 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 Quellencode 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 „hindeuten“, 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 geschaltet wird oder z. B. an ihm ein offener Schalter nach Masse (0 V) liegt.

Quellentext

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

Von der Idee zum Programm

Wie entsteht eine typische Applikation für einen Einplatinen-Computer? An einem einfachen Beispiel, einer Steuerung für eine Relaisfunkstelle, wollen wir uns das einmal ansehen – inklusive aller typischen Schwierigkeiten, die dabei auftreten.

Die Aufgabenstellung lautet: Eine Relaisfunkstelle ist mit einer möglichst preiswerten und einfach zu bauenden Steuerung zu versehen, die die Aufgabe hat, auf ein Empfangs-Signal zu warten, den Sender bei Bedarf einzuschalten, automatisch das postalisch geforderte Relaisfunkstellen-Rufzeichen im Morsecode auszustrahlen und auf Abruf (nach Empfang eines Tonrufes) einen vorher fest programmierten Text, z. B. mit technischen Daten der Funkstelle, ebenfalls im Morsecode zu senden. Wird kein Signal mehr empfangen, so muß der Sender mit einer Verzögerung (Haltezeit) von etwa drei Sekunden wieder ausge-

schaltet werden. Bild 1 zeigt das Blockschaltbild des Systems.

Welche Funktionen sind per Software realisierbar?

Es gibt in vielen Anwendungen die Möglichkeit, bestimmte Teilaufgaben entweder per Software vom Mikrocomputer oder per Hardware von speziellen Bauelementen übernehmen zu lassen. Bei unserer Relaisfunkstelle ist es naheliegend, rein digitale Schaltfunktionen (Sender ein/aus) vom Mikrocomputer ausführen zu lassen. Es müssen aber auch Töne erzeugt (Morsezeichen-Aus-

sendung) und erkannt werden (1750-Hz-Tonruf zum Einschalten des Senders und zum Abruf des Festtextes). Diese beiden Aufgaben könnten mit Hardware-Oszillator (z. B. NE555) und Phase-Locked-Loop-Tondecoder (z. B. NE567) ausgeführt werden, wenn dem Programmierer dazu keine Software-Lösung einfällt oder der Prozessor dafür zu langsam wäre.

Eine Alternative ist die Verwendung von Software zur Tonerzeugung (kein Problem mit Programmschleifen) und zur Tondecodierung (hier gelöst mit dem Verfahren der Autokorrelation, vgl. [1]). Daß das entwickelte Programm auf Anhieb das tut, was man sich ursprünglich vorgestellt hat, ist sehr unwahrscheinlich und bei längeren Programmen praktisch auszuschließen. Typische Fehlermöglichkeiten sind:

- Fehler bei der Erstellung des Ablaufplans. Es kann sein, daß es sich herausstellt, daß es in der Praxis eben doch nicht so geht, wie man es sich an Hand des Flußdiagramms vorgestellt hat, z. B. wegen Zeitproblemen oder logischen Irrtümern.
- Fehler bei der Erstellung des Assemblerprogrammes aus dem Flußdiagramm. Möglicherweise hat man das Flußdiagramm doch zu grob gezeichnet und so die Übersicht verloren. Eine kleine Hilfe: Zeichnen Sie nachträglich verwendete Assembler-Symbole (Labels) an die entsprechenden Stellen des Flußdiagramms ein.
- Syntaktische Fehler. Diese sind am leichtesten auszuräumen, weil sich der Assembler während der Übersetzung des Quellencodes in den Objektcode automatisch darüber beschwert.

Der Hardware wird bei vielen Lösungen der Vorzug gegeben, soweit es sich nur um Einzelstücke handelt, da die Kosten der zusätzlich benötigten Bauelemente geringer sind als die zeitintensive Erarbeitung von Software-Know-How über Spezialaufgaben durch den Entwickler. Bei größeren Stückzahlen treten dagegen

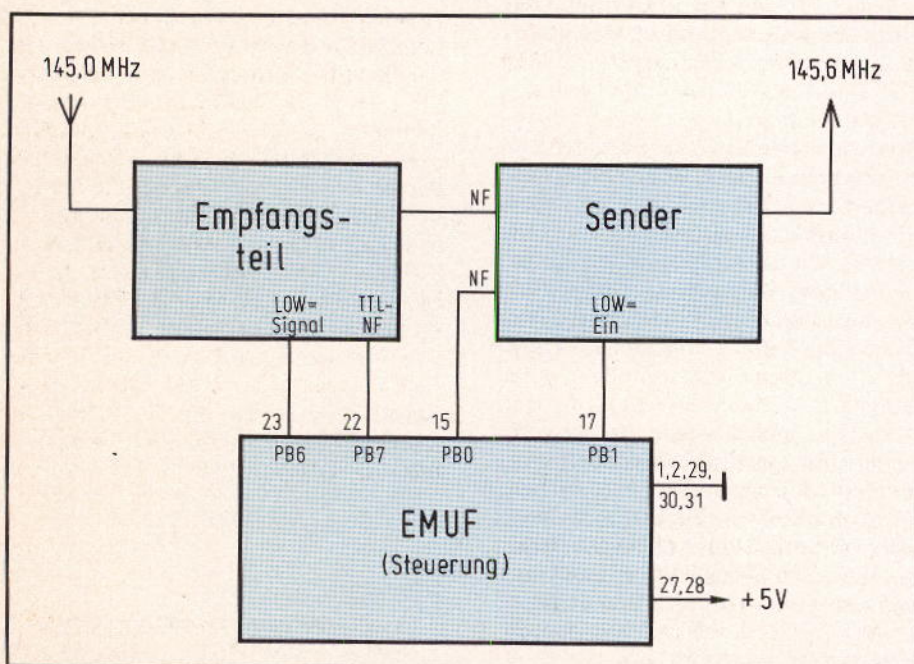


Bild 1. Blockschaltbild einer mit dem Einplatinen-Computer EMUF realisierten Relaisfunkstelle. Der EMUF enthält ein Rauschsperr-Signal vom Empfänger, das ihm sagt, ob ein Signal empfangen wird, sowie die Niederfrequenz, um einen 1750-Hz-Tonruf decodieren zu können. An den Sender liefert er ein Ein-/Aus-Signal sowie die Modulations-Nf der erzeugten Morsezeichen

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.

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 Minimalconfiguration 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.

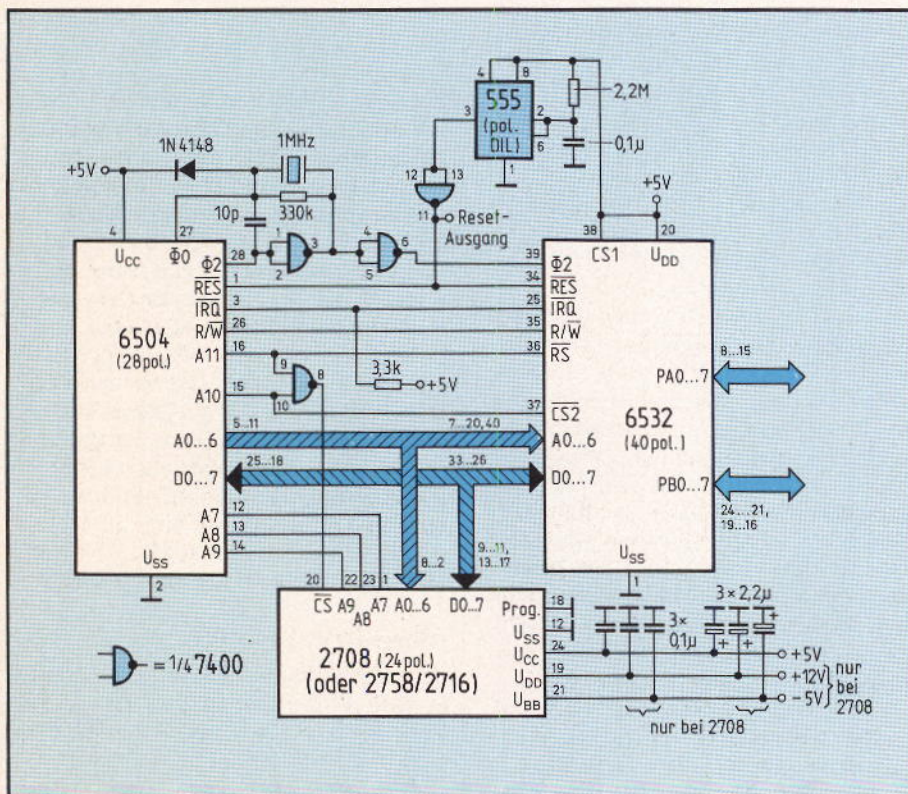


Bild 1. Gesamtschaltung des 6504-Computers mit 1 KByte ROM, 128 Byte RAM, einem programmierbaren Interrupt-Timer und 16 I/O-Leitungen

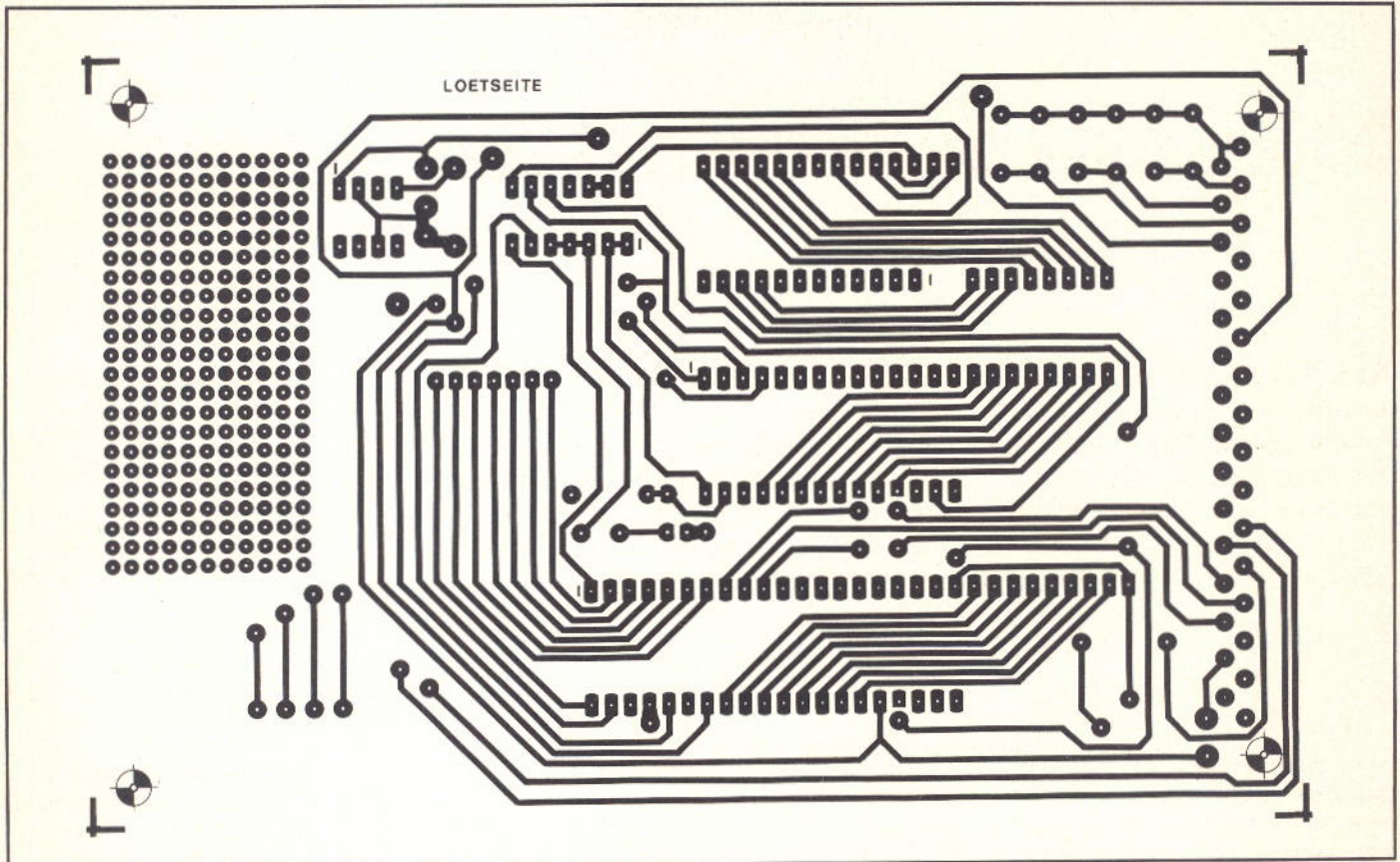


Bild 2. Lötseite der Platine. Sie enthält ein Lochraster-Feld, das vom Anwender für besondere Aufgaben frei verdrahtet werden kann, z. B. für die Nachrüstung eines D/A-Wandlers

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 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-

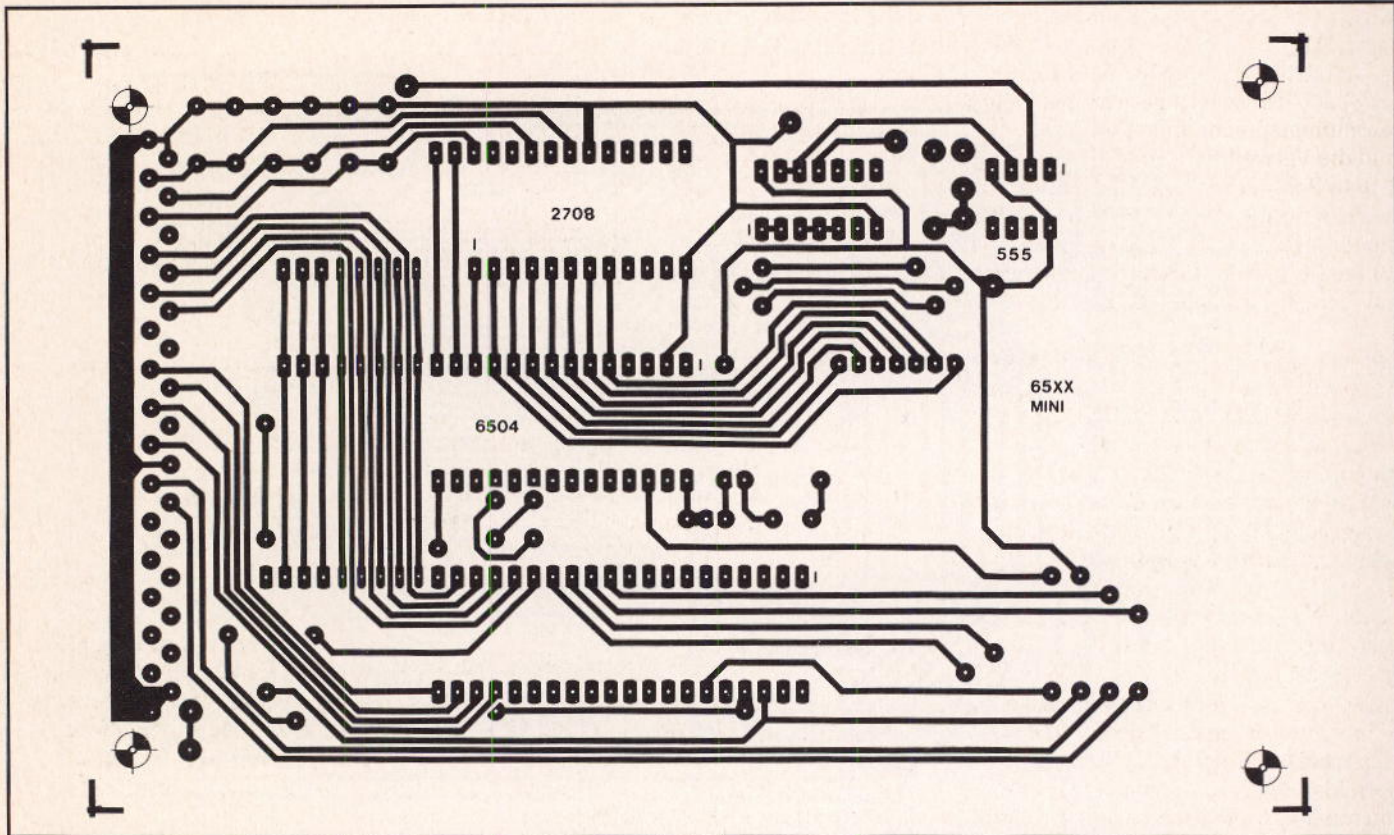


Bild 3. Bestückungsseitige Leiterbahnen der (doppelseitigen, durchkontaktierten) Platine. Die 31polige Steckerleiste ist später auf diese Seite zu löten

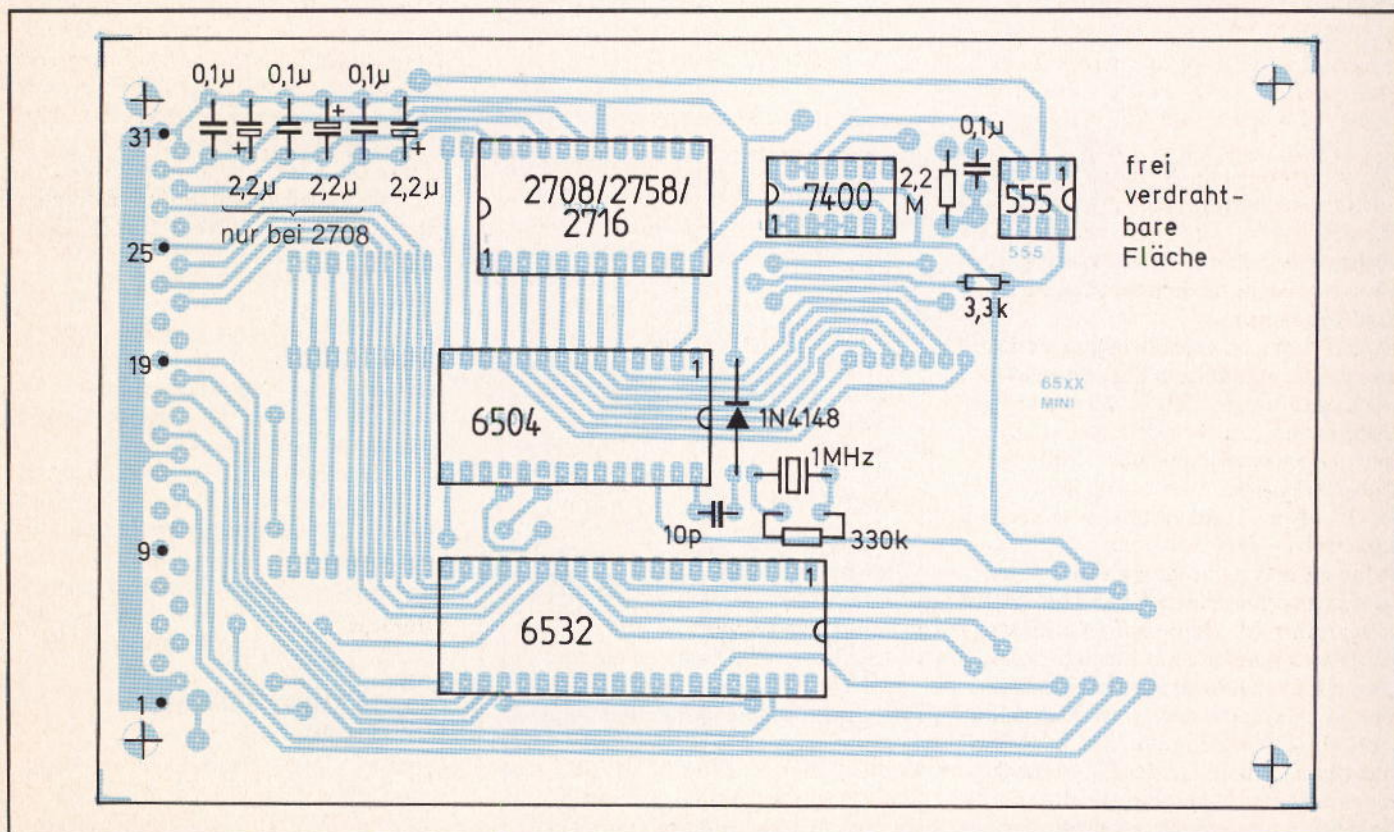


Bild 4. Bestückungsplan des 6504-Computers. Es sei erwähnt, daß das 2-KByte-EPROM 2716 z. T. schon preiswerter angeboten wird als der 5-V-/1-KByte-Typ 2758. Beim 2716 kann man entweder eine Hälfte „verschenken“ oder auch mit einem Schalter zwischen zwei 1-KByte-Betriebsprogrammen wählen

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 Unterprogrammsprünge 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 Unterprogrammsprünge 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	
11AA AAAA AAAA	EPROM (1 KByte)	C00...FFF 1 k
01AA AAAA AAAA	Expansion (1 KByte)	400...7FF 1 k

(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 Wolln-

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 Elektronik-laden, 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.

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 preiswerten, 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-

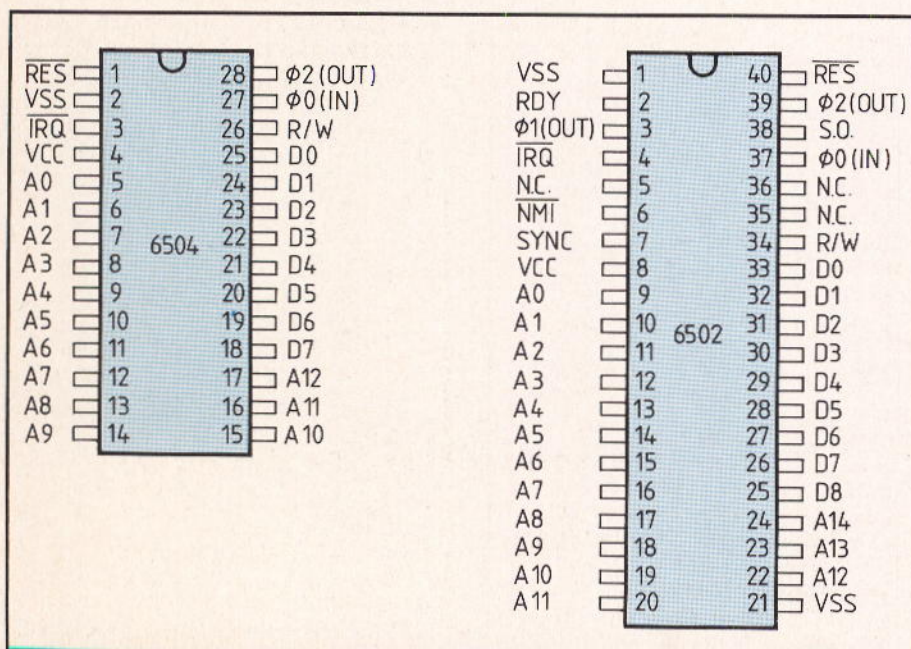


Bild 1. Anschlußbelegung der 28-Pin-CPU 6504, die im EMUF verwendet wird, im Vergleich zur software-kompatiblen CPU 6502, die man in vielen Tischcomputern findet

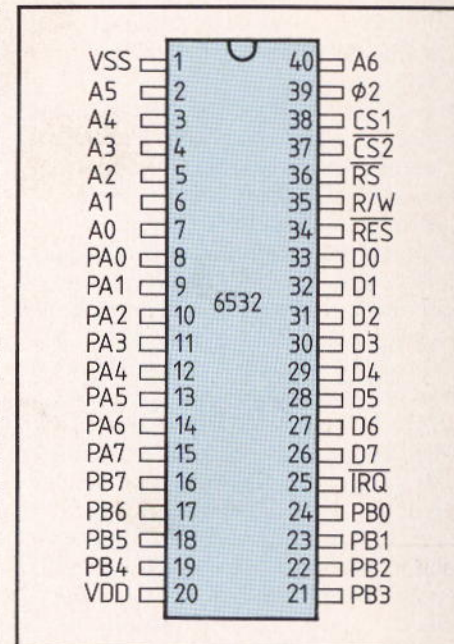


Bild 2. Anschlußbelegung des RIOT-Bausteins 6532 (RAM, I/O, Timer). Er nimmt typisch 100 mA bei 5 V Versorgungsspannung auf

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.

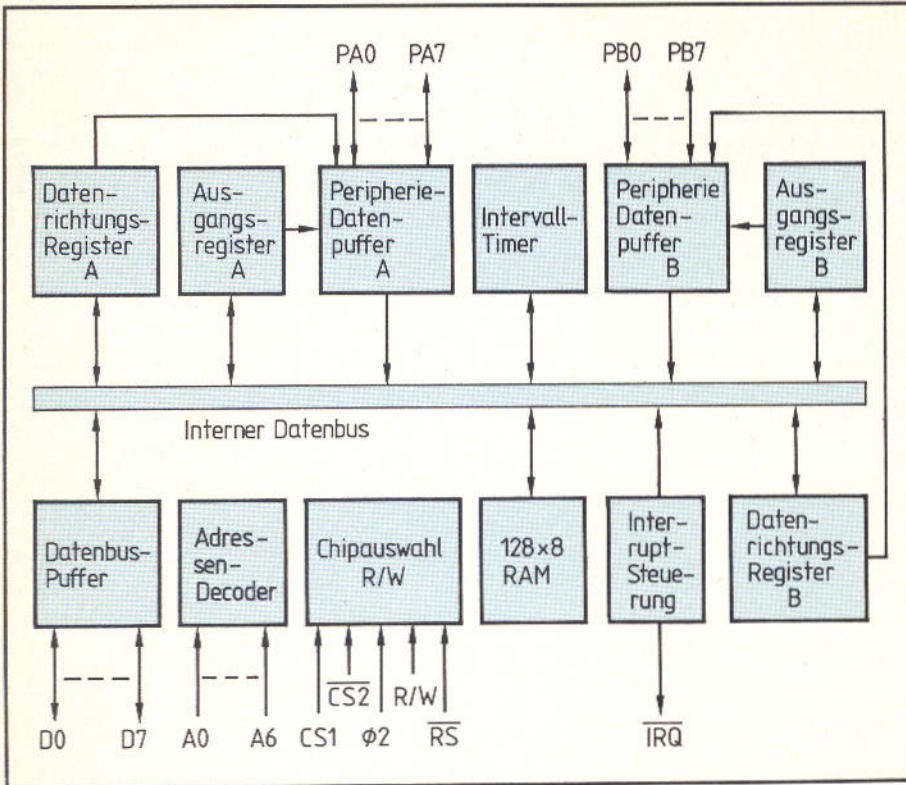


Bild 3. Internes Blockschaubild des RIOT-Bausteins 6532. Die Anschlüsse des Port PB können bis 3 mA liefern, da sie über Gegentaktausgänge verfügen

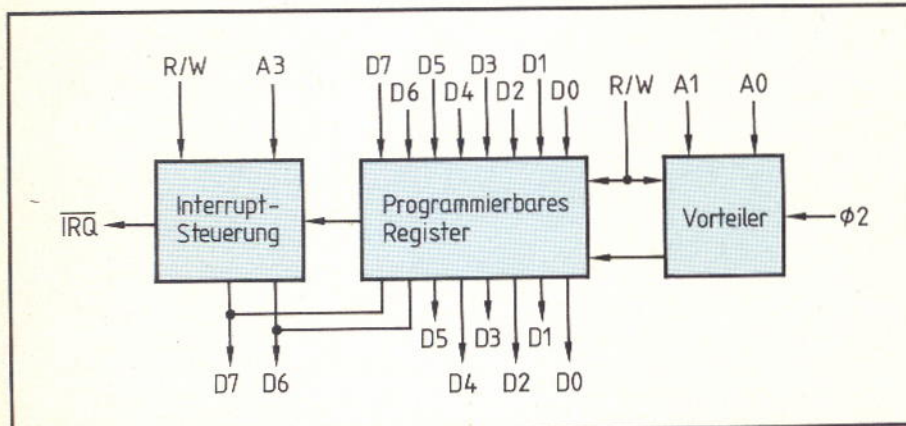


Bild 4. Struktur des programmierbaren Intervall-Timers im 6532. Der Vorteiler teilt den 1-MHz-Systemtakt je nach Programmierung durch 1, 8, 64 oder 1024. Das Timerregister kann mit einem 8-Bit-Wert geladen werden, von dem aus abwärts gezählt wird

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.

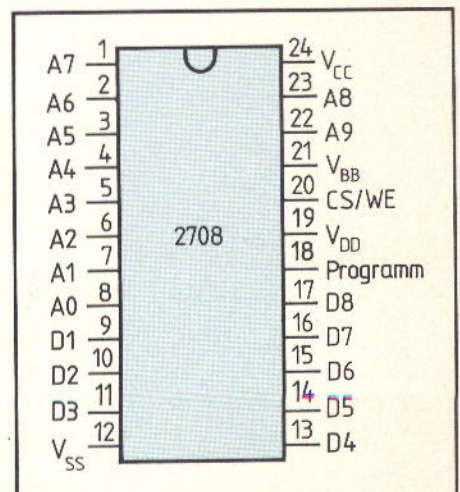


Bild 5. Das EPROM 2708, lange Zeit ein Standard-Chip, braucht noch drei Versorgungsspannungen

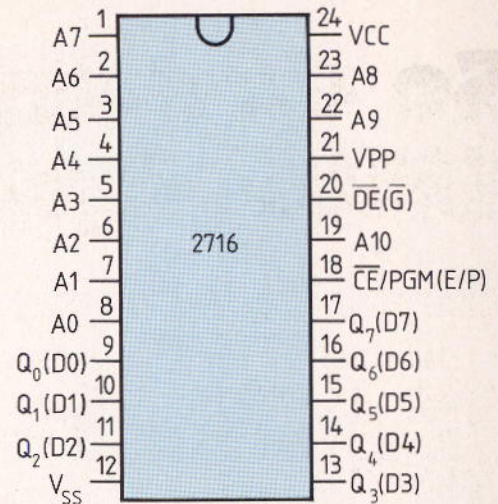
Bild 6 gibt das Anschlußbild des EPROMs 2716 wieder. Es sei darauf hingewiesen, daß Texas Instruments eine 2716-Version fertigt, die drei Versorgungsspannungen benötigt und zu den „normalen“ 2716-Versionen nicht kompatibel ist, so daß sie nicht im EMUF verwendet werden kann.

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 werden die entsprechenden RAM-ICs des AIM-65 herausgezogen.

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.

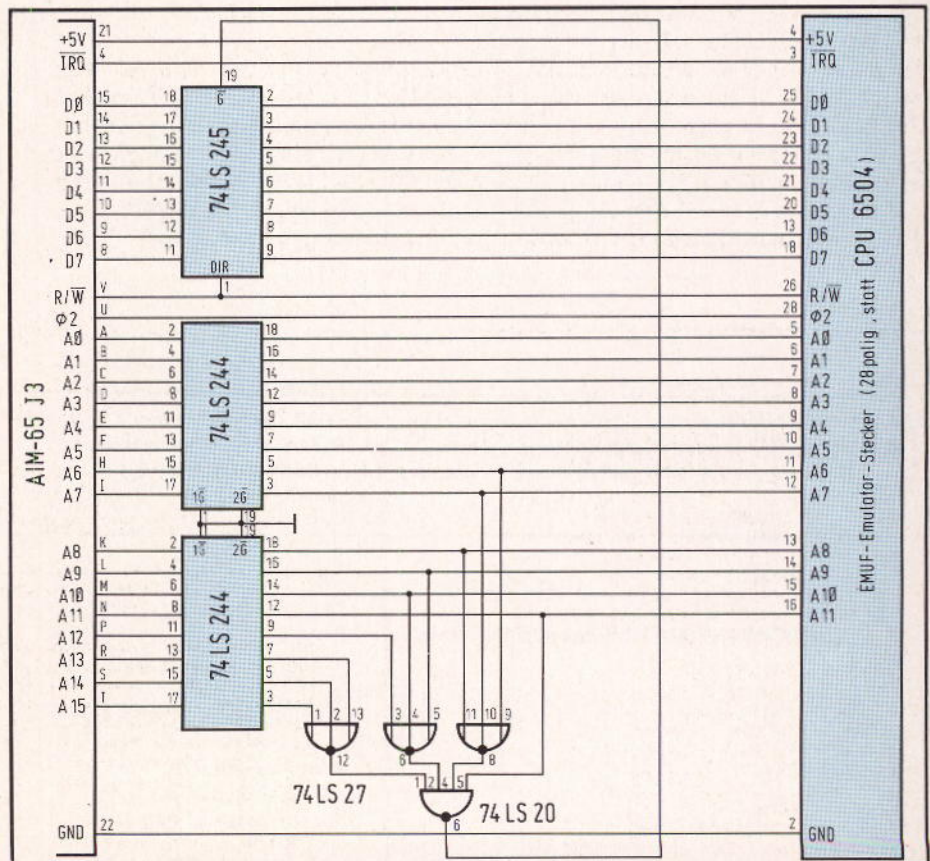
Wird der Interrupt verwendet, so muß die IRQ-Adresse des AIM in A400 auf den Anfang der Anwenderroutine umgestellt werden.

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 wie-

der, 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-Programmierungsmöglichkeit 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-Programmiers tips“ 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.

```

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 VORAUS. 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? 2000
WIEVIELE GERAETE AUF 650X-
BASIS WERDEN VORAUS. 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 ausnützbare

	EMUF	AIM-	CBM-	PET-	KIM-
	6504	65	3001	2001	1
PA	0800	A00F	E84F	E84F	1700
PAD	0801	A003	E843	E843	1701
PB	0802	A000	-----	-----	1702
PBD	0803	A002	-----	-----	1703
T1	0814	A494	-----	-----	1704
T8	0815	A495	-----	-----	1705
T64	0816	A496	-----	-----	1706
TK	0817	A497	-----	-----	1707
TI1	081C	A49C	-----	-----	170C
TI8	081D	A49D	-----	-----	170D
TI64	081E	A49E	-----	-----	170E
TIK	081F	A49F	-----	-----	170F
IRQL	0FFE	A404	0090	0219	17FE
IRQH	0FFF	A405	0091	021A	17FF

Herwig Feichtinger

EMUF-Programmierspit

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 auf 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-

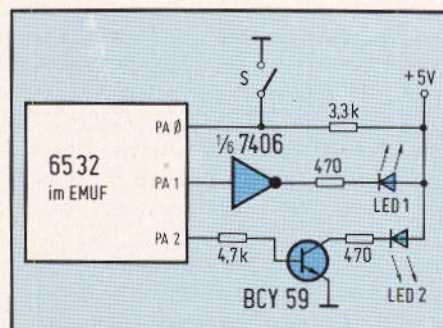


Bild 1. Will man einen Schalter an eine I/O-Leitung anschließen, empfiehlt sich ein Pull-Up-Widerstand (hier 3,3 kΩ). Leuchtdioden können über eine Transistor- oder TTL-Treiberschaltung angesteuert werden

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      ;EMUF-INITIALISIERUNG
0000 PA      = $800
0000 PAD     = $801
0000 PB      = $802
0000 PBD     = $803
0000      * = $FFC      ;RESET
0FFC      000C      .W0R RESET ;VECTOR
0FFE      * = $C00
0C00 RESET A2FF LDX #$FF      ;STACK-
0C02      9A      TXS          ;POINTER
0C03      A9F8 LDA #11111000
0C05      8D0108 STA PAD      ;PORT-
0C08      A93E LDA #10011110
0C0A      8D0308 STA PBD      ;RICHTUNG
0C0D      D8      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 Gekuld 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 D8	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 BD0108 43 STA DRA ;NUR PA6=EING.
6C0A A980 44 LDA #100000000
6C0C BD0308 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 B500 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 ;SHIFT LOCK CODE
6C31 20576C 65 SHIFT JSR KEY3
6C34 68 66 SAME PLA
6C35 67 ;PRESS 1 KEY
6C35 A205 68 KEY LDX #5 ;DURATION
6C37 48 69 PHA
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 BD0008 78 KEY2 STA PA ;CODE OUT
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 ;DURATION FOR SHIFT
6C59 48 88 PHA
6C5A 293F 89 AND ##3F
6C5C 20466C 90 JSR KEY1
6C5F A214 91 LDX ##14 ;DELAY FOR SH +SH.A
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
6CB0 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 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 EB nicht ab. Hier wäre es denkbar, mit 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
6CCB 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 B601 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 A080   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
    
```

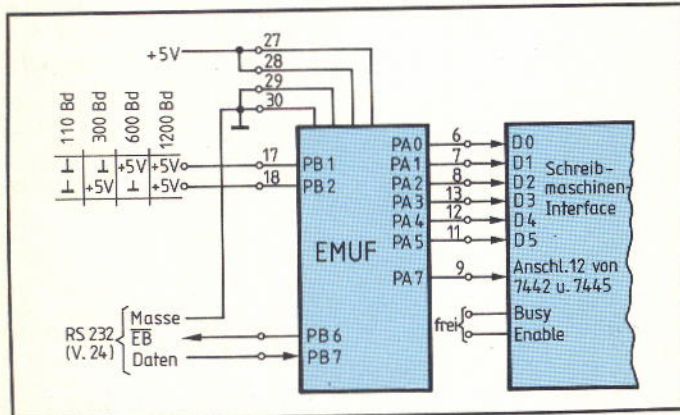


Bild 2. EMUF als V.24-Schnittstelle für den Schreibmaschinen-Drucker. Format des Eingangssignals: 8 Bit oder 7 Bit mit Parität (das 8. Bit wird in jedem Fall ignoriert)

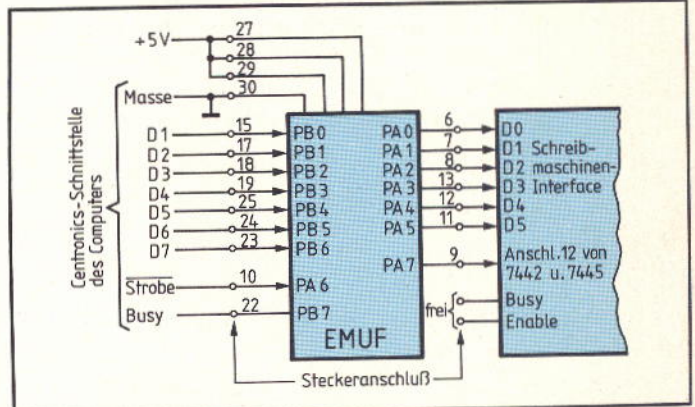


Bild 3. EMUF als Centronics-Schnittstelle für den Schreibmaschinen-Drucker



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, 029F, 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-Symbole 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.

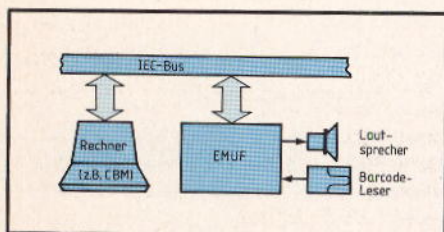


Bild 1. Am IEC-Bus muß der EMUF immer zuerst angesprochen werden. Die dabei verwendete Adresse merkt er sich

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 .. 6 IEC DATABUS
                                ;PORT B
                                ; 7      6      5      4      3      2      1      0
                                ;  BAR  BEL  -EOI  -NOAC  -NRDF  -OAC  -ATN
                                ;
                                ;
0060 PA EQU $800
0061 PAD EQU $801
0062 PB EQU $802
0063 PBD EQU $803
                                ;
0064 XTEMP1 EQU $0
0065 REF0 EQU $1
0066 REF1 EQU $2
0067 REF EQU $3
0068 BUF EQU $4
0069 CHCKL EQU $5
0070 CHCKH EQU $6
0071 XTEMP EQU $7
0072 PFLAG EQU $8
0073 PNT EQU $9
0074 FLAG EQU $A ; IEC MERKER ATN ..
0075 ZEICH EQU $B ; ZWSPEICHER
0076 COUNT EQU $C ; ZWSPEICHER
0077 PADR EQU $D ; FIRST TIME
0078 ZEIT EQU $E
                                ;
                                ;
                                ; INIT ROUTINE
0079 ORG $FFC
0080 DW $C00
                                ;
0081 ORG $C00 ; START
0082 RESET: LOX $FF ; STACKPOINTER
                                ;
0083 TXS
0084 LDA #00000000 ; ALL INPUT
0085 STA PAD
0086 LDA #00100100 ; SET UP
0087 STA PBD
0088 LDA #00000000 ; NOT READY NOT ACCEPT
0089 CLD
0090 D8
0091 SEI
0092 LDA #$FF
0093 STA PFLAG
0094 STA PADR ; START WERT
0095 JSR BELL1 ; AUSGEBEN TON FUER CHECK
0096 JMP MAIN
                                ; UPRGE
0097 BELL1: LDA #$FF ; BELL AUSGEREN
0098 LOP1: PIA
0099 LDA PB
0100 EOR #01000000 ; CHANGE
    
```



```

0C25 8D0200      STA      PB
0C28 A950        LDA      #50
0C2A EA          LOP2:   NOP
0C2B EA          NOP
0C2C EA          NOP
0C2D E901        SBC      #1
0C2F D6F9        BNE      LOP2
0C31 58          PLA
0C32 E901        SBC      #1
0C34 D0E9        BNE      LOP1
0C36 AD0000      LDA      PA
0C39 29BF        AND      #10111111
0C3B 8D0000      STA      PA
0C3E 60          RTS

;
0C3F 3500        ;READER: STX      XTEMP1
0C41 2450        BIT      PFFLAG
0C43 300E        BMI      START
0C45 A609        LDX      PNT
0C47 E404        EING:   CPX      BUF
0C49 F000        BEQ      START
0C4B 8D1000      LDA      BUFFER,X
0C4F E609        INC      PNT
0C50 A600        LDX      XTEMP1
0C52 60          RTS

;
0C53 2C0200      START:  BIT      PB
0C56 30F0        BMI      START
0C58 28F30C      JSR      SYNCH
0C5B 3602        STX      REF1
0C5D 3601        STX      REF0
0C5F 4601        LSR      REF0
0C61 28F30C      JSR      SYNCH
0C64 20A30C      JSR      AUSW
0C67 7076        BCC      FEHLER
0C69 20E20C      JSR      BYTE
0C6C 8504        STA      BUF
0C6E A200        LDX      #0
0C70 A900        LDA      #0
0C72 3503        STA      CHCKL
0C74 8506        STA      CHCKH
0C76 3607        VOR:   STX      XTEMP
0C78 20E20C      JSR      BYTE
0C7B A607        LDX      XTEMP

```

```

0C7D 9D1000      STA      BUFFER,X
0C80 10          CLC
0C81 6305        ADC      CHCKL
0C83 8505        STA      CHCKL
0C85 9002        BCC      NULL
0C87 E606        INC      CHCKH
0C89 E8          NULL:  INX
0C8A E404        CPX      BUF
0C8C D0E3        BNE      VOR
0C8E 20E20C      JSR      BYTE
0C91 C505        CMP      CHCKL
0C93 D04A        BNE      FEHLER
0C95 20E20C      JSR      BYTE
0C98 C506        CMP      CHCKH
0C9A D943        BNE      FEHLER
0C9C 201D0C      JSR      BELL1 ;READY MEI.DEN
0C9F A200        LDX      #0
0CA1 8609        STX      PNT
0CA3 3600        STX      PFFLAG
0CA5 4C470C      JMP      EING

;
0CAB A501        AUSW:  LDA      REF0
0CAA 4A          LSR      A
0CAB 6501        ADC      REF0
0CAD 8503        STA      REF
0CAF E403        CPX      REF
0CB1 1009        BPL      EINS
0CB3 8601        STX      REF0
0CB5 A501        LDA      REF0
0CB7 0A          ASL      A
0CB8 8502        STA      REF1
0CBA 10          CLC
0CBB 60          RTS
0CBC 8602        EINS:  STX      REF1
0CBE A502        LDA      REF1
0CC0 4A          LSR      A
0CC1 8501        STA      REF0
0CC3 30          SEC
0CC4 60          RTS

;
0CC5 40          BITTST: PHA
0CC6 A200        LDX      #0
0CC8 2C0200      BIT:   BIT      PB
0CCR 10F0        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

```

10000 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 Interndarstellung 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.

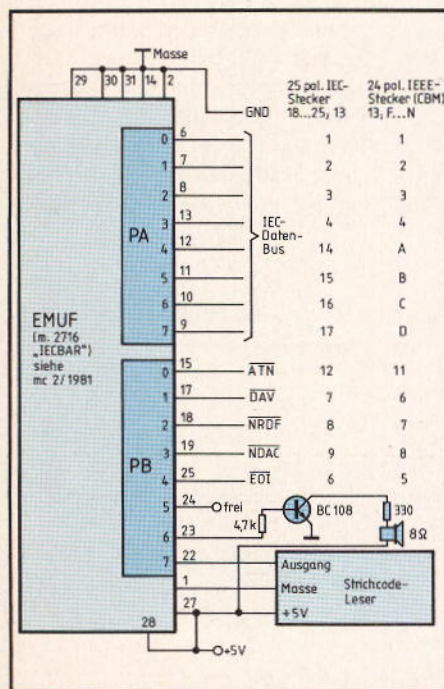


Bild 2. Anschluß von Strichcode-Leser, IEC-Bus-Stecker und Lautsprecher an den Einplatinen-Computer EMUF

```

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 2C6268  BIT      PB
0CD8 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
0CEE 88      DEY
0CEF D0F4    BNE      NBIT
0CF1 68      PLA
0CF2 60      RTS

0CF3 48      ;
SYNCH: PHA
0CF4 20C50C  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 8D0268  STA      PB
0D07 A00208  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 ;RETEN
0D19 AD0208  LDA      PB
0D1C 850A    STA      FLAG
0D1E A908    LDA      #%00001000 ;DAC
0D20 8D0208  STA      PB
0D23 A00208  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
    
```



```

;
;
0062 48      SEND1: PHA
0063 208800  SEND: JSR CKATN ;TESTER GGF
0066 AD0208  LDA PB
0069 2904    AND #Z00000100
006B F0F6    BEQ SEND ;WARTEN
006D 68      PLA
006E 49FF    EOR #Z11111111
0070 8D0008  STA PA
0073 A910    LDA #Z00010000 ;DAV
0075 8D0208  STA PB
0078 208800  CONSE: JSR CKATN
007B AD0208  LDA PB
007E 2908    AND #Z00001000
0080 F0F6    BEQ CONSE
0082 A912    LDA #Z00010010 ;PASSIV
0084 8D0208  STA PB
0087 68      RTS

;
0088 68      CKATN: RTS ;Z.Z.
;
0089 48      SENE01: PHA
008A 208800  SENE01: JSR CKATN
008D AD0208  LDA PB
0090 2904    AND #Z00000100
0092 F0F6    BEQ SENE01
0094 68      PLA
0095 49FF    EOR #Z11111111
0097 8D0008  STA PA
009A A900    LDA #Z00000000 ;EOI DAV
009C 8D0208  STA PB
009F 4C7800  JMP CONSE

;
00A2 20820D  MAIN: JSR GETCHA ;IEC ZEICHEN
00A3 8508    STA ZEICH ;RETZEN
00A7 A50A    LDA FLAG
00A9 2901    AND #Z00000001 ;ATN
00AB D0F5    BNE MAIN ;HIGH DANN NEIN
00AD A508    LDA ZEICH
00AF 2900    AND #400 ;TEST LISTEN TALK
00B1 C928    CMP #20
00B3 F007    BEQ LISTPA
00B5 C940    CMP #40
00B7 F006    BEQ TALPA
00B9 4CA20D  JMP MAIN

;
00BC 4CA20D  LISTPA: JMP MAIN ;HIER NICHT
;
00BF A50D    TALPA: LDA PADR ;PRIM ADRESSE
00C1 C9FF    CMP #FF
00C3 D009    BNE SK2 ;WEITER SONST
00C5 A50B    LDA ZEICH
00C7 290F    AND #0F
00C9 850D    STA PADR ;NEUE ADRESSE
00CB 4CD60D  JMP SK3
00CE A50B    SK2: LDA ZEICH
00D0 290F    AND #0F
00D2 C50D    CMP PADR ;VERGLEICH
00D4 D0CC    BNE MAIN
00D6 203F0C  SK3: JSR READER ;HOLE EIN ZEICHEN
00D9         ;FUER GET
00D9         ;GET WEGEN : UND ,
00D9         STA ZEICH
00DB 20020D  JSR GETCHA ;IGNORE SA
00DE         ;SA MUSS DA SEIN
00DE 20310D  JSR TALKON ;CHANGE DIR
00E1 A50B    LDA ZEICH
00E3 20890D  JSR SENE01 ;1 ZEICHEN SENDEN NUN
00E6 20409D  JSR TALKOF ;UMSCHALTEN
00E9 4CA20D  JMP MAIN ;ALLES VON VORNE

;
;
; RAM GEBIET
;
0010         ORG $10
;
0010         BUFFER DS 24 ;PUFFER BARCODE
0028         ENE DS 1 ;CHECK
;
0060         END

```

SYMBOL TABLE

AUSW	0CA8
BELL1	0C1D
BIT	0CC8
BLTTST	0CC5
BUF	0004
BUFFER	0010
BYTE	0CE2
CHCKH	0006
CHCKL	0005
CKATN	0088
CONSE	0D78
COUNT	000C
EING	0C47
EINS	0C8C
ENE	0028
FEHLER	0CDF
FLAG	000A
GETCHA	0D92
LISTPA	0DBC
LOP1	0C1F
LOP2	0C2A
LOPA	0D67
LOPB	0D23
MAIN	0DA2
NBIT	0CE3
NULL	0C89
PA	0000
PAD	0001
PADR	0000
PB	0002
PBD	0003
PFFLAG	0000
PNT	0009
READER	0C3F
REF	0003
REF0	0001
REF1	0002
RESET	0C00
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
ZAEHL	0CCD
ZEICH	000B
ZEIT	0008

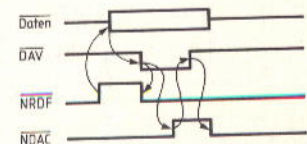


Bild 4. Handshake-Verhalten beim IEC-Bus. Die Signale können in beide Richtungen übertragen werden

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.

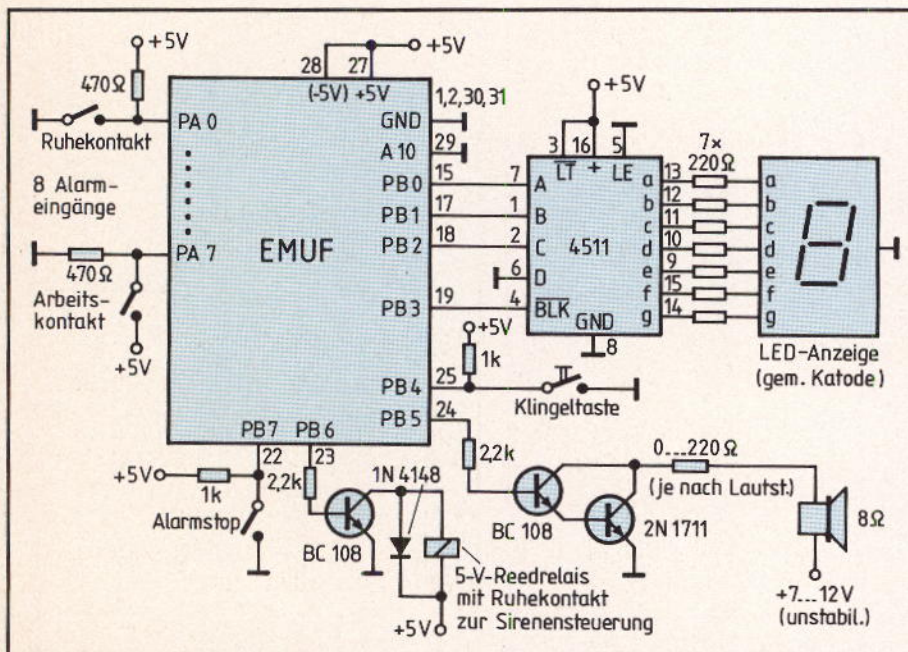


Bild 1. Verdrahtung des EMUF. An jeden der acht Alarmeingänge kann wahlweise ein Arbeits- oder Ruhekontakt angeschlossen werden

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 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 durch Interrupt-Programmierung 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 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
0000 MUS = $D00
0000 PA = $800
0000 PB = $802
0000 PBD = $803
0000 CLK = $81F
0000 RESV = $FFC
0000 IRQV = $FFE
0000 RES A2FF LDX £$FF ;RESET
0002 8602 STX LST ;NORMAL:
0004 9A TXS ;PA0-7=H
0005 A96F LDA £%01101111
0007 8D0308 STA PBD
    
```


OCOA	A9D0	LDA £%11010000	OC8E SLOOP	E000	CPX £0
OCOC	8D0208	STA PB	OC90	D008	BNE CONT
OCOF	8D1F08	STA CLK ;IRQ-START	OC92	A60F	LDX XSAV
OC12	58	CLI	OC94	C60E	DEC TIMER
OC13	D8	CLD	OC96	D0F6	BNE SLOOP
OC14 FRST	A205	LDX £5 ;ERSTE	OC98	F016	BEQ SEX
OC16 LP1	BDB20C	LDA INIT,X ;MEL.	OC9A CONT	8D0208	STA PB
OC19	9503	STA WORK,X	OC9D	CA	DEX
OC1B	CA	DEX	OC9E	C60B	DEC LIMIT+2
OC1C	10F8	BPL LP1	OCA0	DOEC	BNE SLOOP
OC1E DEB	A200	LDX £0 ;KLINGEL-	OCA2	C60A	DEC LIMIT+1
OC20 WAIT	ADO208	LDA PB ;TASTE?	OCA4	DOE8	BNE SLOOP
OC23	2910	AND £%00010000	OCA6	A403	LDY WORK
OC25	D0F7	BNE DEB ;NEIN	OCA8	840A	STY LIMIT+1
OC27	CA	DEX	OCAA	C609	DEC LIMIT
OC28	D0F6	BNE WAIT	OCAC	DOE0	BNE SLOOP
OC2A GO	A000	LDY £0 ;MUSIK-	OCAE	A9FF	LDA £\$FF
OC2C	B107	LDA (WORK+4),Y	OCB0 SEX	58	CLI
OC2E	C9FF	CMP £\$FF ;PROGR.	OCB1	60	RTS
OC30	F0E2	BEQ FRST	OCB2 INIT	30	.BYT 48,2,1,\$FF
OC32	E607	INC WORK+4 ;AEHNL.	OCB3	02	
OC34	D002	BNE *+4 ;JIM BUTTER-	OCB4	01	
OC36	E608	INC WORK+5 ;FIELD'S	OCB5	FF	
OC38	C9FA	CMP £\$FA ;MUSICBOX	OCB6	000D	.WOR MUS
OC3A	FOE4	BEQ WAIT	OCB8		;INTERRUPT-ROUTINE
OC3C NEXT	900F	BCC NOTE ;AUS	OCB8		;FUER ALARMANLAGE
OC3E	E9FB	SBC £\$FB ;DEM	OCB8 IRQ	48	PHA
OC40	AA	TAX ;FIRST	OCB9	ADO208	LDA PB ;STOPTASTE?
OC41	B107	LDA (WORK+4),Y	OCBC	3009	BMI IRQ0 ;NEIN
OC43	E607	INC WORK+4 ;BOOK	OCBE	0940	ORA £%01000000
OC45	D002	BNE *+4 ;OF KIM	OCCE	29F7	AND £%11110111
OC47	E608	INC WORK+5	OCC2	8D0208	STA PB ;AL.+DISP.
OC49	9503	STA WORK,X	OCC5	D030	BNE IRQ2 ;AUS
OC4B	B0DD	BCS GO	OCC7 IRQ0	AD0008	LDA PA
OC4D NOTE	A603	LDX WORK	OCCA	4502	EOR LST ;AENDERG.?
OC4F	860A	STX LIMIT+1	OCCE	F019	BEQ IRQ1
OC51	A604	LDX WORK+1	OCCD	8600	STX SAVX ;JA
OC53	A8	TAY	OCDO	AEO008	LDX PA
OC54	3002	BMI OVER	OCD3	8602	STX LST ;X RETTEN
OC56	A201	LDX £1	OCD5	A2FF	LDX £\$FF
OC58 OVER	8609	STX LIMIT	OCD7 IRQ3	2A	ROL A ;DISPLAY-
OC5A	297F	AND £\$7F	OCD8	E8	INX ;CODE
OC5C	850C	STA VAL2	OCD9	90FC	BCC IRQ3 ;ERZEUGEN
OC5E	F002	BEQ HUSH	OCDB	8A	TXA
OC60	850D	STA VAL1	OCDC	A600	LDX SAVX
OC62 HUSH	A50C	LDA VAL2	OCDE	0998	ORA £%10011000
OC64	2506	AND WORK+3	OCEO	8D0208	STA PB ;AL.EIN
OC66	F004	BEQ ON	OCE3	A9FA	LDA £DUR
OC68	E60D	INC VAL1	OCE5	8501	STA TI
OC6A	C60C	DEC VAL2	OCE7 IRQ1	A501	LDA TI ;ZEIT
OC6C ON	A60C	LDX VAL2	OCE9	F00C	BEQ IRQ2 ;VORBEI?
OC6E	78	SEI ;WEGEN PB-	OCEB	C601	DEC TI
OC6F	ADO208	LDA PB ;PROBL.	OCED	D008	BNE IRQ2
OC72	0920	ORA £%00100000	OCEF	ADO208	LDA PB ;JA
OC74	20880C	JSR SOUND ;BEI IRQ	OCF2	0940	ORA £%01000000
OC77	30B1	BMI GO	OCF4	8D0208	STA PB ;AL.AUS
OC79	A60D	LDX VAL1	OCF7 IRQ2	A964	LDA £100
OC7B	78	SEI	OCF9	8D1F08	STA CLK
OC7C	ADO208	LDA PB	OCFC	68	PLA
OC7F	29DF	AND £%11011111	OCFD	40	RTI
OC81	20880C	JSR SOUND	OCFE		*=RESV
OC84	30A4	BMI GO	OFFC	000C	.WOR RES
OC86	10DA	BPL HUSH	OFFE		*=IRQV
OC88 SOUND	A405	LDY WORK+2	OFFE	B80C	.WOR IRQ
OC8A	840E	STY TIMER	1000		.END
OC8C	860F	STX XSAV	1000		ERRORS= 0C00

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 FF 39 40 44 39 2F A4 29 2F
( ) 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	Tief		Hoch	
	Hex (kurz)	Hex (lang)	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-

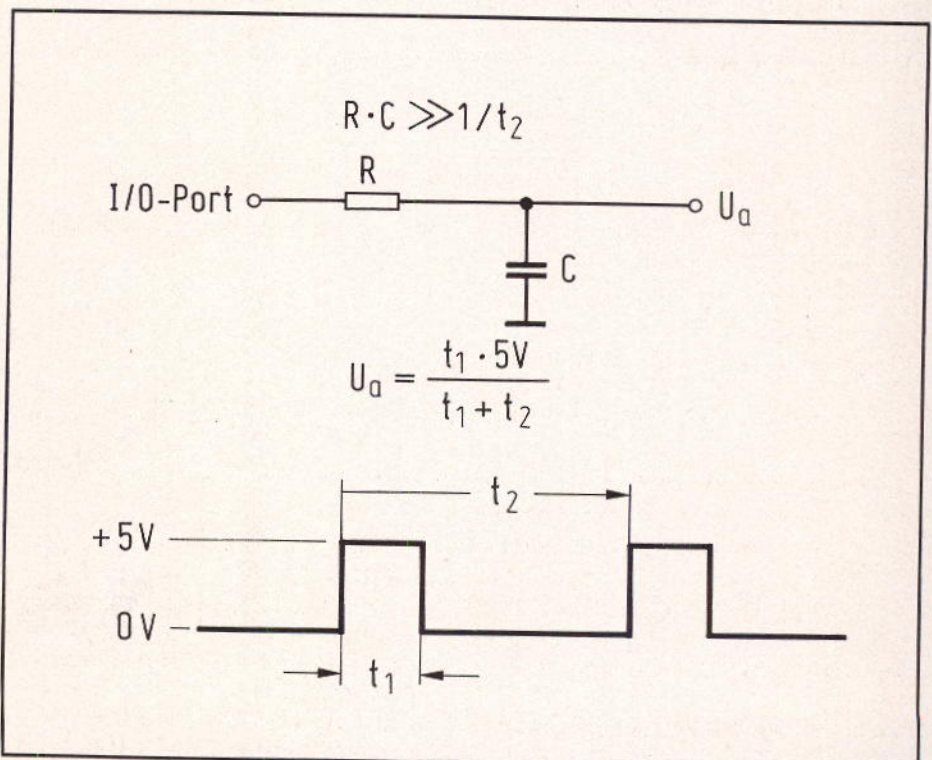


Bild 1. Prinzip der Software-D/A-Umsetzung


```

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 4C000C JMP START
010F *= $C00
OC00 START A901 LDA £1 ;START
OC02 8D9EA4 STA CL64 ;TIMER
OC05 A908 LDA £8 ;PB3=
OC07 8D02A0 STA PBD ;OUTP.
OC0A 58 CLI
OC0B 60 RTS
OC0C IRQ 48 PHA
OC0D A500 LDA U ;U=0?
OC0F FO13 BEQ HI ;YES
OC11 AD00A0 LDA PB
OC14 2908 AND £8 ;PB3=H?
OC16 D00C BNE HI ;YES
OC18 AD00A0 LDA PB
OC1B 0908 ORA £8 ;NO, MAKE
OC1D 8D00A0 STA PB ;IT H
OC20 A500 LDA U ;T=T1
OC22 D011 BNE TI ;JUMP
OC24 HI AD00A0 LDA PB ;MAKE
OC27 29F7 AND £$F7 ;PB3=L
OC29 8D00A0 STA PB
OC2C A900 LDA £0 ;T2=256
OC2E 38 SEC ;T=
OC2F E500 SBC U ;T2-T1
OC31 D002 BNE TI
OC33 A9FF LDA £$FF ;AVOID
OC35 TI 8D9EA4 STA CL64 ;T=0
OC38 68 PLA
OC39 40 RTI
OC3A *=IRQV
A404 OC0C .WOR IRQ
A406 .END
A406 ERRORS= 0000

```

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

gen. 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.

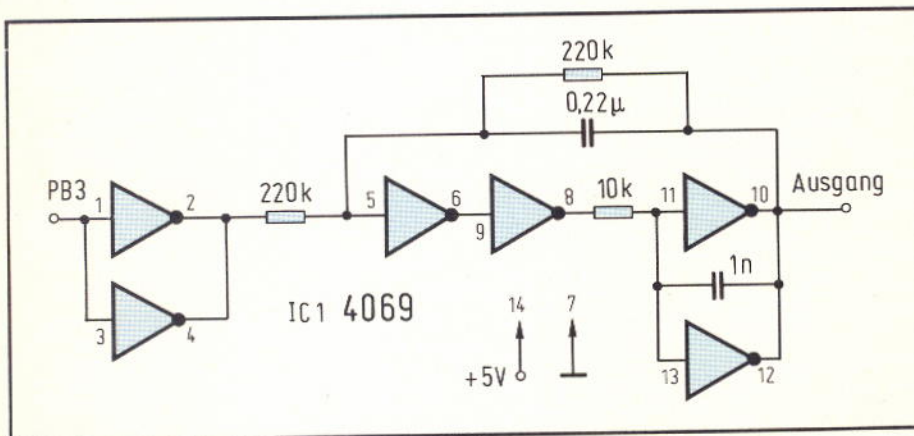


Bild 3. Verbesserte Tiefpaß-Schaltung mit niederohmigem Ausgang

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.

Programmierte EPROMs für den EMUF

Für die meisten der in diesem Heft beschriebenen EMUF-Applikationen liefert der Franzis-Software-Service (Postfach 37 01 20, 8000 München 37) fertig programmierte EPROMs vom Typ 2716, für die nur eine Versorgungsspannung von +5 V erforderlich ist. Ein EPROM enthält jeweils zwei EMUF-Programme ($2 \times 1\text{ KByte}$) und kostet 19,50 DM plus Versandkosten. Bitte richten Sie entsprechende Anfragen und Bestellungen direkt an den Franzis-Software-Service. 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-Ausgabeeinheit zu schaffen, die über den IEC-Bus an einen CBM-Rechner angeschlossen werden kann.

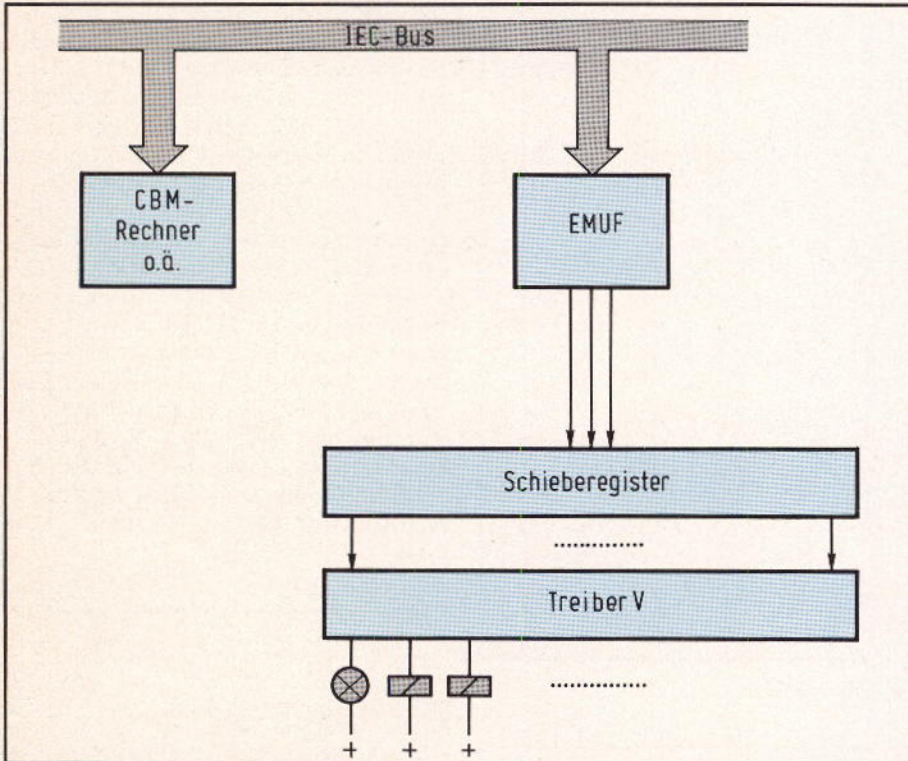


Bild 1. Anschlußbeispiel für den Mini-Logger

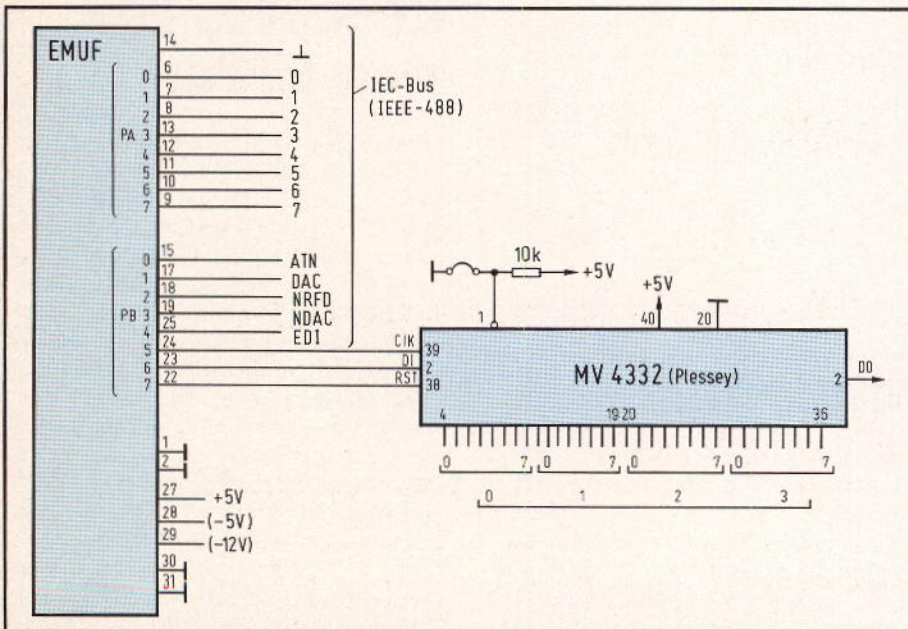


Bild 2. Die Steckbelegung des EMUF

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 Leitung 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


```

00C6 48 PHA ;RETTEN
00C7 A00208 LDA PB
00C8 850A STA FLAG
00C9 A908 LDA #200001000
00CA 800208 STA PB
00CB A00208 LOPB: #200000010
00CC 2902 AND #200000000
00CD F0F9 BEQ LOPB
00CE A900 LDA #200000000
00CF 800208 STA PB
00D0 68 PLA
00D1 60 RTS

;
00D2 2901 TALKON: LDA
00D3 F0F9 BEQ AND
00D4 A912 LDA #200010010
00D5 800208 STA PB
00D6 A9FF LDA #211111111
00D7 800108 STA #201010010
00D8 A952 LDA #201010010
00D9 800308 STA #200010010
00DA A912 LDA #200010010
00DB 800208 STA PB
00DC 60 RTS

;
00DD A900 TALKOFF: LDA
00DE 800108 STA #200000000
00DF A900 LDA #200000000
00E0 800208 STA #200000000
00E1 A94C LDA #201001100
00E2 800308 STA #200000000
00E3 A900 LDA #200000000
00E4 800208 STA #200000000
00E5 60 RTS

;
00E6 48 SEND1: PHA
00E7 201600 JSR SEND1:
00E8 A00208 LDA SEND1:
00E9 F0F6 BEQ SEND1:
00EA 68 PLA
00EB 49FF EOR SEND1:
00EC 800008 STA SEND1:
00ED A910 LDA #200010000
00EE 800208 STA #200010000
00EF 201600 JSR SEND1:
00F0 A00208 LDA SEND1:
00F1 2908 AND #200001000
00F2 F0F6 BEQ CONSE:
00F3 800008 EOR CONSE:
00F4 800008 STA CONSE:
00F5 A910 LDA #200010010
00F6 800208 STA #200010010
00F7 60 RTS

;
00F8 60 CKATN: RTS
00F9 48 SEND01: PHA
00FA 201600 JSR SEND01:
00FB A00208 LDA SEND01:
00FC F0F6 BEQ SEND01:
00FD 49FF EOR SEND01:
00FE 800008 STA SEND01:
00FF A910 LDA #200010010
0100 800208 STA #200010010
0101 60 RTS

;
0102 48 MAIN:
0103 20900C JSR
0104 8508 STA
0105 60 MAIN1:
0106 4C0600 JMP

;
0107 48 CKATN: RTS
0108 201600 JSR
0109 A00208 LDA
010A 2904 AND #200000100
010B F0F6 BEQ SEND01:
010C 68 PLA
010D 49FF EOR SEND01:
010E 800008 STA SEND01:
010F A900 LDA #200000000
0110 800208 STA #200000000
0111 60 RTS

;
0112 60 CKATN: RTS
0113 48 SEND01: PHA
0114 201600 JSR SEND01:
0115 A00208 LDA SEND01:
0116 2904 AND #200000100
0117 F0F6 BEQ SEND01:
0118 49FF EOR SEND01:
0119 800008 STA SEND01:
011A A900 LDA #200000000
011B 800208 STA #200000000
011C 60 RTS

;
011D 48 MAIN:
011E 20900C JSR
011F 8508 STA
0120 4C0600 JMP

;
0121 48 CKATN: RTS
0122 201600 JSR
0123 A00208 LDA
0124 2904 AND #200000100
0125 F0F6 BEQ SEND01:
0126 68 PLA
0127 49FF EOR SEND01:
0128 800008 STA SEND01:
0129 A900 LDA #200000000
012A 800208 STA #200000000
012B 60 RTS

;
012C 48 MAIN:
012D 20900C JSR
012E 8508 STA
012F 4C0600 JMP

;
0130 48 CKATN: RTS
0131 201600 JSR
0132 A00208 LDA
0133 2904 AND #200000100
0134 F0F6 BEQ SEND01:
0135 68 PLA
0136 49FF EOR SEND01:
0137 800008 STA SEND01:
0138 A900 LDA #200000000
0139 800208 STA #200000000
013A 60 RTS

;
013B 48 MAIN:
013C 20900C JSR
013D 8508 STA
013E 4C0600 JMP

;
013F 48 CKATN: RTS
0140 201600 JSR
0141 A00208 LDA
0142 2904 AND #200000100
0143 F0F6 BEQ SEND01:
0144 68 PLA
0145 49FF EOR SEND01:
0146 800008 STA SEND01:
0147 A900 LDA #200000000
0148 800208 STA #200000000
0149 60 RTS

;
014A 48 MAIN:
014B 20900C JSR
014C 8508 STA
014D 4C0600 JMP

;
014E 48 CKATN: RTS
014F 201600 JSR
0150 A00208 LDA
0151 2904 AND #200000100
0152 F0F6 BEQ SEND01:
0153 68 PLA
0154 49FF EOR SEND01:
0155 800008 STA SEND01:
0156 A900 LDA #200000000
0157 800208 STA #200000000
0158 60 RTS

;
0159 48 MAIN:
015A 20900C JSR
015B 8508 STA
015C 4C0600 JMP

;
015D 48 CKATN: RTS
015E 201600 JSR
015F A00208 LDA
0160 2904 AND #200000100
0161 F0F6 BEQ SEND01:
0162 68 PLA
0163 49FF EOR SEND01:
0164 800008 STA SEND01:
0165 A900 LDA #200000000
0166 800208 STA #200000000
0167 60 RTS

;
0168 48 MAIN:
0169 20900C JSR
016A 8508 STA
016B 4C0600 JMP

;
016C 48 CKATN: RTS
016D 201600 JSR
016E A00208 LDA
016F 2904 AND #200000100
0170 F0F6 BEQ SEND01:
0171 68 PLA
0172 49FF EOR SEND01:
0173 800008 STA SEND01:
0174 A900 LDA #200000000
0175 800208 STA #200000000
0176 60 RTS

;
0177 48 MAIN:
0178 20900C JSR
0179 8508 STA
017A 4C0600 JMP

;
017B 48 CKATN: RTS
017C 201600 JSR
017D A00208 LDA
017E 2904 AND #200000100
017F F0F6 BEQ SEND01:
0180 68 PLA
0181 49FF EOR SEND01:
0182 800008 STA SEND01:
0183 A900 LDA #200000000
0184 800208 STA #200000000
0185 60 RTS

;
0186 48 MAIN:
0187 20900C JSR
0188 8508 STA
0189 4C0600 JMP

;
018A 48 CKATN: RTS
018B 201600 JSR
018C A00208 LDA
018D 2904 AND #200000100
018E F0F6 BEQ SEND01:
018F 68 PLA
0190 49FF EOR SEND01:
0191 800008 STA SEND01:
0192 A900 LDA #200000000
0193 800208 STA #200000000
0194 60 RTS

;
0195 48 MAIN:
0196 20900C JSR
0197 8508 STA
0198 4C0600 JMP

;
0199 48 CKATN: RTS
019A 201600 JSR
019B A00208 LDA
019C 2904 AND #200000100
019D F0F6 BEQ SEND01:
019E 68 PLA
019F 49FF EOR SEND01:
01A0 800008 STA SEND01:
01A1 A900 LDA #200000000
01A2 800208 STA #200000000
01A3 60 RTS

;
01A4 48 MAIN:
01A5 20900C JSR
01A6 8508 STA
01A7 4C0600 JMP

;
01A8 48 CKATN: RTS
01A9 201600 JSR
01AA A00208 LDA
01AB 2904 AND #200000100
01AC F0F6 BEQ SEND01:
01AD 68 PLA
01AE 49FF EOR SEND01:
01AF 800008 STA SEND01:
01B0 A900 LDA #200000000
01B1 800208 STA #200000000
01B2 60 RTS

;
01B3 48 MAIN:
01B4 20900C JSR
01B5 8508 STA
01B6 4C0600 JMP

;
01B7 48 CKATN: RTS
01B8 201600 JSR
01B9 A00208 LDA
01BA 2904 AND #200000100
01BB F0F6 BEQ SEND01:
01BC 68 PLA
01BD 49FF EOR SEND01:
01BE 800008 STA SEND01:
01BF A900 LDA #200000000
01C0 800208 STA #200000000
01C1 60 RTS

;
01C2 48 MAIN:
01C3 20900C JSR
01C4 8508 STA
01C5 4C0600 JMP

;
01C6 48 CKATN: RTS
01C7 201600 JSR
01C8 A00208 LDA
01C9 2904 AND #200000100
01CA F0F6 BEQ SEND01:
01CB 68 PLA
01CC 49FF EOR SEND01:
01CD 800008 STA SEND01:
01CE A900 LDA #200000000
01CF 800208 STA #200000000
01D0 60 RTS

;
01D1 48 MAIN:
01D2 20900C JSR
01D3 8508 STA
01D4 4C0600 JMP

;
01D5 48 CKATN: RTS
01D6 201600 JSR
01D7 A00208 LDA
01D8 2904 AND #200000100
01D9 F0F6 BEQ SEND01:
01DA 68 PLA
01DB 49FF EOR SEND01:
01DC 800008 STA SEND01:
01DD A900 LDA #200000000
01DE 800208 STA #200000000
01DF 60 RTS

;
01E0 48 MAIN:
01E1 20900C JSR
01E2 8508 STA
01E3 4C0600 JMP

;
01E4 48 CKATN: RTS
01E5 201600 JSR
01E6 A00208 LDA
01E7 2904 AND #200000100
01E8 F0F6 BEQ SEND01:
01E9 68 PLA
01EA 49FF EOR SEND01:
01EB 800008 STA SEND01:
01EC A900 LDA #200000000
01ED 800208 STA #200000000
01EE 60 RTS

;
01EF 48 MAIN:
01F0 20900C JSR
01F1 8508 STA
01F2 4C0600 JMP

;
01F3 48 CKATN: RTS
01F4 201600 JSR
01F5 A00208 LDA
01F6 2904 AND #200000100
01F7 F0F6 BEQ SEND01:
01F8 68 PLA
01F9 49FF EOR SEND01:
01FA 800008 STA SEND01:
01FB A900 LDA #200000000
01FC 800208 STA #200000000
01FD 60 RTS

;
01FE 48 MAIN:
01FF 20900C JSR
0200 8508 STA
0201 4C0600 JMP

;
0202 48 CKATN: RTS
0203 201600 JSR
0204 A00208 LDA
0205 2904 AND #200000100
0206 F0F6 BEQ SEND01:
0207 68 PLA
0208 49FF EOR SEND01:
0209 800008 STA SEND01:
020A A900 LDA #200000000
020B 800208 STA #200000000
020C 60 RTS

;
020D 48 MAIN:
020E 20900C JSR
020F 8508 STA
0210 4C0600 JMP

;
0211 48 CKATN: RTS
0212 201600 JSR
0213 A00208 LDA
0214 2904 AND #200000100
0215 F0F6 BEQ SEND01:
0216 68 PLA
0217 49FF EOR SEND01:
0218 800008 STA SEND01:
0219 A900 LDA #200000000
021A 800208 STA #200000000
021B 60 RTS

;
021C 48 MAIN:
021D 20900C JSR
021E 8508 STA
021F 4C0600 JMP

;
0220 48 CKATN: RTS
0221 201600 JSR
0222 A00208 LDA
0223 2904 AND #200000100
0224 F0F6 BEQ SEND01:
0225 68 PLA
0226 49FF EOR SEND01:
0227 800008 STA SEND01:
0228 A900 LDA #200000000
0229 800208 STA #200000000
022A 60 RTS

;
022B 48 MAIN:
022C 20900C JSR
022D 8508 STA
022E 4C0600 JMP

;
022F 48 CKATN: RTS
0230 201600 JSR
0231 A00208 LDA
0232 2904 AND #200000100
0233 F0F6 BEQ SEND01:
0234 68 PLA
0235 49FF EOR SEND01:
0236 800008 STA SEND01:
0237 A900 LDA #200000000
0238 800208 STA #200000000
0239 60 RTS

;
023A 48 MAIN:
023B 20900C JSR
023C 8508 STA
023D 4C0600 JMP

;
023E 48 CKATN: RTS
023F 201600 JSR
0240 A00208 LDA
0241 2904 AND #200000100
0242 F0F6 BEQ SEND01:
0243 68 PLA
0244 49FF EOR SEND01:
0245 800008 STA SEND01:
0246 A900 LDA #200000000
0247 800208 STA #200000000
0248 60 RTS

;
0249 48 MAIN:
024A 20900C JSR
024B 8508 STA
024C 4C0600 JMP

;
024D 48 CKATN: RTS
024E 201600 JSR
024F A00208 LDA
0250 2904 AND #200000100
0251 F0F6 BEQ SEND01:
0252 68 PLA
0253 49FF EOR SEND01:
0254 800008 STA SEND01:
0255 A900 LDA #200000000
0256 800208 STA #200000000
0257 60 RTS

;
0258 48 MAIN:
0259 20900C JSR
025A 8508 STA
025B 4C0600 JMP

;
025C 48 CKATN: RTS
025D 201600 JSR
025E A00208 LDA
025F 2904 AND #200000100
0260 F0F6 BEQ SEND01:
0261 68 PLA
0262 49FF EOR SEND01:
0263 800008 STA SEND01:
0264 A900 LDA #200000000
0265 800208 STA #200000000
0266 60 RTS

;
0267 48 MAIN:
0268 20900C JSR
0269 8508 STA
026A 4C0600 JMP

;
026B 48 CKATN: RTS
026C 201600 JSR
026D A00208 LDA
026E 2904 AND #200000100
026F F0F6 BEQ SEND01:
0270 68 PLA
0271 49FF EOR SEND01:
0272 800008 STA SEND01:
0273 A900 LDA #200000000
0274 800208 STA #200000000
0275 60 RTS

;
0276 48 MAIN:
0277 20900C JSR
0278 8508 STA
0279 4C0600 JMP

;
027A 48 CKATN: RTS
027B 201600 JSR
027C A00208 LDA
027D 2904 AND #200000100
027E F0F6 BEQ SEND01:
027F 68 PLA
0280 49FF EOR SEND01:
0281 800008 STA SEND01:
0282 A900 LDA #200000000
0283 800208 STA #200000000
0284 60 RTS

;
0285 48 MAIN:
0286 20900C JSR
0287 8508 STA
0288 4C0600 JMP

;
0289 48 CKATN: RTS
028A 201600 JSR
028B A00208 LDA
028C 2904 AND #200000100
028D F0F6 BEQ SEND01:
028E 68 PLA
028F 49FF EOR SEND01:
0290 800008 STA SEND01:
0291 A900 LDA #200000000
0292 800208 STA #200000000
0293 60 RTS

;
0294 48 MAIN:
0295 20900C JSR
0296 8508 STA
0297 4C0600 JMP

;
0298 48 CKATN: RTS
0299 201600 JSR
029A A00208 LDA
029B 2904 AND #200000100
029C F0F6 BEQ SEND01:
029D 68 PLA
029E 49FF EOR SEND01:
029F 800008 STA SEND01:
02A0 A900 LDA #200000000
02A1 800208 STA #200000000
02A2 60 RTS

;
02A3 48 MAIN:
02A4 20900C JSR
02A5 8508 STA
02A6 4C0600 JMP

;
02A7 48 CKATN: RTS
02A8 201600 JSR
02A9 A00208 LDA
02AA 2904 AND #200000100
02AB F0F6 BEQ SEND01:
02AC 68 PLA
02AD 49FF EOR SEND01:
02AE 800008 STA SEND01:
02AF A900 LDA #200000000
02B0 800208 STA #200000000
02B1 60 RTS

;
02B2 48 MAIN:
02B3 20900C JSR
02B4 8508 STA
02B5 4C0600 JMP

;
02B6 48 CKATN: RTS
02B7 201600 JSR
02B8 A00208 LDA
02B9 2904 AND #200000100
02BA F0F6 BEQ SEND01:
02BB 68 PLA
02BC 49FF EOR SEND01:
02BD 800008 STA SEND01:
02BE A900 LDA #200000000
02BF 800208 STA #200000000
02C0 60 RTS

;
02C1 48 MAIN:
02C2 20900C JSR
02C3 8508 STA
02C4 4C0600 JMP

;
02C5 48 CKATN: RTS
02C6 201600 JSR
02C7 A00208 LDA
02C8 2904 AND #200000100
02C9 F0F6 BEQ SEND01:
02CA 68 PLA
02CB 49FF EOR SEND01:
02CC 800008 STA SEND01:
02CD A900 LDA #200000000
02CE 800208 STA #200000000
02CF 60 RTS

;
02D0 48 MAIN:
02D1 20900C JSR
02D2 8508 STA
02D3 4C0600 JMP

;
02D4 48 CKATN: RTS
02D5 201600 JSR
02D6 A00208 LDA
02D7 2904 AND #200000100
02D8 F0F6 BEQ SEND01:
02D9 68 PLA
02DA 49FF EOR SEND01:
02DB 800008 STA SEND01:
02DC A900 LDA #200000000
02DD 800208 STA #200000000
02DE 60 RTS

;
02DF 48 MAIN:
02E0 20900C JSR
02E1 8508 STA
02E2 4C0600 JMP

;
02E3 48 CKATN: RTS
02E4 201600 JSR
02E5 A00208 LDA
02E6 2904 AND #200000100
02E7 F0F6 BEQ SEND01:
02E8 68 PLA
02E9 49FF EOR SEND01:
02EA 800008 STA SEND01:
02EB A900 LDA #200000000
02EC 800208 STA #200000000
02ED 60 RTS

;
02EE 48 MAIN:
02EF 20900C JSR
02F0 8508 STA
02F1 4C0600 JMP

;
02F2 48 CKATN: RTS
02F3 201600 JSR
02F4 A00208 LDA
02F5 2904 AND #200000100
02F6 F0F6 BEQ SEND01:
02F7 68 PLA
02F8 49FF EOR SEND01:
02F9 800008 STA SEND01:
02FA A900 LDA #200000000
02FB 800208 STA #200000000
02FC 60 RTS

;
02FD 48 MAIN:
02FE 20900C JSR
02FF 8508 STA
0300 4C0600 JMP

;
0301 48 CKATN: RTS
0302 201600 JSR
0303 A00208 LDA
0304 2904 AND #200000100
0305 F0F6 BEQ SEND01:
0306 68 PLA
0307 49FF EOR SEND01:
0308 800008 STA SEND01:
0309 A900 LDA #200000000
030A 800208 STA #200000000
030B 60 RTS

;
030C 48 MAIN:
030D 20900C JSR
030E 8508 STA
030F 4C0600 JMP

;
0310 48 CKATN: RTS
0311 201600 JSR
0312 A00208 LDA
0313 2904 AND #200000100
0314 F0F6 BEQ SEND01:
0315 68 PLA
0316 49FF EOR SEND01:
0317 800008 STA SEND01:
0318 A900 LDA #200000000
0319 800208 STA #200000000
031A 60 RTS

;
031B 48 MAIN:
031C 20900C JSR
031D 8508 STA
031E 4C0600 JMP

;
031F 48 CKATN: RTS
0320 201600 JSR
0321 A00208 LDA
0322 2904 AND #200000100
0323 F0F6 BEQ SEND01:
0324 68 PLA
0325 49FF EOR SEND01:
0326 800008 STA SEND01:
0327 A900 LDA #200000000
0328 800208 STA #200000000
0329 60 RTS

;
032A 48 MAIN:
032B 20900C JSR
032C 8508 STA
032D 4C0600 JMP

;
032E 48 CKATN: RTS
032F 201600 JSR
0330 A00208 LDA
0331 2904 AND #200000100
0332 F0F6 BEQ SEND01:
0333 68 PLA
0334 49FF EOR SEND01:
0335 800008 STA SEND01:
0336 A900 LDA #200000000
0337 800208 STA #200000000
0338 60 RTS

;
0339 48 MAIN:
033A 20900C JSR
033B 8508 STA
033C 4C0600 JMP

;
033D 48 CKATN: RTS
033E 201600 JSR
033F A00208 LDA
0340 2904 AND #200000100
0341 F0F6 BEQ SEND01:
0342 68 PLA
0343 49FF EOR SEND01:
0344 800008 STA SEND01:
0345 A900 LDA #200000000
0346 800208 STA #200000000
0347 60 RTS

;
0348 48 MAIN:
0349 20900C JSR
034A 8508 STA
034B 4C0600 JMP

;
034C 48 CKATN: RTS
034D 201600 JSR
034E A00208 LDA
034F 2904 AND #200000100
0350 F0F6 BEQ SEND01:
0351 68 PLA
0352 49FF EOR SEND01:
0353 800008 STA SEND01:
0354 A900 LDA #200000000
0355 800208 STA #200000000
0356 60 RTS

;
0357 48 MAIN:
0358 20900C JSR
0359 8508 STA
035A 4C0600 JMP

;
035B 48 CKATN: RTS
035C 201600 JSR
035D A00208 LDA
035E 2904 AND #200000100
035F F0F6 BEQ SEND01:
0360 68 PLA
0361 49FF EOR SEND01:
0362 800008 STA SEND01:
0363 A900 LDA #200000000
0364 800208 STA #200000000
0365 60 RTS

;
0366 48 MAIN:
0367 20900C JSR
0368 8508 STA
0369 4C0600 JMP

;
036A 48 CKATN: RTS
036B 201600 JSR
036C A00208 LDA
036D 2904 AND #200000100
036E F0F6 BEQ SEND01:
036F 68 PLA
0370 49FF EOR SEND01:
0371 800008 STA SEND01:
0372 A900 LDA #200000000
0373 800208 STA #200000000
0374 60 RTS

;
0375 48 MAIN:
0376 20900C JSR
0377 8508 STA
0378 4C0600 JMP

;
0379 48 CKATN: RTS
037A 201600 JSR
037B A00208 LDA
037C 2904 AND #200000100
037D F0F6 BEQ SEND01:
037E 68 PLA
037F 49FF EOR SEND01:
0380 800008 STA SEND01:
0381 A900 LDA #200000000
0382 800208 STA #200000000
0383 60 RTS

;
0384 48 MAIN:
0385 20900C JSR
0386 8508 STA
0387 4C0600 JMP

;
0388 48 CKATN: RTS
0389 201600 JSR
038A A00208 LDA
038B 2904 AND #200000100
038C F0F6 BEQ SEND01:
038D 68 PLA
038E 49FF EOR SEND01:
038F 800008 STA SEND01:
0390 A900 LDA #200000000
0391 800208 STA #200000000
0392 60 RTS

;
0393 48 MAIN:
0394 20900C JSR
0395 8508 STA
0396 4C0600 JMP

;
0397 48 CKATN: RTS
0398 201600 JSR
0399 A00208 LDA
039A 2904 AND #200000100
039B F0F6 BEQ SEND01:
039C 68 PLA
039D 49FF EOR SEND01:
039E 800008 STA SEND01:
039F A900 LDA #200000000
03A0 800208 STA #200000000
03A1 60 RTS

;
03A2 48 MAIN:
03A3 20900C JSR
03A4 8508 STA
03A5 4C0600 JMP

;
03A6 48 CKATN: RTS
03A7 201600 JSR
03A8 A00208 LDA
03A9 2904 AND #200000100
03AA F0F6 BEQ SEND01:
03AB 68 PLA
03AC 49FF EOR SEND01:
03AD 800008 STA SEND01:
03AE A900 LDA #200000000
03AF 800208 STA #200000000
03B0 60 RTS

;
03B1 48 MAIN:
03B2 20900C JSR
03B3 8508 STA
03B4 4C0600 JMP

;
03B5 48 CKATN: RTS
03B6 201600 JSR
03B7 A00208 LDA
03B8 2904 AND #200000100
03B9 F0F6 BEQ SEND01:
03BA 68 PLA
03BB 49FF EOR SEND01:
03BC 800008 STA SEND01:
03BD A900 LDA #200000000
03BE 800208 STA #200000000
03BF 60 RTS

;
03C0 48 MAIN:
03C1 20900C JSR
03C2 8508 STA
03C3 4C0600 JMP

;
03C4 48 CKATN: RTS
03C5 201600 JSR
03C6 A00208 LDA
03C7 2904 AND #200000100
03C8 F0F6 BEQ SEND01:
03C9 68 PLA
03CA 49FF EOR SEND01:
03CB 800008 STA SEND01:
03CC A900 LDA #200000000
03CD 800208 STA #200000000
03CE 60 RTS

;
03CF 48 MAIN:
03D0 20900C JSR
03D1 8508 STA
03D2 4C0600 JMP

;
03D3 48 CKATN: RTS
03D4 201600 JSR
03D5 A00208 LDA
03D6 2904 AND #200000100
03D7 F0F6 BEQ SEND01:
03D8 68 PLA
03D9 49FF EOR SEND01:
03DA 800008 STA SEND01:
03DB A900 LDA #200000000
03DC 800208 STA #200000000
03DD 60 RTS

;
03DE 48 MAIN:
03DF 20900C JSR
03E0 8508 STA
03E1 4C0600 JMP

;
03E2 48 CKATN: RTS
03E3 201600 JSR
03E4 A00208 LDA
03E5 2904 AND #200000100
03E6 F0F6 BEQ SEND01:
03E7 68 PLA
03E8 49FF EOR SEND01:
03E9 800008 STA SEND01:
03EA A900 LDA #200000000
03EB 800208 STA #200000000
03EC 60 RTS

;
03ED 48 MAIN:
03EE 20900C JSR
03EF 8508 STA
03F0 4C0600 JMP

;
03F1 48 CKATN: RTS
03F2 201600 JSR
03F3 A00208 LDA
03F4 2904 AND #200000100
03F5 F0F6 BEQ SEND01:
03F6 68 PLA
03F7 49FF EOR SEND01:
03F8 800008 STA SEND01:
03F9 A900 LDA #200000000
03FA 800208 STA #200000000
03FB 60 RTS

;
03FC 48 MAIN:
03FD 20900C JSR
03FE 8508 STA
03FF 4C0600 JMP

;
0400 48 CKATN: RTS
0401 201600 JSR
0402 A00208 LDA
0403 2904 AND #200000100
0404 F0F6 BEQ SEND01:
0405 68 PLA
0406 49FF EOR SEND01:
0407 800008 STA SEND01:
0408 A900 LDA #200000000
0409 800208 STA #200000000
040A 60 RTS

;
040B 48 MAIN:
040C 2090
```



```

00B7 C920      CMP    #20      ;FALLS BLANK
00B9 F0EF      BEQ    LM3      ;ICORIEREN
00BB C92C      CMP    #2C      ;AUCH KOMMA
00BD F0EB      BEQ    LM3
00BF 2901      AND    #20000001 ;NUR 0,1 ERLAUBT
00C1 D00D      BNE    ORN      ;WENN = 1 DANN ODERN
00C3 98        TYA
00C4 49FF      EOR    #11111111 ;HOLE WERT
00C6 3510      AND    VAL1,X    ;COMPLEMENTIEREN
00C8 9510      STA    VAL1,X    ;IN X IST 0..3
00CA 20750C    JSR    BITSET    ;NEUER WERT
00CD 4C300D    JMP    MAIN      ;UND BELEGEN
0000           ORN:
0000 98        TYA
0001 1510      ORA    VAL1,X    ;HIER ODERN NOETIG
0003 9510      STA    VAL1,X    ;NEUER WERT
0005 20750C    JSR    BITSET
0008 4C300D    JMP    MAIN
;
;
0000           END

```

---- SYMBOL TABLE ----

B1	0C6D	LPPB	0C63	SKM1	0DA9
BITSET	0C75	MAIN	0D30	TALKOF	0CDB
BYTA	0C5D	MAIN1	0D35	TALKON	0CBF
CKATN	0D16	MODE	0D0E	TALPA	0D4A
CNT	0D0F	NULA	0C35	TIM1	0B14
CONSE	0D06	ORM	0D00	TIM64	0B16
COUNT	0D0C	PA	0D00	TIM8	0B15
COUV	0D14	PAD	0D01	TIMFLC	0B17
EINSA	0C48	PADR	0D0D	TIMIN	0B16
FLAG	0D0A	PB	0D02	TMP1	0B15
GETCHA	0C90	PBD	0D03	VAL1	0B10
INIA	0C25	RESET	0C00	VAL2	0B11
LISTPA	0D4D	S1	0C70	VAL3	0B12
LM1	0D6E	SEND	0CF1	VAL4	0B13
LM2	0D86	SEND1	0CF0	XTEMP1	0D00
LM3	0DAA	SENE01	0D17	ZC0U	0D09
LOPA	0C95	SENE01	0D18	ZEICH	0D0B
LOPB	0C81	SK2	0D66	ZOUT	0D08
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 Unterprogrammprung 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 #FFF, 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.

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			0C57	08	FHP	#SAVE Z FLAG
0000	FREQ	=10		0C58	20930C	JSR INCP	
0000	SPD1	=110		0C58	28	PLP	
0000	SPD2	=70		0C5C	D0F7	BNE STXT	
0000	PNT	*=*+2		0C5E	A214	LDX #20	#DELAY
0002	TIL	*=*+1		0C60	D002	BNE SPC	
0003	TIH	*=*+1		0C62	A202	LDX #2	#PAUSE
0004	CHR	*=\$FFC		0C64	18	CLC	#BETW. CHR'S
0FFC	000C	.WOR RES		0C65	209A0C	JSR ELEM	#Y=0
0FFE	PA	=\$800		0C68	B100	LDA (PNT),Y	
0FFE	PB	=\$802		0C6A	30A5	BMI WAIT	
0FFE	PBD	=\$803		0C6C	20930C	JSR INCP	
0FFE	CL64	=\$816		0C6F	38	SEC	
0FFE	CLK	=\$817		0C70	E920	SBC #\$20	
0FFE		*=\$C00		0C72	D004	BNE CONV	
0C00	RES	A2FF	LDX #\$FF	0C74	A205	LDX #5	#IT'S A
0C02	9A	TXS		0C76	D0EC	BNE SPC	#SPACE
0C03	78	SEI		0C78	CONV	AA	TAX
0C04	D8	CLD		0C79	BDC40C	LDA CTAB,X	
0C05	A9A0	LDA #A0	#PB7=AF	0C7C	8504	STA CHR	
0C07	8D0308	STA FBD	#PB5=TX	0C7E	SHFT	0604	ASL CHR
0C0A	A900	LDA #0		0C80	F0E0	BEQ NXT	
0C0C	8D0208	STA PB	#TX OFF	0C82	A201	LDX #1	
0C0F	F029	BEQ START		0C84	9002	BCC DOT	
0C11	WAIT	AD0008	LDA PA	0C86	A203	LDX #3	#DASH
0C14	2903	AND #3	#PA0-1	0C88	DOT	38	SEC
0C16	AA	TAX	#FOR REPEAT	0C89	209A0C	JSR ELEM	
0C17	BDC00C	LDA RPT,X	#TIME	0C8C	18	CLC	
0C1A	8503	STA TIH		0C8D	E8	INX	#1 ELEM
0C1C	A901	LDA #1		0C8E	209A0C	JSR ELEM	#PAUSE
0C1E	8D0208	STA PB	#TX OFF	0C91	F0EB	BEQ SHFT	#ALW.
0C21	8502	STA TIL		0C93		#INCREMENT POINTER	
0C23	WAIT0	A962	LDA #98	0C93	INCP	E600	INC PNT
0C25	8D1708	STA CLK		0C95	D002	BNE *+4	
0C28	WAIT1	2C1708	BIT CLK	0C97	E601	INC PNT+1	
0C2B	10FB	BPL WAIT1		0C99	60	RTS	
0C2D	2C0208	BIT PB		0C9A		#SEND X ELEMENTS WITH	
0C30	7008	BVS START		0C9A		#AF OUTPUT IF C=1	
0C32	C602	DEC TIL		0C9A	ELEM	A06E	LDY #SPD1
0C34	D0ED	BNE WAIT0		0C9C	A910	LDA #10	
0C36	C603	DEC TIH		0C9E	2D0208	AND PB	#MASK PB4
0C38	D0E9	BNE WAIT0		0CA1	F002	BEQ ELEM1	#FOR SPEED
0C3A	START	2C0208	BIT PB	0CA3	A046	LDY #SPD2	
0C3D	70FB	BVS START	#PB6KEY	0CA5	ELEM1	2C1708	BIT CLK
0C3F	A920	LDA #20	#TX ON	0CA8	10FB	BPL ELEM1	
0C41	8D0208	STA PB		0CAA	A90A	LDA #FREQ	
0C44	A904	LDA #<TXT	#SET PNT	0CAC	8D1608	STA CL64	
0C46	8500	STA PNT	#TO BEGIN	0CAF	9008	BCC QUIET	
0C48	A90D	LDA #>TXT	#OF TEXT	0CB1	AD0208	LDA PB	
0C4A	8501	STA PNT+1		0CB4	4980	EOR #80	
0C4C	AD0208	LDA PB	#PB0-3	0CB6	8D0208	STA PB	
0C4F	290F	AND #F	#TO SELECT	0CB7	QUIET	88	DEY
0C51	0980	ORA #80	#TEXT	0CBA	D0E9	BNE ELEM1	
0C53	A000	LDY #0		0CBC	CA	DEX	
0C55	STXT	D100	CMP (PNT),Y	0CBD	D0DB	BNE ELEM	

00BF	60	RTS	00F6	50	
00C0		REPEAT TIMES 0-3	00F7	10	
00C0 RPT	14	.BYT 20,5,2,1	00F8	C0	.BYT \$C0,\$30,\$18,\$70
00C1	05		00F9	30	
00C2	02		00FA	18	
00C3	01		00FB	70	
00C4		ASCII TO CW CONVERSION	00FC	98	.BYT \$98,\$88,\$C8,\$80
00C4 CTAB	80	.BYT \$80,\$36,\$4A,1	00FD	B8	
00C5	36		00FE	C8	
00C6	4A		00FF	80	
00C7	01		0D00	80	.BYT \$80,0,\$80,\$80
00C8	16	.BYT \$16,\$80,\$44,\$7A	0D01	00	
00C9	80		0D02	80	
00CA	44		0D03	80	
00CB	7A		0D04		
00CC	B6	.BYT \$B6,\$86,\$80,\$80	0D04		MORSE TEXT IN ASCII
00CD	B6		0D04		(EXAMPLE FOR DC1YB / DK8CD)
00CE	80		0D04 TXT	8F	.BYT \$8F,'DK8CD'
00CF	80		0D05	444B	
00D0	CE	.BYT \$CE,\$86,\$56,\$94	0D0A	8E	.BYT \$8E,'DK8CD/P', \$8D
00D1	86		0D0B	444B	
00D2	56		0D12	8D	
00D3	94		0D13	444B	.BYT 'DK8CD/M', \$8C
00D4	FC	.BYT \$FC,\$7C,\$3C,\$1C	0D1A	8C	
00D5	7C		0D1B	5445	.BYT 'TEST DE DK8CD'
00D6	3C		0D28	2049	.BYT ' IN MUENCHEN', \$8B
00D7	1C		0D34	8B	
00D8	0C	.BYT \$C,4,\$84,\$C4	0D35	5445	.BYT 'TEST DE DK8CD/P', \$8A
00D9	04		0D44	8A	
00DA	84		0D45	5445	.BYT 'TEST DE DK8CD/M', \$89
00DB	C4		0D54	89	
00DC	E4	.BYT \$E4,\$F4,\$56,\$56	0D55	444B	.BYT 'DK8CD IN ATV', \$88
00DD	F4		0D61	88	
00DE	56		0D62	5053	.BYT 'PSE QRX QRX DE DK8CD', \$87
00DF	56		0D76	87	
00E0	AC	.BYT \$AC,\$8C,\$54,\$32	0D77	4351	.BYT 'CQ CQ CQ DE DK8CD', \$86
00E1	8C		0D88	86	
00E2	54		0D89	5152	.BYT 'QRZ QRZ DE DK8CD', \$85
00E3	32		0D99	85	
00E4	4C	.BYT \$4C,\$60,\$88,\$A8	0D9A	4443	.BYT 'DC1YB', \$84
00E5	60		0D9F	84	
00E6	88		0DA0	4443	.BYT 'DC1YB/P', \$83
00E7	A8		0DA7	83	
00E8	90	.BYT \$90,\$40,\$28,\$D0	0DA8	5445	.BYT 'TEST DE DC1YB', \$82
00E9	40		0DB5	82	
00EA	28		0DB6	5445	.BYT 'TEST THE QUICK'
00EB	D0		0DC5	2042	.BYT ' BROWN FOX JUMPS'
00EC	08	.BYT \$8,\$20,\$7B,\$B0	0DD5	204F	.BYT ' OVER THE LAZY'
00ED	20		0DE3	2044	.BYT ' DOG 1234567890'
00EE	78		0DF2	3C21	.BYT '<!"#\$%&'()*+,-./:;>@'
00EF	B0		0E00	4443	.BYT 'DC1YB IN MUENCHEN', \$80
00F0	48	.BYT \$48,\$E0,\$A0,\$F0	0E11	80	
00F1	E0		0E12	454D	.BYT 'EMUF-CW V1.1'
00F2	A0		0E1E	2044	.BYT ' DC1YB 8/1981'
00F3	F0		0E2B	80	.BYT \$80
00F4	68	.BYT \$68,\$D8,\$50,\$10	0E2C		.END
00F5	D8		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	
17	PB 1	Textauswahl (0...15)
18	PB 2	
19	PB 3	
22	PB 7	Nf-Ausgang (U _{ss} = 5 V), Ruhelage Low Starttaste oder Schalter, aktiv Low Steuersignal-Ausgang, aktiv High Geschwindigkeit langsam/schnell + 5 V, verbinden mit 28
23	PB 6	
24	PB 5	
25	PB 4	
27	—	

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

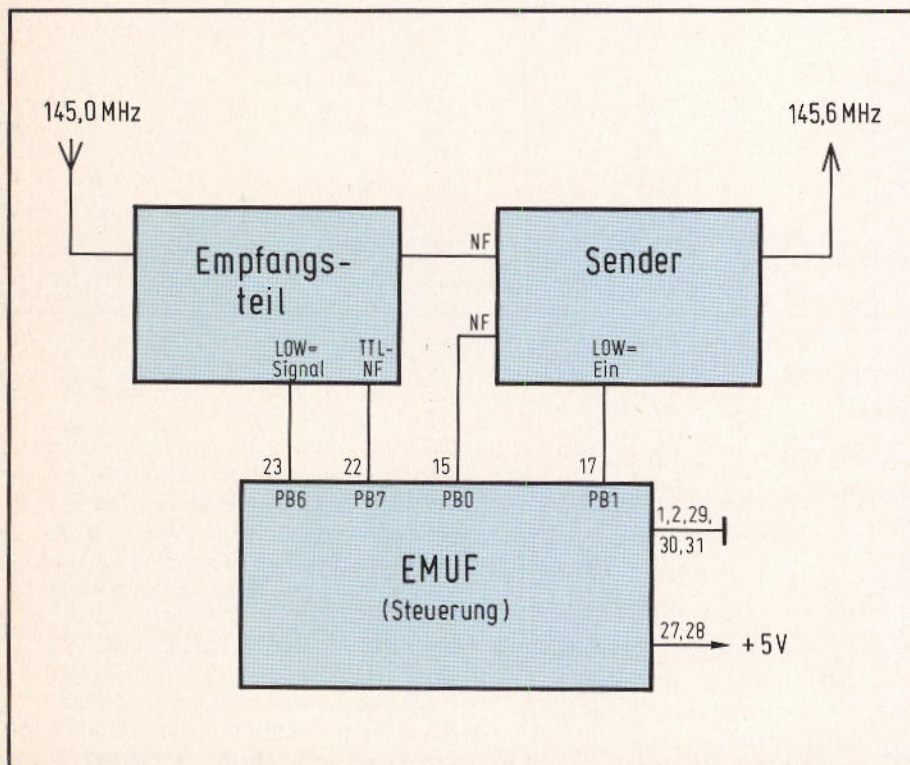


Bild 1. Der EMUF erhält ein Rauschsperr- und ein Nf-Signal vom Empfangsteil, schaltet den Sender ein und aus und erzeugt die Nf der Morse-Ausstrahlung

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 Rauschsperr 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 Abruf 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 Abruf des vorprogrammierten Textes; 0C73 = Tonruf-Auswertefrequenz (hier 1750 Hz); 0C99 = Tonruf-Auswerteemp-

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	CO
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	40	O	F0	Y	B8
6	84	F	28	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=QUELCH 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 PBD *= $FFC
0007 OFFC 000C .WOR RES
0007 OFFE *= $C00
0000 RES A2FF LDX $FFF
0002 9A TXS
0003 D8 CLD
0004 78 SEI
0005 A903 LDA £3
0007 8D0308 STA PBD
000A ;TX OFF
000A OFF ADO208 LDA PB
000D 0902 ORA £2
000F 8D0208 STA PB
0012 ;WAIT FOR 1750 HZ
0012 A264 LDX £100
0014 OFF1 20670C JSR AK
0017 10F1 BPL OFF
0019 CA DEX
001A DOF8 BNE OFF1
001C ;TX ON
001C ADO208 LDA PB
001F 29FD AND £$FD
0021 8D0208 STA PB
0024
0024 ;SEND CW CALL
0024 CA A200 LDX £0
0026 20A40C JSR CW
0029 A902 LDA £2
002B 8502 STA TIH
002D ON A9FE LDA £$FE
002F 8503 STA THLD
0031 ;COUNT DOWN TIMER
0031 ON3 C600 DEC TIL
0033 D020 BNE ON2
0035 C601 DEC TIM
0037 D004 BNE ON1
0039 C602 DEC TIH
003B FOE7 BEQ CA
003D ON1 2C0208 BIT PB
0040 50EB BVC ON
0042 E603 INC THLD
0044 FO08 BEQ BEEP
0046 A90A LDA £10
0048 C503 CMP THLD
004A 30BE BMI OFF
004C 1007 BPL ON2
004E ;SEND ROGER BEEP
004E BEEP
0050 8606 STX SPL2
0052 20AA0C JSR NXT
0055 ;TEXT REQUEST?
0055 ON2 A264 LDX £100
0057 REQ 20670C JSR AK
005A 10D5 BPL ON3
005C CA DEX
005D DOF8 BNE REQ
005F A20B LDX £TXT-CALL
0061 20A40C JSR CW
0064 4C310C JMP ON3
0067
0067 ;AUTOCORRELATION
0067 ;1750HZ, A/X/Y SAVED
0067 AK 48 PHA
0068 98 TYA
0069 48 PHA
006A 8A TXA
006B 48 PHA
006C A900 LDA £0
006E 8504 STA SUM
0070 A010 LDY £16
0072 AK1 A20A LDX £10
0074 AK2 CA DEX
0075 DOFD BNE AK2
0077 ADO208 LDA PB
007A 0A ASL A
007B 6605 ROR SPL1
007D 6606 ROR SPL2
007F 88 DEY
0080 DOFO BNE AK1
0082 A505 LDA SPL1
0084 FO10 BEQ AK5
0086 C9FF CMP £$FF
0088 FO0C BEQ AK5
008A 4506 EOR SPL2
008C A007 LDY £7
008E AK3 6A ROR A
008F B002 BCS AK4
0091 C604 DEC SUM
0093 AK4 88 DEY
0094 10F8 BPL AK3
0096 AK5 A504 LDA SUM
0098 6905 ADC £5
009A 8504 STA SUM
009C 68 PLA
009D AA TAX
009E 68 PLA
009F A8 TAY
00A0 68 PLA
00A1 2404 BIT SUM
00A3 RDY 60 RTS
00A4 ;SEND CW CODE
00A4 CW 8606 STX SPL2
00A6 A21E LDX £30
00A8 D002 BNE *+4
00AA A202 LDX £2
00AC 18 CLC
00AD 20D80C JSR ELEM
00AB A606 LDX SPL2
00B2 E606 INC SPL2
00B4 BDF30C LDA CALL,X
00B7 FOEA BEQ RDY
00B9 8505 STA SPL1
00BB C9FF CMP £$FF
00BD D004 BNE SHFT
00BF A207 LDX £7
00C1 DOE9 BNE SPC
00C3 SHFT 0605 ASL SPL1
00C5 FOE3 BEQ NXT
00C7 A201 LDX £1
00C9 9002 BCC DOT
00CB A203 LDX £3
00CD DOT 38 SEC
00CE 20D80C JSR ELEM
00D1 18 CLC
00D2 E8 INX
00D3 20D80C JSR ELEM
00D6 FOEB BEQ SHFT
00D8 ELEM A096 LDY £150
00DA ELEM1 A932 LDA £50
00DC 8504 STA SUM
00DE ELEM2 C604 DEC SUM
00E0 DOFC BNE ELEM2
00E2 9008 BCC QUIET
00E4 ADO208 LDA PB
00E7 4901 EOR £1
00E9 8D0208 STA PB
00EC QUIET 88 DEY
00ED DOEB BNE ELEM1
00EF CA DEX
00F0 DOE6 BNE ELEM
00F2 60 RTS
00F3 ;CW CALL 'DC1YB'
00F3 CALL 90 .BYT $90,$A8
00F4 A8
00F5 7C .BYT $7C,$B8
00F6 B8
00F7 88 .BYT $88,0
00F8 00
00F9 FF .BYT $FF,$FF,$FF
00FA FF
00FB FF
00FC ;ROGER BEEP 'K'
00FC K B0 .BYT $B0,0
00FD 00
00FE ;CW TEXT
00FE ;'TEST DE DC1YB'
00FE TXT CO .BYT $C0,$40
00FF 40
00D0 10 .BYT $10,$C0
00D1 CO
00D2 FF .BYT $FF,$90
00D3 90
00D4 40 .BYT $40,$FF
00D5 FF
00D6 90 .BYT $90,$A8
00D7 A8
00D8 7C .BYT $7C,$B8
00D9 B8
00DA 88 .BYT $88,0
00DB 00
00DC .END
00DC 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.

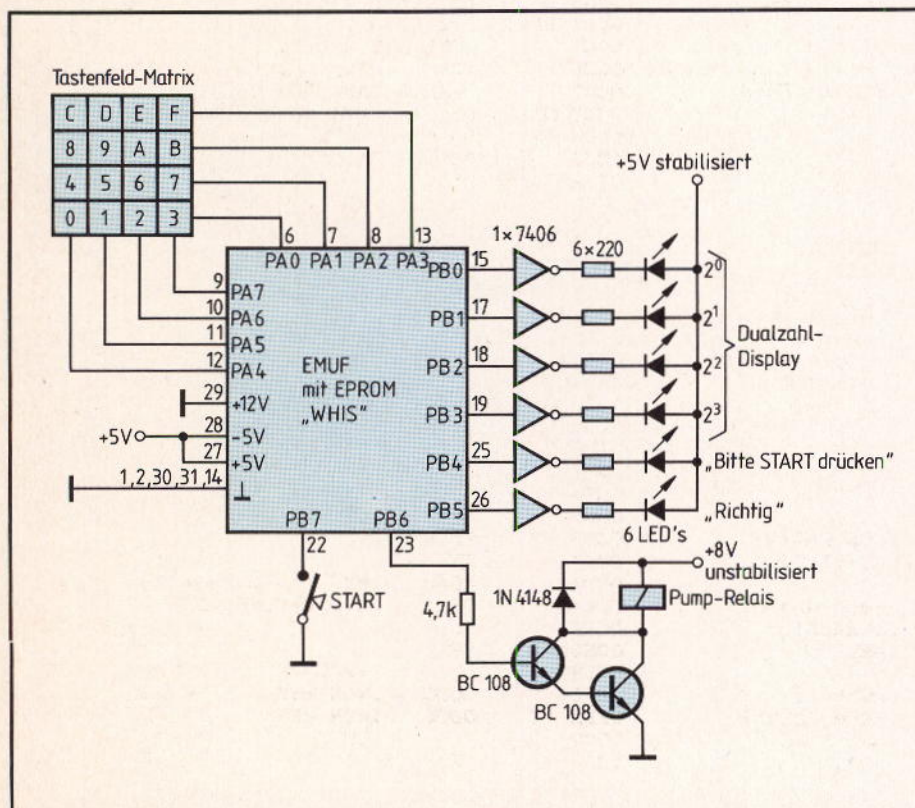


Bild 1. Schaltbild des Whisky-EMUF. Als Pumpe, die sinnvollerweise (wegen des hohen Strombedarfs) über ein kleines Relais angesteuert wird, eignen sich Ausführungen aus der Kfz-Scheibenwaschanlage oder aus dem Aquarium

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 „Trostpreis“ 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 ;MC-WHISKY-EMUF
0000 ;V1.1/FE/26.7.1981
0000 *=0
0000 TIME *=+1
0001 KEY *=+1
0002 RND *=+1
0003 TI *=+1
0000 PA =$800
0000 PAD =$801
0000 PB =$802
0000 PBD =$803
0000 CLK =$817
0000 RES A90F LDA £$F ;INIT
0002 8D0108 STA PAD ;PORTS
0005 A97F LDA £$7F
0007 8D0308 STA PBD
000A A2FF LDX £$FF ;AND STACK
000C 9A TXS
000D A946 LDA £70 ;DEFAULT
000F 8503 STA TI ;TIMING
0011 D8 CLD
0012 78 SEI
0013 WAIT E8 INX ;X=RANDOM
0014 2C1708 BIT CLK
0017 100F BPL WAIT1
0019 A503 LDA TI
001B 8D1708 STA CLK
001E AD0208 LDA PB
0021 2910 AND £$10
0023 4910 EOR £$10 ;FLASH
0025 8D0208 STA PB ;START LED
0028 WAIT1 2C0208 BIT PB ;START?
002B 30E6 BMI WAIT
002D 8A TXA ;YES
002E 290F AND £$F
0030 8502 STA RND ;DISPLAY NR
0032 8D0208 STA PB
0035 A900 LDA £0 ;START TIMER
0037 8500 STA TIME
0039 LP 20BC0C JSR DLY
003C E600 INC TIME
003E 30D3 BMI WAIT
0040 20980C JSR SCAN
0043 DOF4 BNE LP
0045 A501 LDA KEY ;CORRECT
0047 C502 CMP RND ;KEY?
0049 D03B BNE ERR
004B 0920 ORA £$20 ;YES,
004D 8D0208 STA PB ;LED ON
0050 A500 LDA TIME
0052 A602 LDX RND
0054 18 CLC
0055 7DC70C ADC MAL,X
0058 C91E CMP £30
005A B02A BCS ERR ;NO WHISKY
005C OK C91C CMP £28
005E B00A BCS HALF
0060 A503 LDA TI ;TOO EASY?!
0062 E905 SBC £5 ;ADAPT TI
0064 8503 STA TI
0066 A214 LDX £20 ;FULL GLASS,
0068 D002 BNE PMP ;CHEERS!
006A HALF A20A LDX £10 ;HALF
006C PMP AD0208 LDA PB
006F 0940 ORA £$40 ;PUMP IT
0071 8D0208 STA PB ;FROM BOTTLE
0074 PMP1 A962 LDA £98
0076 20BEOC JSR DLYO
0079 CA DEX

007A DOF8 BNE PMP1
007C AD0208 LDA PB
007F 292F AND £$2F ;PUMP OFF
0081 8D0208 STA PB
0084 D002 BNE PAU
0086 ERR E603 INC TI ;NOT EASY
0088 PAU A23C LDX £60 ;WAIT
008A PAU1 20BC0C JSR DLY ;A FEW
008D CA DEX ;SECONDS
008E DOFA BNE PAU1 ;LOOP IF
0090 2C0208 BIT PB ;START KEY
0093 10F3 BPL PAU ;STILL DOWN
0095 4C130C JMP WAIT
0098 ;SCAN KEYBOARD
0098 ;HEX VALUE IN KEY,
0098 ;Z=1 IF KEY PRESSED
0098 SCAN A900 LDA £0
009A 8501 STA KEY
009C A203 LDX £3
009E A90E LDA £$E
00A0 8D0008 STA PA
00A3 SCAN1 A004 LDY £4
00A5 A910 LDA £$10
00A7 SCAN2 2C0008 BIT PA
00AA FOOF BEQ FND
00AC E601 INC KEY
00AE OA ASL A
00AF 88 DEY
00B0 DOF5 BNE SCAN2
00B2 OEO008 ASL PA
00B5 EEO008 INC PA
00B8 CA DEX
00B9 10E8 BPL SCAN1
00BB FND 60 RTS
00BC ;DELAY TI
00BC DLY A503 LDA TI
00BE DLYO 8D1708 STA CLK
00C1 DLY1 2C1708 BIT CLK
00C4 10FB BPL DLY1
00C6 60 RTS
00C7 ;MALUS TAB. FOR O-F
00C7 MAL 05 .BYT 5,10,8,5
00C8 OA
00C9 08
00CA 05
00CB 08 .BYT 8,0,0,6
00CC 00
00CD 00
00CE 06
00CF 08 .BYT 8,2,0,0
00D0 02
00D1 00
00D2 00
00D3 02 .BYT 2,0,2,5
00D4 00
00D5 02
00D6 05
00D7 ;THAT'S ALL!
00D7 FF .BYT $FF,$FF,$FF
00D8 FF
00D9 FF
00DA 4D43 .BYT 'MC-EMUF/FE'
00E4 FF .BYT $FF,$FF,$FF
00E5 FF
00E6 FF
00E7 *=FFFC
00FC .WOR RES ;VECTOR
00FE .WOR RES
1000 .END
1000 ERRORS= 0000

```


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 900 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-

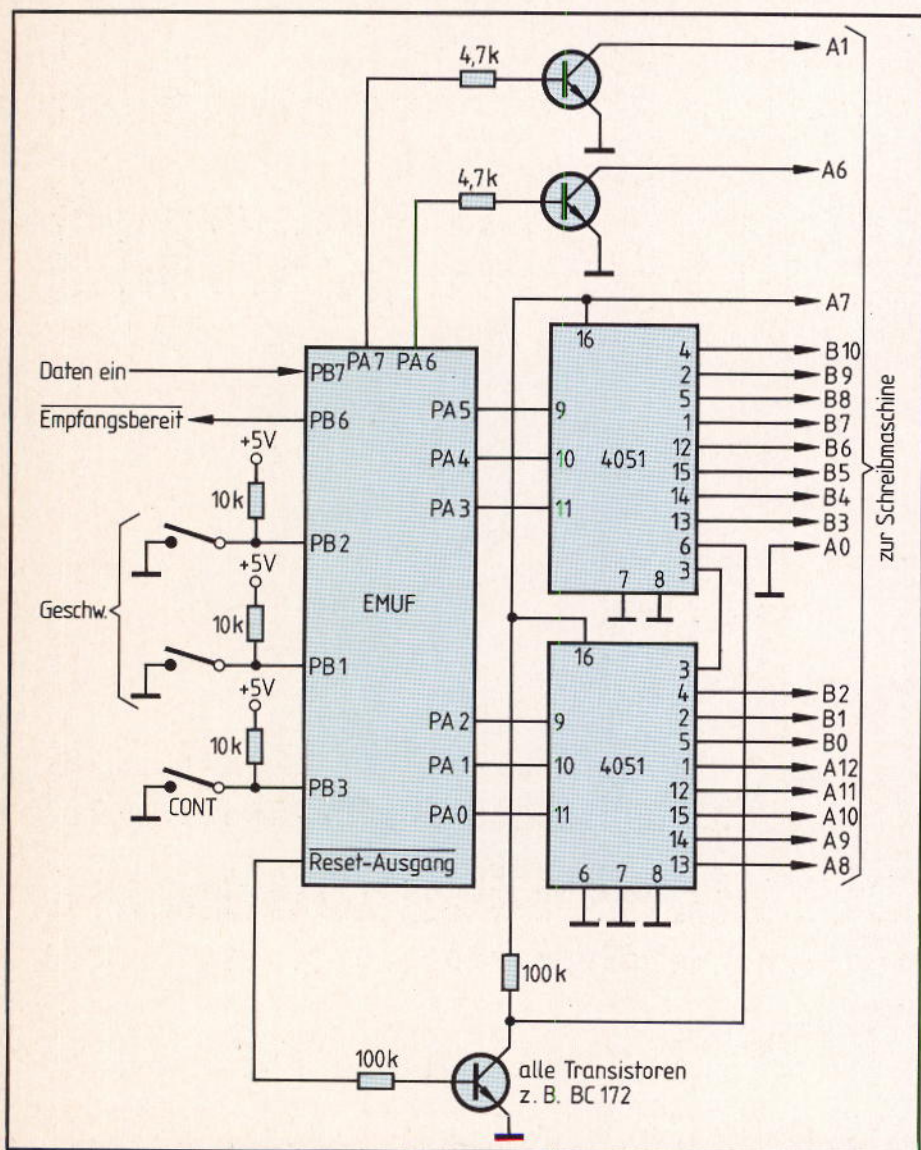


Bild 1. So wird die Tastaturmatrix der Olivetti-Schreibmaschine P30 mit der EMUF-Platine verbunden

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

0800	**END OF PASS 1	1	;	V24-INTERFACE FUER PRIVILEG	OC00	57	TSD	EQU \$10	
0800	**END OF PASS 2	2	;	*****	OC00	58	TBD	EQU \$30	; TABULATOR
0800		3	;	COPYRIGHT:	OC00	59	;	*****	*****
0800		4	;	RUDOLF HOFER	OC00	60	;	INITIALISIERUNG	
0800		5	;	FRUNHOFERSTR.22	OC00	61	INIT	LDX #FFF	
0800		6	;	8260 MUEHLDOERF/INN	OC02	62		TXS	
0800		7	;	*****	OC03	63		CLD	
0800		8	;	PB1/2 GESCHW. FUER V24	OC04	64		SEI	
0800		9	;	0 0 = 110 BD	OC05	65		STX DRA	
0800		10	;	1 0 = 300 BD	OC08	66		LDA #S0C	
0800		11	;	0 1 = 600 BD	OC0A	67		STA PA	
0800		12	;	1 1 = 1200 BD	OC0D	68		LDA #40	
0800		13	;	PB3 = ON LINE	OC0F	69		STA DRB	
0800		14	;	PB6=EMPANGSBEREIT (0)	OC12	70		LDA #0	; PB6=
0800		15	;	PB7=DATENEINGANG	OC14	71		STA NEUD	; AUSG. (BUSY)
0800		16	;	*****	OC16	72		STA DIRFL	
0800		17	;	PA = AUSGANG FUER SCHREIBMASCHINE	OC18	73		LDX #20	
0800		18	;	*****	OC1A	74	NUL	STA NDEF, X	; KEINE
0800		19	;	*****	OC1C	75		DEX	
0800		20	;	RESET-VEKTOR FFC:00	OC1D	76		BNE NUL	
0800		21	;	FFD:0C	OC1F	77		STA CHRCT	
0800		22	;	ORG \$FFC	OC21	78		STA LINCT	
0800		23	;	HEX 000C	OC23	79		LDA #60	
0800		24	;	ORG \$C00	OC25	80		STA LMAX	
0800		25	;	*****	OC27	81		LDA #60	
0800		26	;	PA EQU \$800	OC29	82		STA CHMAX	
0800		27	;	PB EQU \$802	OC2B	83		LDX #11	
0800		28	;	DRA EQU \$801	OC2D	84	NUL1	LDA ITAB, X	
0800		29	;	DRB EQU \$803	OC30	85		STA TABUL, X	
0800		30	;	WTIMER EQU \$814	OC32	86		DEX	
0800		31	;	RTIMER EQU \$815	OC33	87		BPL NUL1	
0800		32	;	FLAG EPZ 0	OC35	88		LDA #CRD	
0800		33	;	XTEMP EPZ 1	OC37	89		JSR ZAUS	
0800		34	;	ZEIT1 EPZ 2	OC3A	90	LOOP	JSR ZEIN	
0800		35	;	ZEIT2 EPZ 3	OC3D	91		JSR ZAUS	
0800		36	;	DIRFL EPZ 4	OC40	92		JMP LOOP	
0800		37	;	LMAX EPZ 4	OC43	93	;	*****	*****
0800		38	;	NDEF EPZ 4	OC43	94	KTAB	HEX 414243476A7172767702076A	
0800		39	;	CHMAX EPZ \$19	OC46	95			
0800		40	;	LINCT EPZ \$1A	OC49	96		LDX #0	
0800		41	;	CHRCT EPZ \$1B	OC4F	97	KFILT	PHA	
0800		42	;	YTEMP EPZ \$1C	OC52	98	VERGL	AND #7F	
0800		43	;	XTEM1 EPZ \$1E	OC54	99		CMP KTAB, X	
0800		44	;	YTEM1 EPZ \$1F	OC57	100		BEQ FOUND	
0800		45	;	NEUD EPZ \$20	OC59	101		INX	
0800		46	;	KFLAG EPZ \$21	OC5A	102		CPX #12	
0800		47	;	LFD EPZ \$22	OC5C	103		BNE VERGL	
0800		48	;	TABUL EPZ \$22	OC5E	104		CLC	
0800		49	;	CRD EQU \$06	OC5F	105			
0800		50	;	DC1 EQU \$11	OC60	106		PLA	
0800		51	;	DC2 EQU \$12	OC61	107	FOUND	RTS	
0800		52	;	PRESS EQU \$140	OC62	108		SEC	
0800		53	;	PAUSE EQU \$173	OC63	109		PLA	
0800		54	;	CLTD EQU \$0B	OC64	110		RTS	
0800		55	;	REPTD EQU \$04	OC67	111		JMP TABR	
0800		56	;		OC67	111		JMP TABR	

OC6A 20400E	112	CMA	JSR ZEIN	171	OCDE 204A0D	JSR CODE	;	TABELLENWERT
OC6D 851A	113	STA CHMAX	RTS	172	OCE1 C906	CMP #LFD	;	LINE FEED?
OC6F 60	114	RTS	JSR ZEIN	173	OCE3 F036	BEQ #NFD	;	CARRIAGE RET.?
OC70 20400E	115	LMAXI	DCM "PR#2"	174	OCE5 C900	CMP #CRD	;	Z. DOPPELT BELEGT?
OC73 8519	116	STA LMAX	RTS	175	OCE7 F045	BEQ CRRTN	;	SPRINGE, WENN JA
OC75 60	117	RTS	JSR ZEIN	176	OCE9 48	PHA		
OC76 20400E	118	DEFI	LX NEUD	177	OCEA 204F0C	JSR KFILT		
OC79 A620	119	CPX #10	BEQ ERR	178	OCEB B0C3	BCS SAME		
OC7B E00A	120	BEQ ERR	STA NDEF+1, X	179	OCEC 68	PLA		
OC7F 9505	121	STA NDEF+1, X	INC NEUD	180	OCEQ 20330D	JSR OUT		
OC81 20400E	122	STA NDEF+11, X	RTS	181	OCE3 E61C	INC CHRCT		
OC84 950F	123	INC NEUD	RTS	182	OCE5 A51C	LDA CHRCT		
OC86 E620	124	RTS	JMP ERROR	183	OCE7 C51A	CMP CHMAX		
OC88 60	125	ERR	DIRKT LDA #1	184	OCE9 F00F	BEQ NZEIL		
OC89 4C050E	126	DIRKT	STA DIRFL	185	OCEB A51B	LDA LINCT		
OC8C A901	127	RTS	RTS	186	OCEB C519	CMP LMAX		
OC8E 8504	128	RTS	RTS	187	OCEB F06E	BEQ ONLIN		
OC90 60	129	RTS	RTS	188	OD01 A900	LDA #0		
OC91 20400E	130	RTS	RTS	189	OD03 8504	STA DIRFL		
OC94 C941	131	RTS	RTS	190	OD05 A601	LX XTEMP		
OC96 F0D2	132	RTS	RTS	191	OD07 A41D	LDY YTEMP		
OC98 C942	133	RTS	RTS	192	OD09 60	RTS		
OC9A F0DA	134	RTS	RTS	193	OD0A	NZEIL		
OC9C C943	135	RTS	RTS	194	OD0A A906	LDA #LFD		
OC9E F0D0	136	RTS	RTS	195	OD0C 20330D	JSR OUT		
OC9F F0D0	137	RTS	RTS	196	OD0F A900	LDA #0		
OC9F F0D0	138	RTS	RTS	197	OD11 851C	STA CHRCT		
OC9F F0D0	139	RTS	RTS	198	OD13 207D0D	JSR CRTIM		
OC9F F0D0	140	RTS	RTS	199	OD16 E61B	INC LINCT		
OC9F F0D0	141	RTS	RTS	200	OD18 4CFB0C	JMP WTR		
OC9F F0D0	142	RTS	RTS	201	OD1B	INC LINCT		
OC9F F0D0	143	RTS	RTS	202	OD1B E61B	LDA #LFD		
OC9F F0D0	144	RTS	RTS	203	OD1D A906	JSR OUT		
OC9F F0D0	145	RTS	RTS	204	OD1F 20330D	LDY CHRCT		
OC9F F0D0	146	RTS	RTS	205	OD22 A41C	JSR CRTIM		
OC9F F0D0	147	RTS	RTS	206	OD24 207D0D	LDA #0		
OC9F F0D0	148	RTS	RTS	207	OD27 A900	STA CHRCT		
OC9F F0D0	149	RTS	RTS	208	OD29 851C	JMP WEI		
OC9F F0D0	150	RTS	RTS	209	OD2B 4CF50C	CRRTN		
OC9F F0D0	151	RTS	RTS	210	OD2E	LDA #CRD		
OC9F F0D0	152	RTS	RTS	211	OD2E A900	JMP CRRT		
OC9F F0D0	153	RTS	RTS	212	OD30 4C1F0D	OUT		
OC9F F0D0	154	RTS	RTS	213	OD33	ORA KFLAG		
OC9F F0D0	155	RTS	RTS	214	OD33 0521	STA PA		
OC9F F0D0	156	RTS	RTS	215	OD35 8D0008	LDA #PRESS		
OC9F F0D0	157	RTS	RTS	216	OD38 A9BC	JSR DELAY		
OC9F F0D0	158	RTS	RTS	217	OD3A 20630D	LDA #0C		
OC9F F0D0	159	RTS	RTS	218	OD3D A90C	ORA KFLAG		
OC9F F0D0	160	RTS	RTS	219	OD3F 0521	STA PA		
OC9F F0D0	161	RTS	RTS	220	OD41 8D0008	LDA #PAUSE		
OC9F F0D0	162	RTS	RTS	221	OD44 A9AD	JSR DELAY		
OC9F F0D0	163	RTS	RTS	222	OD46 20630D	RTS		
OC9F F0D0	164	RTS	RTS	223	OD49 60	AND #7F		
OC9F F0D0	165	RTS	RTS	224	OD4A	TAY		
OC9F F0D0	166	RTS	RTS	225	OD4A 297F	LDX #10		
OC9F F0D0	167	RTS	RTS	226	OD4C AB	CMP NDEF, X		
OC9F F0D0	168	RTS	RTS	227	OD4D A20A	BEQ NDEF		
OC9F F0D0	169	RTS	RTS	228	OD4F D504	NCHST		
OC9F F0D0	170	RTS	RTS	229	OD51 F007	BEQ NDEF		


```

OD53 CA      230      DEX      BNE NCHST      ; NORMAL
OD54 DOF9    231      LDA TAB, Y
OD56 B9A90E  232      RTS
OD59 60      233      NDEF+10, X      ; DRUCKERCODE LADEN
OD5A B50E    234      LDA NDEF+10, X
OD5C 60      235      RTS
OD5D         236      ; =====
OD5D A00E    237      LDY #14      ; 0,5 SEKUNDEN
OD5F 207E0D  238      KBDL      JSR CTIM
OD62 60      239      RTS
OD63         240      ; =====
OD63 38      241      DELAY SEC
OD64 48      242      LO PHA
OD65 E901    243      SBC #1
OD67 DOFC    244      LOP
OD69 68      245      PLA
OD6A E901    246      SBC #1
OD6C DOF6    247      BNE LO
OD6E 60      248      RTS
OD6F         249      ; =====
OD6F AD0208  250      ONLIN LDA PB      ; PB3=ON-LINE
OD72 2908    251      AND #200001000    ; CONT., WENN NULL
OD74 DOF9    252      BNE ONLIN
OD76 A900    253      LDA #0
OD78 851B    254      STA LINCT
OD7A 4C010D  255      JMP CONTI
OD7D         256      ; =====
OD7D CB      257      CRTIM INY      ; Y ENTH. N
OD7E 20850D  258      CTIM JSR DELY
OD81 88      259      DEY
OD82 DOFA    260      BNE CTIM
OD84 60      261      RTS
OD85         262      ; =====
OD85 A978    263      DELY LDA #120      ; 36 MS
OD87 20630D  264      JSR DELAY
OD8A 60      265      RTS
OD8B         266      ; =====
OD8B A200    267      ; TABULATOR EINSTELLEN
OD8B A00C    268      TABR LDY #0
OD8B A00C    269      LDY #12
OD8F 20400E  270      ZE1 JSR ZEIN
OD92 C51A    271      CMP CHMAX
OD94 106F    272      BPL ERROR
OD96 C900    273      CMP #0
OD98 F04A    274      BEQ REST
OD9A 9522    275      STA TABUL, X
OD9C E8      276      INX
OD9D 88      277      DEY
OD9E DOEF    278      BNE ZE1
OD9E A900    279      LDA #CRD
ODA2 20E10C  280      JSR DIR
ODA5 A920    281      LDA #20
ODA7 20CA0C  282      JSR ZAUS
ODAA A908    283      LDA #CLTD
ODAC 20330D  284      JSR OUT
ODAF A904    285      LDA #REPTD
ODB1 20330D  286      JSR OUT
ODB4 20EC0D  287      JSR CHECK
ODB7 A200    288      LDX #0

ODB9 A00C    289      ODB9 A00C
ODBB B522    290      ODBB B522
ODBD DO06    291      ODBD DO06
ODBF A900    292      ODBF A900
ODC1 20E10C  293      ODC1 20E10C
ODC4 60      294      ODC4 60
ODC5 C51C    295      ODC5 C51C
ODC7 DO11    296      ODC7 DO11
ODC9 A910    297      ODC9 A910
ODCB 20330D  298      ODCB 20330D
ODCE A920    299      ODCE A920
ODD0 20CA0C  300      ODD0 20CA0C
ODD3 E8      301      ODD3 E8
ODD4 88      302      ODD4 88
ODD5 DOE4    303      ODD5 DOE4
ODD7 4CBF0D  304      ODD7 4CBF0D
ODDA 48      305      ODDA 48
ODDB A920    306      ODDB A920
ODDD 20CA0C  307      ODDD 20CA0C
ODE0 68      308      ODE0 68
ODE1 4CC50D  309      ODE1 4CC50D
ODE4         310      ODE4
ODE4 9522    311      ODE4 9522
ODE6 E8      312      ODE6 E8
ODE7 88      313      ODE7 88
ODE8 DOFA    314      ODE8 DOFA
ODEA F0B4    315      ODEA F0B4
ODEC         316      ODEC
ODEC A200    317      ODEC A200
ODEE A00C    318      ODEE A00C
ODFO B522    319      ODFO B522
ODF2 F011    320      ODF2 F011
ODF4 E8      321      ODF4 E8
ODF5 88      322      ODF5 88
ODF6 B522    323      ODF6 B522
ODF8 F00A    324      ODF8 F00A
ODFA D521    325      ODFA D521
ODFC 3007    326      ODFC 3007
ODFE F005    327      ODFE F005
OE00 E8      328      OE00 E8
OE01 88      329      OE01 88
OE02 DOF2    330      OE02 DOF2
OE04 60      331      OE04 60
OE05         332      OE05
OE05 68      333      OE05 68
OE06 68      334      OE06 68
OE07 4C3A0C  335      OE07 4C3A0C
OE0A         336      OE0A
OE0A A200    337      OE0A A200
OE0C A00C    338      OE0C A00C
OE0E B522    339      OE0E B522
OE10 F025    340      OE10 F025
OE12 C51A    341      OE12 C51A
OE14 1021    342      OE14 1021
OE16 C51C    343      OE16 C51C
OE18 F002    344      OE18 F002
OE1A 1004    345      OE1A 1004
OE1C E8      346      OE1C E8
OE1D 88      347      OE1D 88

LDY #12
LDA TABUL, X
BNE FRT
LDA #00
JSR DIR
RTS
CMP CHRCT
BNE FRTS
LDA #TSD
JSR OUT
LDA #20
JSR ZAUS
INX
DEY
BNE TOUT
JMP RCK
PHA
LDA #20
JSR ZAUS
PLA
JMP FRT
; =====
STA TABUL, X
INX
DEY
BNE REST
BEQ RLAF
; =====
CHECK LDX #0
LDY #12
LDA TABUL, X
BEQ ERROR
BMI ERROR
BEQ ERROR
INX
DEY
BNE CHE
RTS
; =====
ERROR PLA
PLA
JMP LOOP
; =====
TABE LDX #0
LDY #12
LDA TABUL, X
BEQ TAO
CMP CHMAX
BPL TAO
CMP CHRCT
BEQ INK
BPL TAI
INX
DEY
; HOLE POS>CHRCT
; ENDE
; POS>ZEILE
; POS GEFUNDEN

```


0E1E D0EE	348	TA1	BNE TA	0E85 B601	407	WAIT12	STX XTEMP	
0E20 A8	349		TAY	0E87 A603	408	LDX ZEIT2		
0E21 A61C	350		LDX CHRCT	0E89 9D1408	409	STA WTIMER, X		
0E23 B51C	351		STA CHRCT	0E8C A601	410	LDX XTEMP		
0E25 A930	352	TAUS	LDA #TBD	0E8E 60	411	RTS		
0E27 20330D	353		JSR OUT	0E8F	412	*****		
0E2A 98	354		TYA	0E8F ADO208	413	PARAM LDA PB		
0E2B 38	355		SEC	0E92 4A	414	LSR		
0E2C 861E	356		STX XTEM1	0E93 2903	415	AND #3		
0E2E E51E	357		SBC XTEM1	0E95 AA	416	TAX		
0E30 A8	358		TAY	0E96 BDA10E	417	LDA GTAB, X		
0E31 207D0D	359		JSR CRTIM	0E99 B502	418	STA ZEIT1		
0E34 4CF50C	360		JMP WEI	0E9B BDA50E	419	LDA GTAB+4, X		
0E37 A41A	361	TAO	LDX CHMAX	0E9E B503	420	STA ZEIT2		
0E39 A61C	362		LDX CHRCT	0EA0 60	421	RTS		
0E3B 841C	363		STY CHRCT	0EA1 BC34CC	422	GTAB HEX 8C34CC62		
0E3D 4C250E	364		JMP TAUS	0EA4 62	423	HEX 8C34CC62		
0E40	365		*****	0EA5	424	HEX 8C34CC62		
0E40	366		; V24=INGABE	0EA5 020201	425	HEX 8C34CC62		
0E40	367		; PB7=DATEN	0EA8 01	426	HEX 8C34CC62		
0E40	368		; PB1 U. 2=GESCHW.	0EA9	427	HEX 8C34CC62		
0E40	369		; PB6=EMPFANGSBEREIT-ANZ.	0EAC 0C0C0C		HEX 8C34CC62		
0E40 861E	370	ZEIN	STX XTEM1	0EAF 0C0C0C		HEX 8C34CC62		
0E42 841F	371		STY YTEM1	0EB2 0C060C		HEX 8C34CC62		
0E44 20BFOE	372		JSR PARAM	0EB5 0C060C		HEX 8C34CC62		
0E47 A900	373		LDA #0	0EB8 0C		HEX 8C34CC62		
0E49 BD0208	374		STA PB	0EB9 0C0C0C	427	HEX 8C34CC62		
0E4C 2C0208	375	EMP	BIT PB	0EBC 0C0C0C		HEX 8C34CC62		
0E4F 30FB	376		BMI EMP	0EBF 0C0C0C		HEX 8C34CC62		
0E51 A209	377		LDX #9	0EC2 0C0C0C		HEX 8C34CC62		
0E53 A502	378		LDA ZEIT1	0EC5 0C0C0C		HEX 8C34CC62		
0E55 4A	379		LSR	0EC8 0C		HEX 8C34CC62		
0E56 20B50E	380		JSR WAIT12	0EC9 0EEAF2	428	HEX 8C34CC62		
0E59 2C0208	381		BIT PB	0ECC 7771F1		HEX 8C34CC62		
0E5C 30EE	382		BMI EMP	0ECF F66B75		HEX 8C34CC62		
0E5E A900	383		LDA #0	0ED2 664187		HEX 8C34CC62		
0E60 48	384	NBIT	PHA	0ED5 2FABAA		HEX 8C34CC62		
0E61 207E0E	385		JSR WAIT	0ED8 C7		HEX 8C34CC62		
0E64 ADO208	386		LDA PB	0ED9 573332	429	HEX 8C34CC62		
0E67 0A	387		ASL	0EDC 373136		HEX 8C34CC62		
0E68 68	388		PLA	0EDF 352605		HEX 8C34CC62		
0E69 6A	389		ROR	0EE2 01C273		HEX 8C34CC62		
0E6A CA	390		DEX	0EE5 75F766		HEX 8C34CC62		
0E6B D0F3	391		BNE NBIT	0EE8 6F		HEX 8C34CC62		
0E6D A040	392		LDA #40	0EE9 C15B6E	430	HEX 8C34CC62		
0E6F BC0208	393		STY PB	0EEC 4F5F67		HEX 8C34CC62		
0E72 48	394		PHA	0EEF 594D7E		HEX 8C34CC62		
0E73 207E0E	395		JSR WAIT	0EF2 517D79		HEX 8C34CC62		
0E76 68	396		PLA	0EF5 7F696D		HEX 8C34CC62		
0E77 297F	397		AND #57F	0EF8 57		HEX 8C34CC62		
0E79 A61E	398		LDX XTEM1	0EF9 526361	431	HEX 8C34CC62		
0E7B A41F	399		LDA YTEM1	0EFC 5A6555		HEX 8C34CC62		
0E7D 60	400		RTS	0EFF 49624A		HEX 8C34CC62		
0E7E	401		*****	0F02 48SD7B		HEX 8C34CC62		
0E7E	402		; WARTEN AUF TIME-OUT	0F05 7A5345		HEX 8C34CC62		
0E7E	403		; UND TIMER NEU LADEN	0F08 76		HEX 8C34CC62		
0E7E 2C1508	404	WAIT	BIT RTIMER	0F09 0C1B2E	432	HEX 8C34CC62		
0E81 10FB	405		BPL WAIT	0F0C 0F1F27		HEX 8C34CC62		
0E83 A502	406	WAIT1	LDA ZEIT1					


```

0F0F 190D3E
0F12 113D39
0F15 3F292D
0F18 17
0F19 122321 433      HEX 1223211A251509220A0B1D3B3A13820C
0F1C 1A2515
0F1F 09220A
0F22 0B1D3E
0F25 3A1382
0F28 0C
0F29 434 ; #####
0F29 435 ; ZERO-PAGE-TABELLE
0F29 0A141E 436 ITAB HEX 0A141E283200
0F2C 283200
0F2F 000000 437      HEX 000000000000
0F32 000000
0F35 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 undefiniert 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 8	1 9	2 A	3 B	4 C	5 D	6 E	7 F
00...07	40...47	←	9	^	â	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.

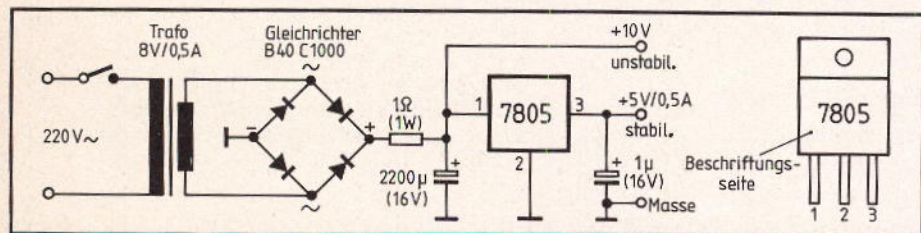
Literatur

- [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



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

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.

Herwig Feichtinger

Funkfernsehreib- Empfänger

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

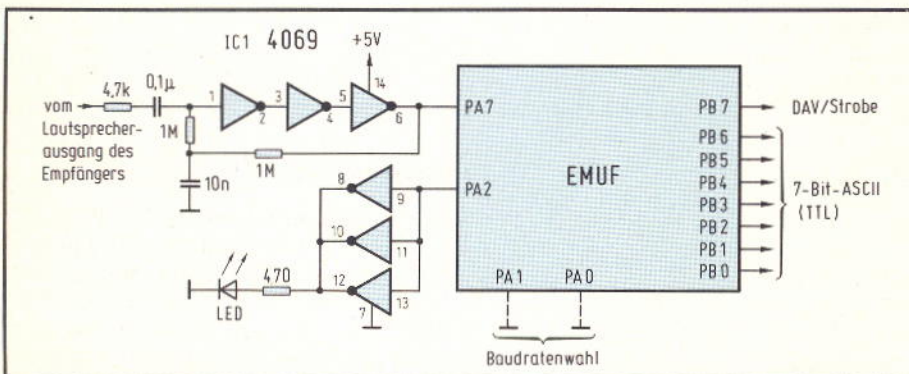


Bild 1. Externe Beschaltung des EMUF mit dem Funkfern-schreib-Empfangsprogramm im EPROM

Eine besondere Eigenschaft des hier vorgestellten Funkferschreib-Decoders ist, daß kein Empfangskonverter im herkömmlichen 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-Pegel anzuliefern; um den Rest kümmert

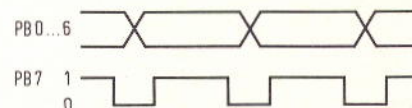


Bild 2. Zeitlicher Zusammenhang zwischen den ASCII-Datenausgängen PB 0...6 und dem DAV/Strobe-Ausgang PB 7

sich der EMUF. Die Empfindlichkeit der Decodierung entspricht derjenigen eines

guten PLL-Decoders; wenn vor dem Begrenzer zusätzlich noch ein Doppelfilter geschaltet wird, das nur die beiden Frequenzen 1275 Hz und 2125 Hz mit einer Bandbreite von je etwa 100 Hz passieren läßt, wird die Empfindlichkeit eines Filterkonverters 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-Pegel, wie die Daten garantiert gültig sind. Sie kann als Strobe-Signal zur Datenübernahme z. B. für ein Video-Interface, einen ASCII-Drucker mit Parallelschnittstelle oder das in diesem Heft beschriebene LCD-Interface verwendet werden. Bild 3 zeigt das vollständige Assembler-Listing.

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

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

Bild 3. Assemblerlisting der EMUF-Software

0000		;EMUF ALS RTTY-RX	OC06	78	SEI	
0000		.OPT GEN ;F.TABELLE	OC07	D8	CLD	
0000		;PB=ASCII-AUSGANG	OC08	E8	INX	
0000		;PB7=DAV/STROBE	OC09	8601	STX FLG	;BU-MODUS
0000		;PA 0/1=BAUDRATE	OC0B	A904	LDA £4	
0000		;PA7=NF-EINGANG	OC0D	8D0108	STA PAD	
0000		;PA2=LED-AUSGANG	OC10	LP	AD0008 LDA PA	;BAUDRATE
0000	PA	=\$800	OC13	2903	AND £3	;IN PA 0-1
0000	PAD	=\$801	OC15	AA	TAX	
0000	PB	=\$802	OC16	BD100D	LDA BDRT,X	
0000	PBD	=\$803	OC19	8500	STA BDR	
0000	BDR	*=+1 ;BAUDRATE	OC1B	20270C	JSR ASC	;EMPFANG
0001	FLG	*=+1 ;BU/ZI	OC1E	20BB0C	JSR OUT	;AUSGABE
0002	SUM	*=+1 ;AUTOKORR.	OC21	30ED	BMI LP	;JUMP
0003	SPL1	*=+1 ;BIT-	OC23		;BAUDOT ZU ASCII	
0004	SPL2	*=\$C00 ;MUSTER	OC23	LTR	A900 LDA £0	
OC00	RES	A2FF LDX ££FF	OC25	FIG	8501 STA FLG	
OC02		8E0308 STX PBD ;PB=AUSG.	OC27	ASC	20430C JSR BDT	;EIN-
OC05		9A TXS	OC2A		FOF7 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	;STARTBIT-	OCBB			;ASCII-AUSGABE	
OC49	CA	DEX	;ZEIT	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	;NEIN	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	;5BITS	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	;BAUDOT-	OCd1			;BAUDOT/ASCII-TAB.	
OC5F	4A	LSR A	;ZEICHEN	OCd1	TAB	4045	.BYT '£E',£A,'A SIU',£D	
OC60	60	RTS	;IN A	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	;MARK:	OCE2		5A4C		
OC6F	60	RTS	;C=1	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	OD01		352B		
OC84		88	DEY	OD03		2932		
OC85		DOFO	BNE RD1	OD05		2136		
OC87		AD0008	LDA PA	OD07		3031		
OC8A		0904	ORA £4	OD09		393F		
OC8C		8D0008	STA PA	OD0B		4040		
OC8F		A503	LDA SPL1	OD0D		2E2F3D		
OC91		F01C	BEQ ERR	OD10				
OC93		C9FF	CMP £\$FF	OD10			;BAUDRATEN-TABELLE	
OC95		F018	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 OC 8D 17 A4 A9 C2 8D 18 A4 A9 E8 A2 OF 8D 06
( ) OFAO A4 8E 07 A4 60 A9 05 A2 EF DO F3 20 AE EA BO 4D
( ) OFBO A2 10 AO 00 20 3E E8 A9 1C 20 58 EB 20 46 EA C8
( ) OFCO CA DO F1 20 FO E9 A9 3C 20 BC E9 20 93 E9 C9 20
( ) OFDO FO 03 4C 8D E1 AO 10 20 CD E2 A9 3E 20 7A E9 20
( ) OFEO 3E E8 20 DB E2 4C BO OF 2C 01 AO 70 FB 48 29 7F
( ) OFFO 20 A8 EE C9 OD DO 05 A9 OA 20 A8 EE 68 60 00 00
(M)=010C 4C 90 OF 4C
( ) 0110 A5 OF 4C AB
( ) 0114 OF
```

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

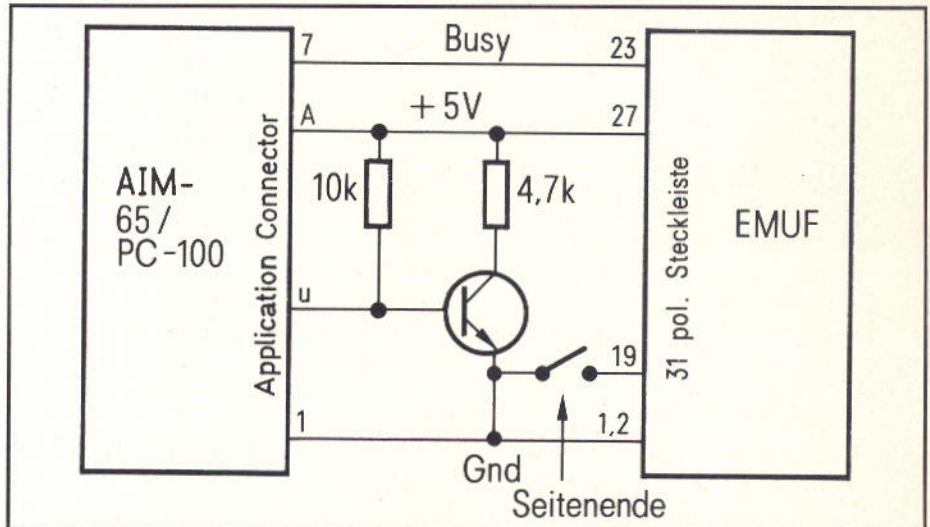


Bild 2. Nur vier Leitungen genügen, um den EMUF an den AIM-65/PC-100 anzuschließen und so die Typenradschreibmaschine anzusteuern

1. Schreibmaschine und AIM-65, wenn nicht schon geschehen, einschalten.
2. Jetzt die Verbindung zwischen AIM und EMUF durch Zusammenstecken z. B. der 31poligen 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 Prüf-text, 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 Prüf-text, 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 Prüf-text unter Beachtung aller im V.24-Standard festgelegten Quittungssignale. Bleibt eines der Signale aus, gibt

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.

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

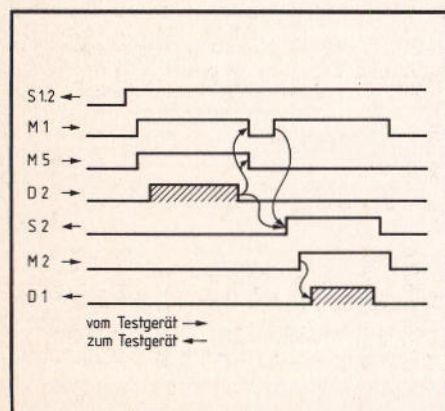


Bild 1. Zeitlicher Ablauf des V.24-Protokolls

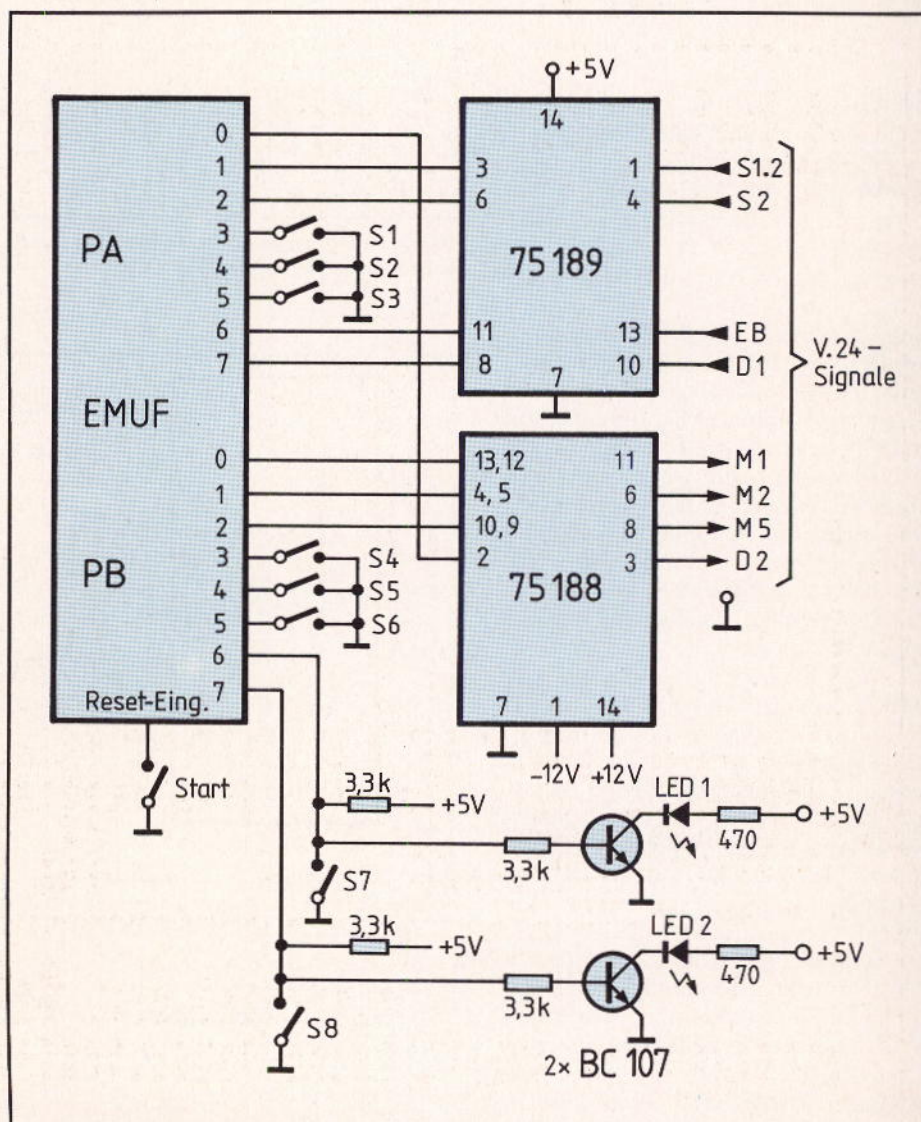


Bild 2. Schaltbild des V.24-Schnittstellentesters

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

0800	1	*****	6C47 48	5	PHA
0800	2	PAR EPZ 0	6C48 20576C	5	JSR WAIT
0800	3	BITS	6C4B 68	6	PLA
0800	4	PARI EPZ 2	6C4C 800008	6	STA PA
0800	5	ZEIT1 EPZ 3	6C4F A901	6	LDA #1
0800	6	ZEIT2 EPZ 4	6C51 F003	6	BEQ EINS
0800	7	XTEMP EPZ 5	6C53 20576C	6	JSR WAIT
0800	8	PTEMP EPZ 6	6C56 60	6	EINS
0800	9	BLMA0 EPZ 7	6C57	6	*****
0800	1	BLMA1 EPZ 8	6C57	6	JUP WARTEN AUF TIMOUT
0800	1	FEHFL EPZ 9	6C57 2C1508	6	WAIT
0800	1	*****	6C5A 10FB	6	BIT RTIMER
0800	1	*****	6C5C A503	7	BPL WAIT
0800	1	*****	6C5E 8605	7	WAIT1 LDA ZEIT1
0800	1	*****	6C60 A604	7	WAIT12 STX XTEMP
0800	1	*****	6C62 901408	7	LDA ZEIT2
0800	1	*****	6C65 A605	7	STA WTIMER,X
0800	1	*****	6C67 60	7	LDA XTEMP
0800	1	*****	6C68	7	RTS
0800	2	ORG \$6C00	6C68 A2FF	7	*****
0800	2	LDA #\$FF	6C6A 78	7	INIT
0800	2	TXS	6C6B A901	7	SEI
0800	2	JMP INIT	6C6B 800108	8	LDA #1
0800	2	SEND LDY #0	6C70 800008	8	STA D0RA
0800	2	JSR WAIT1	6C73 A907	8	STA PA
0800	2	LDA PAR	6C75 800308	8	LDA #7
0800	2	LDA PTEMP	6C78 800208	8	STA D0RB
0800	2	JSR BEGA	6C7B 0A	8	LDA PB
0800	3	INY	6C7C 9006	8	ASL
0800	3	CPY #50	6C7E 0A	8	BCC F0Q
0800	3	BNE NEUZ	6C7F 9055	8	ASL
0800	3	RTS	6C81 4C3C60	8	BCC FMQ
0800	3	*****	6C84 0A	9	JMP PROZ
0800	3	JUP AUSGABE EINES ZEICHENS	6C85 209A6C	9	ASL
0800	3	BEGA JSR WAIT	6C88 20066C	9	JSR PARAM
0800	3	LDA #0	6C8B 202860	9	JSR SEND
0800	3	STA PA	6C8E 202860	9	JSR SEK/4
0800	3	LDA TEXTA.Y	6C91 202860	9	JSR SEK/4
0800	4	WIEDH PHA	6C94 202860	9	JSR SEK/4
0800	4	JSR WAIT	6C97 4C886C	9	JMP WFOQ
0800	4	PLA	6C9A	9	*****
0800	4	STA PA	6C9A A200	9	JPARAMETER IN STEUERBLOCK
0800	4	LSR	6C9C 0A	1	PARAM LDA #0
0800	4	BCC NOCRY	6C9D 9001	1	ASL
0800	4	INC PTEMP	6C9F E8	1	BCC ZWEI
0800	4	NOCRY DEX	6CA0 8602	1	INX
0800	4	BNE WIEDH	6CA2 A207	1	STX PARI
0800	4	LDA PARI	6CA4 0A	1	LDA #7
0800	5	BEQ PRT	6CA5 9001	1	ASL
0800	5	LDA #1	6CA7 E8	1	BCC SIEB
0800	5	AND PTEMP	6CA8 8601	1	INX
0800	5	PHA	6CAA A200	1	STX BITS
0800	5	JSR WAIT	6CAC 0A	1	LDA #0
0800	5	PLA	6CAD 9001	1	ASL
0800	5	STA PA	6CAF E8	1	BCC 6RD
0800	5	PRT LDA #1	6CB0 8600	1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	STX PAR
0800	5	*****		1	GRD
0800	5	*****		1	INX
0800	5	*****		1	ST

[illegible]


```

6E0A 202860 2 JSR SEK/4
6E0D 60 2 RTS
6E0E 02 2 TEXTA
6E0F D4C8C5 2 ASC "THE QUICK BROWN FOX
JUMPS OVER THE LAZY DOG"
6E12 A0105C9C3CBA0C2D2CFD7CEA0C6CFD8A0CA05CD00D3
A0CFD6C5D2A0B4C8C5A0CC1DAD9A0C4CFC7
6E3A 0A0003 2 HEX 0A0003

```

```

6DA1 F003 BEQ NACK1
6DA3 4C4C6D JMP KACKNA
6DA6 20BA6D NACK1 JSR FNACK
6DA9 4C546D JMP SPROZ
6DAC *****
6DAC A980 KACKNA LDA #80
6DAE 8508 STA BLWA1
6DB0 A977 LDA #77
6DB2 8507 STA BLWA0
6DB4 200F6D LOOP JSR BLINK
6DB7 4CB46D JMP LOOP
6DBA A940 FNACK LDA #40
6DBC 8508 STA BLWA1
6DBE A987 LDA #87
6DC0 8507 STA BLWA0
6DC2 200F6D JSR BLINK
6DC5 60 RTS
6DC6 *****
6DC6 EE1408 JINTERRUPT-ROUTINE
6DC9 A509 IRQ INC TIMER
6DCB C901 LDA WFHL
6DCD F009 CMP #1
6DCF C902 BEQ FEH1
6DD1 F008 CMP #2
6DD3 C903 BEQ FEH2
6DD5 F00D CMP #3
6DD7 40 BEQ FEH3
6DD8 *****
6DD8 FEH1 FEHLERBEHANDLUNG
6DDA A940 FEH1 LDA #40
6DDC 800208 STA PB
6DDF 40 RTI
6DE0 A980 FEH2 LDA #80
6DE2 800208 STA PB
6DE3 40 RTI
6DE4 A9C0 FEH3 LDA #C0
6DE6 800208 STA PB
6DE9 40 RTI
6DEA *****
6DEA 20356D JEMPfang DER BESTAETIGUNG
6DEB 2C0008 EMPF JSR FTIME
6DF0 30FB EMP BIT PA
6DF2 78 BMI EMP
6DF3 A209 SEI
6DF5 A910 LDH #9
6DF7 205E6C LDA #10
6DFA A900 JSR WAIT12
6DFC 48 LDA #0
6DFF 20576C NBIT PHA
6E00 68 JSR WAIT
6E01 0E0008 PLA
6E04 6A ASL PA
6E05 CA ROR
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/NAK	
blinkt	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		on	on	on	on	on	on	on
S 2		on	on	on	on	on	on	on
S 3		on	on	on	on	on	on	on
S 4		on	on	on	on	on	on	on
S 5		on	on	on	on	on	on	on
S 6		on	on	on	on	on	on	on
S 7		on	on	on	on	on	on	on
S 8		on	on	on	on	on	on	on
Parität ungerade								
8 Datenbits								
mit Parität								
Parität gerade								
7 Datenbits								
ohne Parität								
Freilaufend ohne 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.

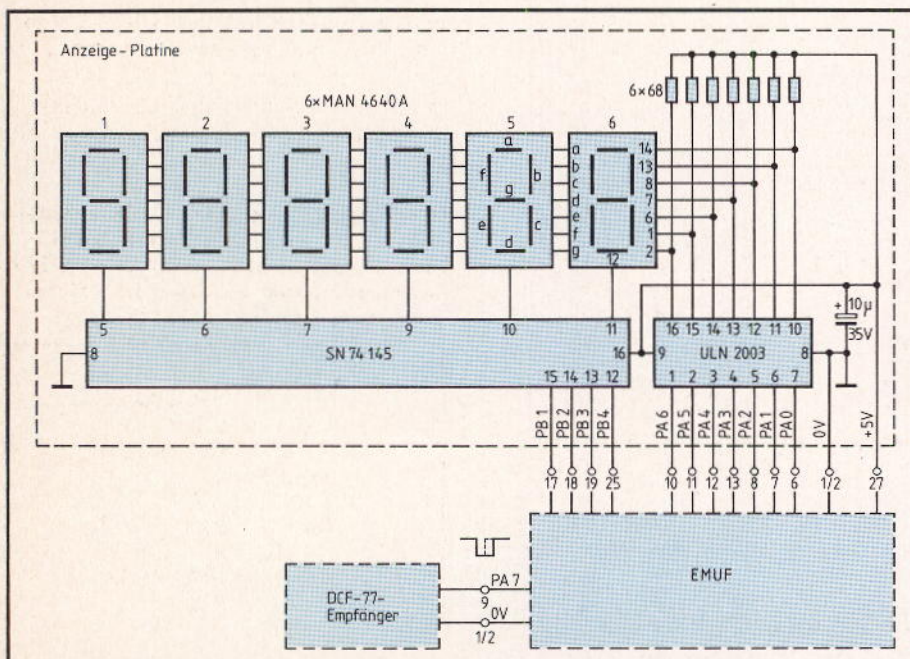


Bild 1. Schaltung zur Ansteuerung eines gemultiplexten sechsstelligen Siebensegment-Displays; der entsprechende Programmteil steht im Adressenbereich 0D8E...0DF8

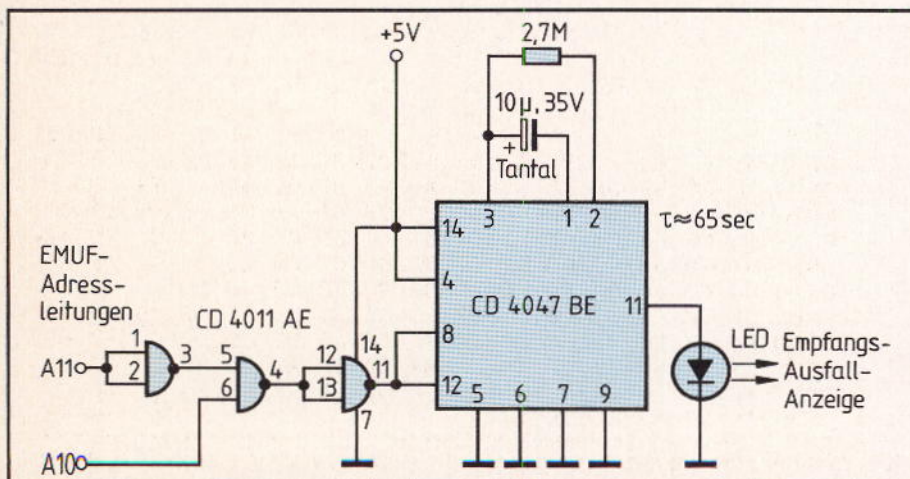


Bild 3. Eine kleine Zusatzschaltung zur Erkennung eines Empfangsausfalls hat noch auf der freien Lochrasterfläche der EMUF-Platine Platz

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 Adressenbereich 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-Ausfall-Anzeige 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-µF-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 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
( ) OCB0 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 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 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
```

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.O.2MS
DOFD BNE LOOP
60 RTS
;READY
.END
```

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

```
(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 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.

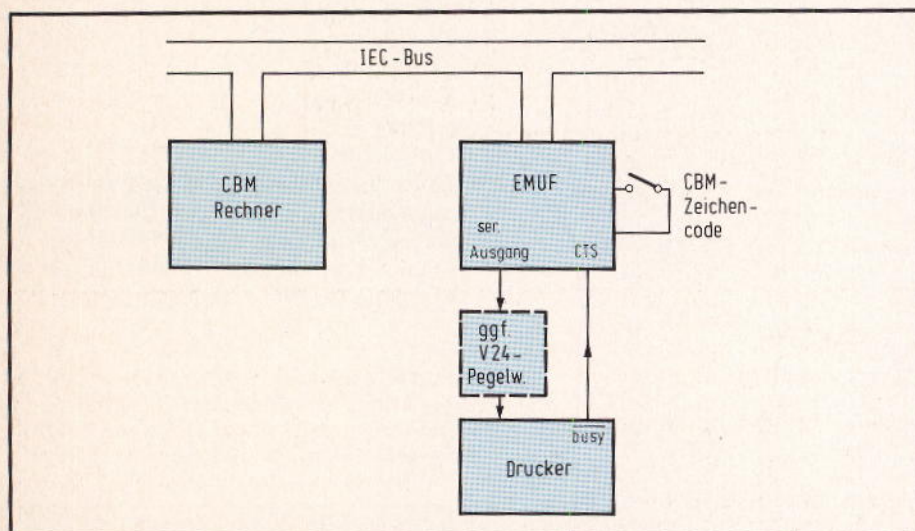


Bild 1. Anschluß eines V24-Druckers an einen CBM-Rechner

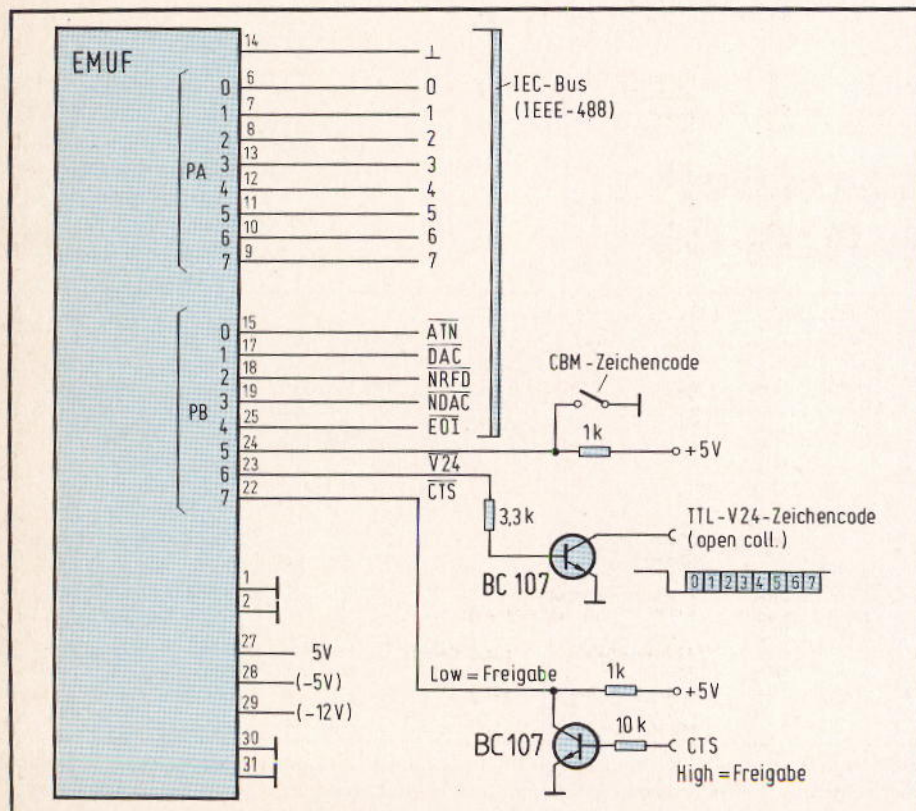


Bild 2. Die Steckerbelegung des EMUFs

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 wird umgewandelt. Ist er nicht 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

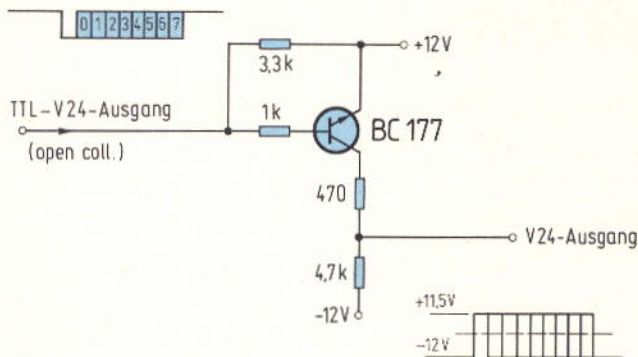


Bild 3. V24-Pegelwandler

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
;
0080      PA      EQU      $800
0081      PAD      EQU      $801
0082      PB      EQU      $802
0083      PBD      EQU      $803
;
0084      TIM1     EQU      $814
0085      TIM8     EQU      $815
0086      TIM64    EQU      $816
0087      TIMIN    EQU      $816
0088      TIMFLG   EQU      $817
;
0089      XTEMP1   EQU      $0
0090      ZOUT      EQU      0
0091      ZCOU      EQU      9
0092      FLAG      EQU      $A      ; IEC MERKER ATN ..
0093      ZEICH      EQU      $B      ; ZWSPEICHER
0094      COUNT      EQU      $C      ; ZWSPEICHER
0095      PADR      EQU      $D      ; FIRST TIME
0096      MDE      EQU      $E      ; MODE 0,1,2
0097      CNT      EQU      $F      ; ZEITSCHL.
;
;
; INIT ROUTINE
0FFC      ORG      $FFC
0FFC 000C      DW      $C00
;
0C00      ORG      $C00      ; START
0C00  A2FF      LD      $FF      ; STACKPOINTER
0C01  9A      TXS
0C02  A900      LDA      #00000000      ; ALL INPUT
0C03  8D0100    STA      PAD
0C04  A94C      LDA      #01001100      ; SET UP
0C05  8D0300    STA      PBD
0C06  A900      LDA      #00000000      ; NOT READY NOT ACCEPT
0C07  D8      CLD
0C08  78      SEI

```


[illegible]

---- SYMBOL TABLE ----

BAUD	0C27	MDE	000E	TALPA	0D5E
CKATN	0D2A	OKSK	0C2C	TIM1	0814
CNT	000F	PA	0000	TIM64	0816
CONSE	0D1A	PA0	0001	TIM8	0815
CONV	0C51	PADR	000D	TIMFLG	0817
CONV1	0C69	PB	0002	TIMIN	0816
CONV2	0C5E	PBD	0003	V241	0C8B
COUNT	000C	PETASC	0C47	V24LP	0C6B
DATEN	0D97	RESET	0C80	V24OUT	0C6A
FINA	0DAE	SEND	0D05	VL1	0C98
FLAG	000A	SEND1	0D04	VLO	0C79
GETCHA	0CA4	SENE01	0D28	VLOP	0C7C
LISTPA	0D61	SENE01	0D2C	VSK	0C94
LOPA	0CA9	SK2	0D78	XTEMP1	0000
LOPB	0CC5	SK3	0D80	ZCOU	0009
LOPMAI	0D92	TABBAU	0C39	ZEICH	000B
MAIN	0D44	TALKOF	0CEF	ZOUT	0008
MAIN1	0D49	TALKON	0CD3		

```

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 Lötstift 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

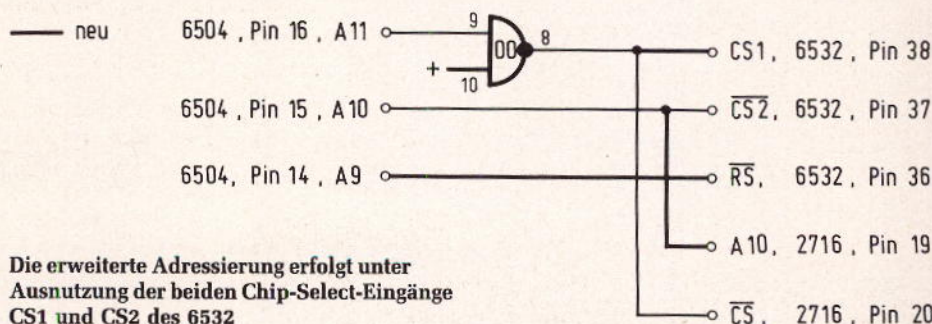
Lötseite mit einer Drahtbrücke mit Pin 8 des 7400 verbinden.

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)



COBOLO

„DER Lern- und Profi-Computer auf 3 Platinen“

COBOLO – ein Computer mit zauberhaften Qualitäten dank eines neuen, raffinierten Hardware-Konzepts und eines sagenhaft komfortablen Betriebssystems,

COBOLO – ein Maschinensprache-Computer auf Basis 6502/65C02, der auch Textverarbeitung, BASIC- und FORTH kann,

COBOLO – DER Computer für alle – auch Ihre – Problemstellung,

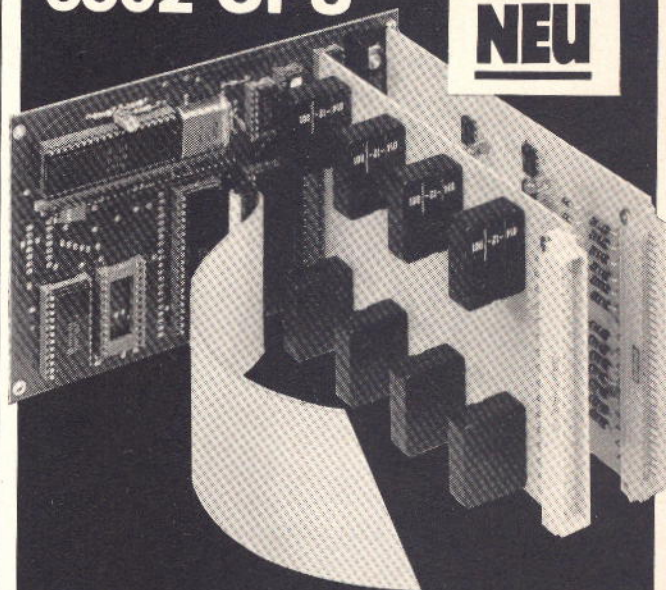
COBOLO – mit Handbuch „6502/65C02 Maschinensprache“ aus dem Heise-Verlag.

WOLFRAM FEISE
MICROPROZESSORTECHNIK

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

6502 CPU

NEU



Einplatinen-Computer

mit I/O-Karten

Bestückung: 6502, 6522, 2x 6521
RAM 2 Kx8, EPROM 2516...2764
(256), PROM-Adress-Decoder.
Flachkabel oder Karten aufsteckbar. Bus auf Stecker geführt.

Karten mit:

Relais, Opto-Kopplern, Triacs für
Netztrennung (VDE).
RS 232 C/V 24. Kalenderuhr.
ADW/DAW. Sensoranpassung.
Induktivegeber-Anschaltung.
Sonderkarten. Fädelkarten.

Lieferung auch mit FORTH und
BASIC für rationellere

Programm-Erstellung
(EPROM auf der Karte).
FORTH läuft mit annähernder
Maschinensprachen-
Geschwindigkeit.

Für kleinere Aufgaben:
EMUF (6504) und MC-Software
in EPROM lieferbar.

Wir liefern
auch fertige
Problemlösungen
in Hard- und
Software (Meß-
technik und Prüf-
automaten).

Beratung und Service:

WIRTH

Dipl. Ing. Hans W. Wirth
Mühlstr. 25, D-7064 Remshalden 1
Tel. (0 71 51) 7 12 26, 7 32 03

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), Flachfassungen für alle
ICs (auch EPROM) sowie 31pol. Stiftleiste.

Komplettpreis (Bausatz) DM 89.- Platine einzeln DM 27.50

2716-MC 1 ... IEC/V24 (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
Selbsttestprogramm mit Beschreibung (noch nicht veröffentlicht) DM 24.50

2716-MC 4 ... V24-Schnittstellentester u. Interface für Praxis 30 aus Sonderheft DM 19.50

2716-MC 5 ... Funklärnschreibempfänger 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

2716-MC 8 ... Heizungsregelung (MC 1/83) und Eieruhr (MC 5/83) DM 19.50

Z80-EMUF-BS nach 4-83 MC DM 99.-

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

Komplettpreis DM 99.-

Z80-Einplatinencomputer 1-82 MC

Komplettpreis (Bausatz) DM 225.- Platine einzeln DM 79.-

BASIC-Einplatinencomputer MC 2/3-83

Kompletter Einplatinencomputer für Steuerungen, Meßaufgaben usw., der direkt in Basic programmierbar ist. Da bisher nahezu alle kleinen Einplatinencomputer in Maschinensprache programmiert werden mußten, war ihr Einsatz relativ aufwendig und nur „Spezialisten“ vorbehalten.

Kurzinformation mit Preisliste kostenlos.

MMC-5-PR : Leierplatte DM 95.- MMC-5-MA : MMC-5, Anleitung separat DM 25.-

MMC-5-BS : Bausatz mit Fassungen, 2 KB-RAM ohne EPROM u. A/D-Wandler-Teil DM 395.-

MMC-5-KP : Kompletter Bausatz mit 2 KB-RAM wie oben DM 550.-

Ladenverkauf: Breslauerstr. 29, 6900 HD-Kirchheim
Geschäftsz. 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
Zwischenverkauf vorbehalten.

NEU

Herwig Feichtinger

Der Eier-EMUF

Eigentlich geht es hier nicht nur um das richtige „Timing“ beim Eierkochen, sondern auch bei der Tee-, Reis- oder sonstigen Speisenbereitung. Entstanden ist das Ganze, um bei „Freak's Frau“ mehr Sinn für die Notwendigkeit der Mikrocomputer im Haushalt zu wecken.

Der Eier-EMUF (Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung [1]) ist eine Art Eieruhr, die sich mit sieben Tasten auf Zeiten von 1, 3, 5, 7, 10, 15 oder 20 Minuten vorprogrammieren läßt. Nach Ablauf der Zeit

rasselt aber nicht etwa eine Klingel, sondern eine von vier vorprogrammierten Melodien ertönt aus dem eingebauten Lautsprecher. Mit einer weiteren Taste ist es möglich, den Ablauf zu unterbrechen, um eine neue Zeit einzugeben.

Auch nach Drücken dieser Taste „0“ wird eine Melodie gespielt. Während die Zeit läuft, blinkt außerdem eine Leuchtdiode im 0,5-Sekunden-Takt.

Das Programm (Bild 1) besteht im wesentlichen aus zwei Teilen: Einer Zeitschleife und einem Musik-Erzeugungs-Programm. Um die gewünschte Zeit zu erhalten, wird zunächst der Port PA abgefragt. Ist eine Taste gedrückt, so wird ihre Nummer (0...7 für PA 0...7) festgestellt; sie steht dann im X-Register. Der gewünschte Zeitwert wird dann X-indiziert aus einer Tabelle TAB gelesen. Damit wird die eigentliche Zeitschleife gestartet; sie benützt den Timer im 6532-Baustein des EMUF, um 0,25-s-Zeitabschnitte zu erzeugen.

Das Musikprogramm wurde in ähnlicher Form und mit den gleichen Melodien bereits im EMUF-Sonderheft des Fran-

```

0000 WORK      *=+6
0006 LIMIT     *=+3
0009 VAL2      *=+1
000A VAL1      *=+1
000B TIMER     *=+1
000C XSAV      *=$C00
0000 MUS       =$D00      ;NOTENTAB.
0000 PA        =$800
0000 PB        =$802
0000 PBD       =$803
0000 TK        =$817
0000 RESV      =$FFC
0000 RES       A2FF LDX $FF      ;RESET
0002           9A TXS
0003           A230 LDX $%00110000
0005           8E0308 STX PBD      ;PB=AUSG
0008           A9DF LDA $%11011111
000A           8D0208 STA PB
000D           78 SEI
000E           D8 CLD
000F FRST      A205 LDX $5      ;ERSTE
0011 LP1       BDD00C LDA INIT,X  ;MEL.
0014           9500 STA WORK,X
0016           CA DEX
0017           10F8 BPL LP1
0019 WAIT      A207 LDX $7      ;TASTE
001B           AD0008 LDA PA      ;GEDR.?
001E SRCH      6A ROR A
001F           9006 BCC FND      ;JA
0021           CA DEX
0022           10FA BPL SRCH      ;NEIN

```

```

OC24           30F3 BMI WAIT
OC26 FND       BDD50C LDA TAB,X  ;ZEIT
OC29           AA TAX
OC2A LOP       AOE6 LDY $230      ;1MIN
OC2C LOP0      A9FF LDA $255
OC2E           8D1708 STA TK
OC31 LOP1      2C0008 BIT PA      ;PA7=L?
OC34           1013 BPL GO      ;JA
OC36           2C1708 BIT TK
OC39           10F6 BPL LOP1
OC3B           AD0208 LDA PB
OC3E           4910 EOR $10      ;BLINKEN
OC40           8D0208 STA PB
OC43           88 DEY
OC44           DOE6 BNE LOP0
OC46           CA DEX
OC47           DOE1 BNE LOP
OC49           ;AB HIER FOLGT DAS
OC49           ;MUSIK-PROGRAMM (GO)
OC49           .OPT LIST
OC49           .BYT 0,20,15,10
OC49           14
OC49           OF
OC49           OA
OC49           07 .BYT 7,5,3,1
OC49           05
OC49           03
OC49           01
OC49           *=$RESV
OC49           000C .WOR RES
OC49           OFFE ERRORS= 0000

```

```

OC00 A2 FF 9A A2 30 8E 03 08 A9 DF 8D 02 08 78 D8 A2
OC10 05 BD CF 0C 95 00 CA 10 F8 A2 07 AD 00 08 6A 90
OC20 05 CA 10 FA 30 F3 BD D5 0C AA A0 E6 A9 FF 8D 17
OC30 08 2C 00 08 10 13 2C 17 08 10 F6 AD 02 08 49 10
OC40 8D 02 08 88 D0 E6 CA D0 E1 A0 00 B1 04 C9 FF F0
OC50 BE E6 04 D0 02 E6 05 C9 FA F0 BE 90 0F E9 FB AA
OC60 B1 04 E6 04 D0 02 E6 05 95 00 B0 DD A6 00 86 07
OC70 A6 01 A8 30 02 A2 01 86 06 29 7F 85 09 F0 02 85
OC80 0A A5 09 25 03 F0 04 E6 0A C6 09 A6 09 AD 02 08
OC90 09 20 20 A6 0C 30 B2 A6 0A 78 AD 02 08 29 DF 20
OCA0 A6 0C 30 A5 10 DB A4 02 84 0B 86 0C E0 00 D0 08
OCB0 A6 0C C6 0B D0 F6 F0 16 8D 02 08 CA C6 08 D0 EC
OCC0 C6 07 D0 E8 A4 00 84 07 C6 06 D0 E0 A9 FF 60 30
OCD0 02 01 FF 00 OD 1E 14 0F 0A 07 05 03 01 00 09 98

```

```

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
ODE0 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
ODFO 2F 29 24 2F 29 A4 2F 29 2F 24 2F 29 A4 2F 29 2F
OE00 24 2F 29 A4 32 A9 AF 80 80 FF FF FF FF FF FF FF
OFFC 00 0C

```

Bild 1. Programmlisting der melodischen Eieruhr mit der Notentabelle ab OD00

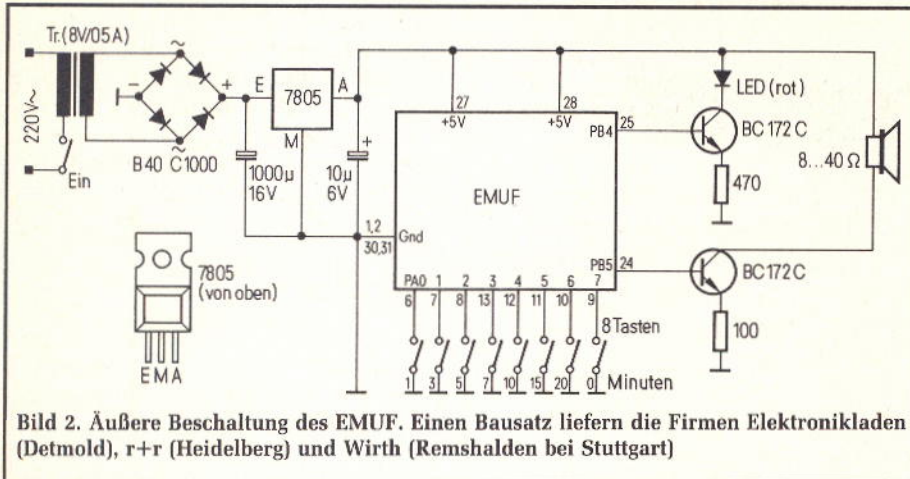


Bild 2. Äußere Beschaltung des EMUF. Einen Bausatz liefern die Firmen Elektronikladen (Detmold), r+r (Heidelberg) und Wirth (Remshalden bei Stuttgart)

zis-Verlages veröffentlicht. Es beruht auf einer von Jim Butterfield einstmals für den KIM-1 entwickelten Routine. Dabei wird eine Notentabelle im Adressenbereich ab 0D00 verwendet. (Die Codierung wurde in [2] bereits beschrieben.) Die äußere Beschaltung des EMUF ist sehr einfach und beschränkt sich auf zwei Treibertransistoren für Lautsprecher und LED sowie ein einfaches 5-V-

Netzteil (Bild 2). Der EMUF-Bausatz ist für weniger als 100 DM u. a. bei r + r (Heidelberg) erhältlich.

Literatur

- [1] Mädchen für alles (EMUF-Bauanleitung), mc 1981, Heft 2, und EMUF-Sonderheft, Franzis-Verlag.
- [2] Türklingel und Alarmanlage. EMUF-Sonderheft, Franzis-Verlag.

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

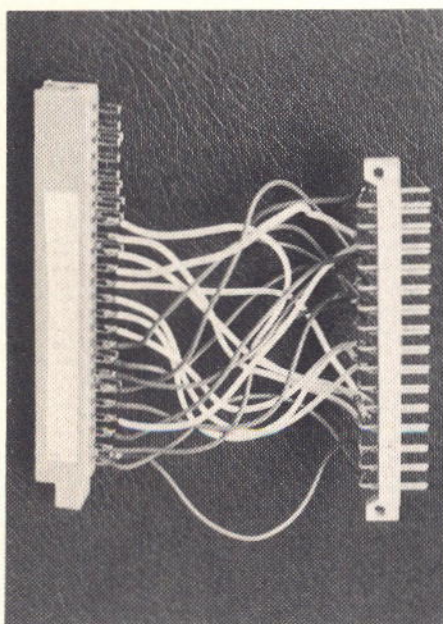
Verwendet man den AIM-65 oder PC-100 zur EMUF-Programmentwicklung, so kann man dazu ein einfaches

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.



Es gibt zwei Sorten Programme.

Die einen sind so kurz, daß sie offensichtlich keine Fehler enthalten.

Die anderen sind so komplex, daß sie keine offensichtlichen Fehler enthalten.

(Aus einer Informatik-Vorlesung an der TU München)

Otmar Hacker, Mathias Ott

Selbstlernende Haus-Heizungs- regelung

Mit geringem Aufwand an Hard- und Software hält die hier beschriebene Heizungsregelung die Haus-Innentemperatur im Tag- und Nachtabsenk-Betrieb auf ein Grad Celsius genau. Irgendwelche Regler-Einstellungen sind nicht nötig, da sich der Computer – hier ein EMUF – den veränderten Bedingungen wie Haustyp, Wetterlagen usw. selbst anpaßt. Aus der stets richtigen Reglersteuerung und der guten Regeldynamik resultiert eine erhebliche Energie-Ersparnis.

Bei dem verwendeten selbstlernenden, adaptiven System handelt es sich um eine suchalgorithmische Optimierung mit veränderlicher Suchschrittweite. Die Messung und Ausregelung der Temperaturen erfolgt mit einer Abtastregelung.

Das Programm wurde zunächst mit einem PC-100 (alias AIM-65) entwickelt und getestet, später aber auf den Einplatinen-Computer EMUF übertragen, den mc 1981 in Heft 2 sowie im EMUF-Sonderheft ausführlich beschrieb und der weniger als 100 DM kostet.

Heizungsregelungen: Ein wenig Theorie

Herkömmliche Haus-Heizungsregelungen verstellen bei Konstanttemperatur-Kesseln die Heißwasser- oder Vorlauftemperatur T_v mit einem Mischventil (Bild 1). Das geschieht abhängig von der Außentemperatur T_a nach gekrümmten Kurven. Die Haus-Innentemperaturen sollen dadurch annähernd auf einem konstanten Wert (z. B. 20 °C) gehalten werden.

Dazu muß die Regelschaltung mit bis zu fünf Einstellknöpfen an das Haus angepaßt werden. Eine wirklich exakte Temperaturregelung ist so nur schwer zu erreichen und ein Überspringen der Raumtemperaturen kaum zu vermeiden. Denn die nötige Vorlauftemperatur hängt nicht nur von der Außentemperatur, sondern auch von anderen Witterungseinflüssen wie Windstärke oder Luftfeuchtigkeit ab. Die nicht exakte Einstellung führt auch zu einem unnötig hohen Energieverbrauch.

Die adaptierende Regelung dagegen nutzt die Möglichkeiten der Mikrocomputer-Intelligenz, um mit Hilfe ausgeklügelter Programme derartige Anpassungsarbeiten zu ersparen. Wie Bild 2 zeigt, wird ein Pendeln oder Überspringen der Temperaturen insbesondere beim Wiederaufheizen am Morgen

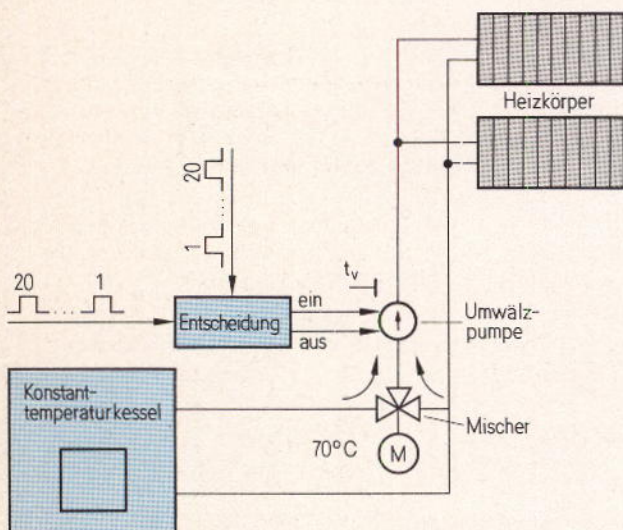


Bild 1. Anordnung einer Heizungsanlage mit Konstanttemperatur-Kessel. Die Vorlauftemperatur, also die Temperatur des Wassers in den Heizkörpern, wird über den Mischer gesteuert

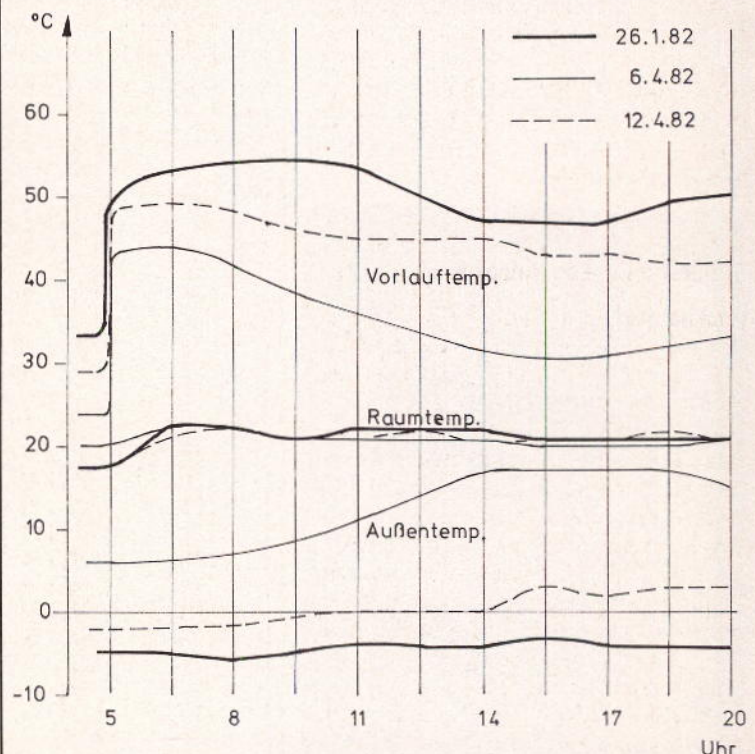


Bild 2. Temperaturverlauf an drei typischen Tagen

vermieden. Auch ein Nachregeln durch Thermostat-Ventile an den Heizkörpern ist nicht nötig; allerdings können solche Ventile Einflüsse wie direkte Sonneneinstrahlung an der Südseite ausgleichen.

Das adaptive Regelsystem

Bei unserer selbstlernenden Regelung handelt es sich im Prinzip um eine Nachlauf-Regelung der Heizwasser-Vorlauftemperatur T_v zu einer per Fühler gemessenen Außentemperatur T_a als Führungsgröße. Die Messung der Außentemperatur ist nötig, weil die unmittelbare Regelung der Innentemperatur wegen der großen Verzögerung verspätet korrigiert würde: Überspringen und Pendeln wäre die Folge.

Unter Abtastregelung versteht man, daß die einzelnen Temperaturfühler stets nacheinander erfaßt und verarbeitet werden, wie es typischerweise in einem Computer geschieht.

Der Computer ermittelt die optimale Kurve der Regelung selbst durch einen Suchalgorithmus. Je kleiner die Temperaturabweichungen werden, desto kleiner werden auch die Suchschritte, so daß mit der Zeit eine immer bessere Anpassung an die tatsächlichen Verhältnisse stattfindet.

Bild 3 zeigt die dafür nötige Schaltung. Der Hardware-Aufwand für das selbstlernende System ist erstaunlich gering: hier ein EMUF (Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung) mit 1 KByte EPROM und einem 6504-Prozessor. Zur Programmentwicklung diente ein Tischcomputer PC-100 mit Assembler.

Trickreiche Temperaturmessung

Eine besonders wirtschaftliche Realisation einer Temperaturmessung per Computer ergibt sich durch die Anwendung eines Tacho-Bausteins (LM 2907). Herkömmliche D/A-Wandler würden dafür mindestens zehn Rechneranschlüsse benötigen; der Tachobaustein kommt mit zweien aus.

Der Computer gibt eine Frequenz aus, die per Programm erzeugt wird, und der Tachobaustein macht daraus eine der Frequenz proportionale Spannung. Ein Operationsverstärker vergleicht diese mit dem Ausgangssignal eines Temperaturfühlers und leitet das Ergebnis wieder dem Computer zu.

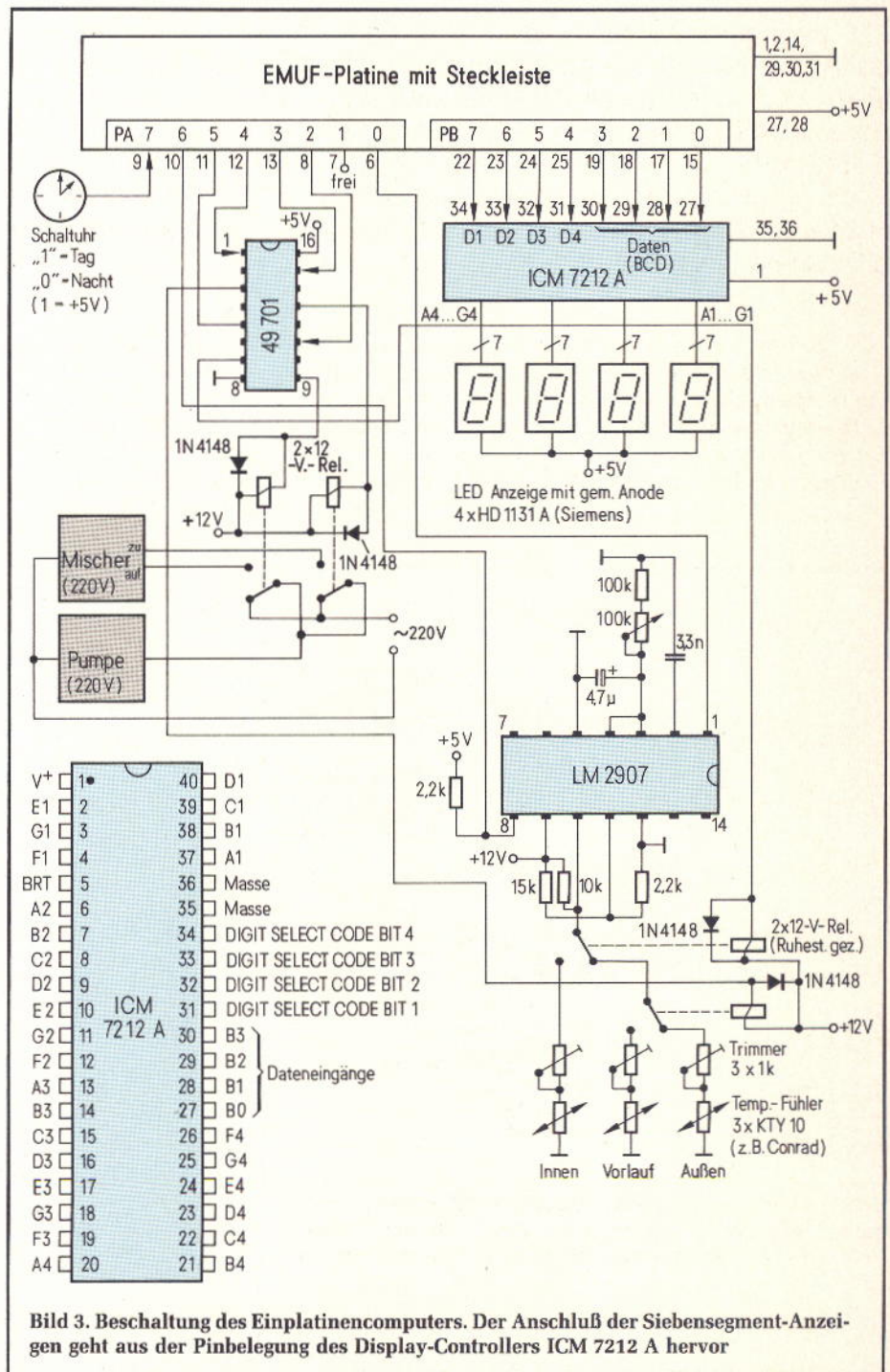
Der Rechner „wobbelt“ zunächst über ein relativ breites Frequenzband. Sobald der Operationsverstärker im LM 2907

ihm meldet, daß die der gemessenen Temperatur entsprechende Frequenz erreicht ist, schaltet er auf ein Frequenzband um, das nur wenig unter dieser beginnt. Dadurch können die Temperaturen dann wesentlich schneller gemessen werden. Vorlauf- und Außentemperaturen werden zur Sicherheit mehrmals gemessen; der Regelzyklus beträgt 90 Sekunden. Über die Pulslänge wird der Mischermotor proportional geregelt, ein weiterer Computer-Ausgang schaltet seine Drehrichtung.

Die Innentemperatur wird wegen der großen Haus-Zeitkonstanten nur etwa alle 90 Minuten gemessen – ebenfalls mehrmals hintereinander, um Falschwerte auszuschalten, die z. B. durch Störpulse entstehen könnten. Auch wird ein zu häufiges Motor-Ein- und -Ausschalten vermieden.

Temperaturabsenkung nachts

Eine externe, netzsynchrone Schaltuhr, wie sie im Handel preisgünstig zu haben



ist, gibt am Abend einen Impuls an den Computer ab. Dieser verschiebt die Regelgerade nach unten; die Regelung selbst bleibt jedoch wie tagsüber wirksam. Allerdings wird die eingestellte Heizkurve nicht mehr verändert. Am Morgen verschiebt ein weiteres Signal der Uhr die Gerade auf den letzten Stand vor der Nachtabenkung. Bei Abwei-

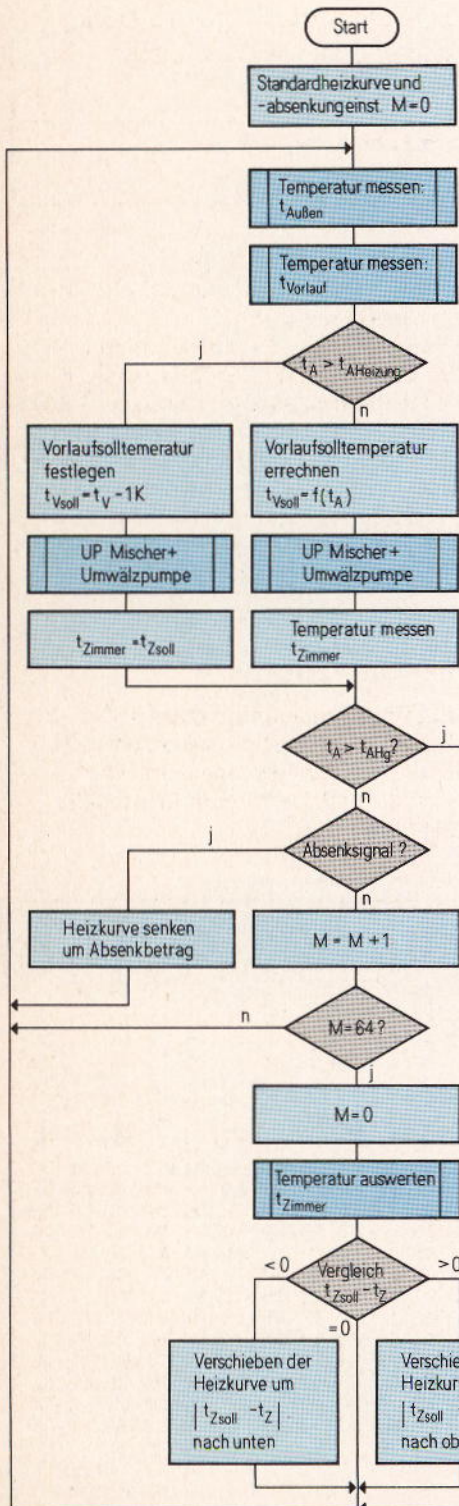
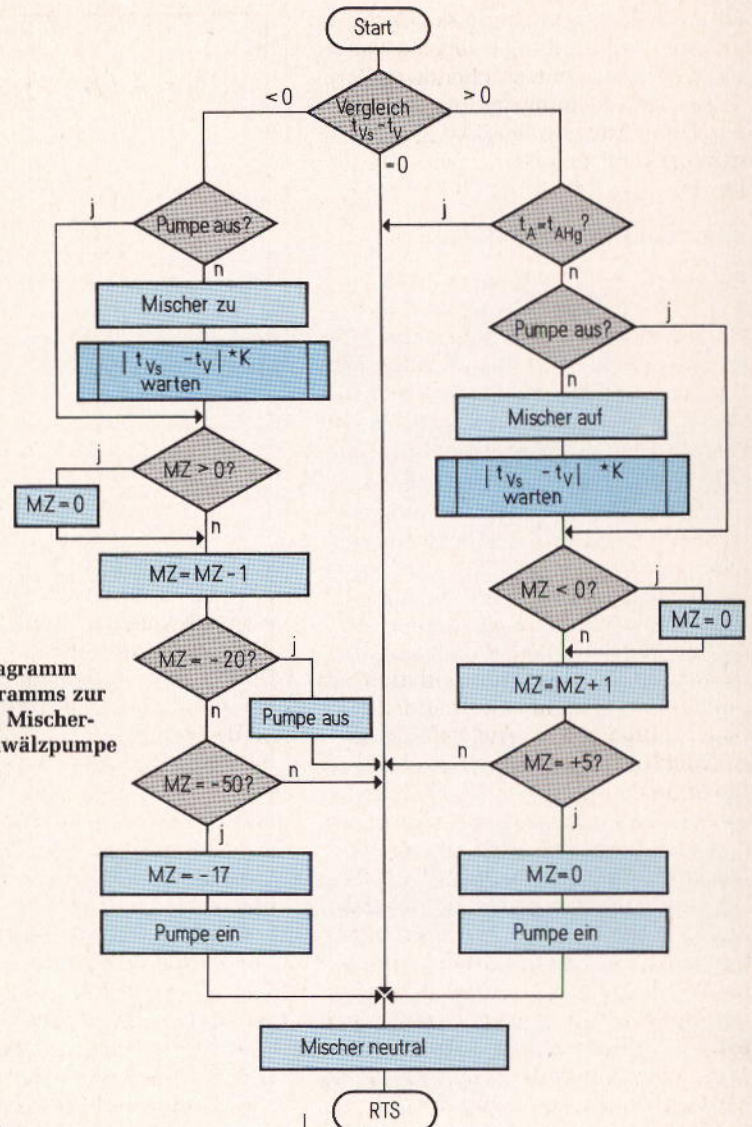


Bild 4. Flußdiagramm des EMUF-Hauptprogramms

Bild 5. Flußdiagramm des Unterprogramms zur Steuerung von Mischer-ventil und Umwälzpumpe



Zeichenerklärung für das Heizungsreglerprogramm

M	= Zykluszähler für Raumtemperaturauswertung
t _A	= t _{Außen} = Meßwert: Außentemperatur
t _V	= t _{Vorlauf} = Meßwert: Vorlauftemperatur
t _Z	= t _{Zimmer} = Meßwert: Zimmertemperatur
t _{Vs}	= t _{Vorlaufsol} = Vorlaufsoltemperatur
t _{Zs}	= t _{Zimmersoll} = Raumsolltemperatur
t _{AHg}	= t _{AHeizgrenze} = Außentemperatur, ab der keine Heizung mehr benötigt wird
MZ	= Zähler für Mischer- und Pumpensteuerung
K	= Konstante zur Zeitsteuerung

chungen der Innentemperatur merkt sich der Rechner für die nächste Nachtabsenkung eine entsprechende weitere kleine Verschiebung der Regelgeraden vor. Diese Anpassung mittels Suchalgorithmus kann sich einige Tage wiederholen, bis eine optimale Regelung erreicht ist.

Bei steigenden Außentemperaturen, z. B. über 14 °C, mußte bisher die Umwälzpumpe noch von Hand abgeschaltet werden, ebenso nachts. Dies besorgt jetzt der Rechner ohne zusätzlichen Hardware-Aufwand, denn der Pumpenschalter wird von den vier Relais-Verstärkern mit zwei Rechnerausgängen mitgesteuert. Die Pumpe wird abgeschaltet, nachdem der Mischer 20 Schließbefehle erhalten hat.

Vor dem Abschalten ist sichergestellt, daß das Mischerventil völlig zu ist, um schädliche „Schwerkraftzirkulation“ im Heißwasserkreislauf zu vermeiden. Gleichzeitig werden „Auf-Befehle“ unterdrückt, um ein Pendeln des Ventiles zu vermeiden. Bei stehender Pumpe wird die Innentemperatur gemessen, aber nicht mehr zur Adaption benutzt. Damit verhindert man, daß bei erhöhten Temperaturen infolge Sonneneinstrahlung z. B. die T_v - T_a -Gerade falsch adaptiert wird. Außerdem wird die Pumpe im Stundenzyklus 2 bis 5 Minuten eingeschaltet. Damit erfaßt der Vorlaufemperatur-Fühler auch die tatsächliche Temperatur des Wassers und Einfrieren wird verhindert. Bei stehender Heizungspumpe werden die Mischerbefehle

Antiquarisches

... Aber ich sage, daß diejenigen Psychologen, die zwischen der gefühlsbetonten Handlung des Menschen und anderer Lebewesen und der Handlung des modernen Typs von automatischen Mechanismen scharfe und prinzipielle Unterschiede machen, ebenso vorsichtig bei ihren Einwänden sein sollen wie ich bei meinen Behauptungen.

Norbert Wiener, im Buch „Mensch und Maschinaschinen“, geschrieben 1949

zunächst unterdrückt. Die Pumpe wird eingeschaltet, nachdem fünf Mischer-Auf-Befehle anstanden. Die LED-Anzeigen zeigen die drei Temperaturen in °C als 3. und 4. Stelle. Die 2. Stelle bedeutet mit „-“ negative, fehlende Anzeige positive Temperaturen. Die 1. Stelle (links) signalisiert mit „1“, „2“ und „3“ die Außen-, Vorlauf- und die Raumtemperaturen im Tagesbetrieb mit „4“, „5“ und „6“ den Nachtabsenkungsbetrieb der zwei rechten Stellen. Mit Hilfe der drei Trimmwiderstände werden die Temperaturwiderstände am Display eingestellt bzw. kontrolliert. Die Bilder 4 und 5 geben die Flußdiagramme des Hauptprogramms und des Unterprogramms zur Steuerung des Mischerventils und der Vorlaufpumpe wieder. In der Tabelle finden sich die dabei

verwendeten Abkürzungen für die Programm-Parameter. Bild 6 zeigt das fertige Maschinenprogramm für den EMUF. Zusätzlich ist noch der Reset-Vektor zu setzen: 0FFC = 00, 0FFD = 0C. Sinnvollerweise überprüft man das Programm vor dem Brennen in ein EPROM mit Prüfsummen. Diese betragen (jeweils von Anfangsadresse bis Endadresse - 1 aufsummiert):

6F38 für 0C00...0D00,
674C für 0D00...0E00,
801E für 0E00...0F00 und
6213 für 0C00...0F28.

Ein Prüfsummenprogramm für den als Entwicklungssystem geeigneten AIM-65 oder PC-100 findet sich in mc 1981, Heft 2, auf Seite 36, oder in mc 1982, Heft 5, Seite 55.

0C00	A2	FF	9A	D8	86	3A	86	3B	A9	FO	85	11	85	21	85	31	0DA0	DO	DA	86	41	A9	OA	85	14	4C	04	OE	EA	EA	86	41	18
0C10	A9	3F	8D	01	08	EA	EA	EA	EA	A9	80	85	38	A9	FO	85	0DB0	A5	42	65	41	85	42	A5	43	69	00	85	43	C6	1A	DO	AA
0C20	36	A9	00	85	35	A9	FF	8D	03	08	A9	00	85	32	A9	00	ODC0	26	42	26	43	26	42	26	43	A9	00	85	14	A5	43	85	41
0C30	85	33	85	34	A5	21	85	41	A9	FF	EA	20	59	OD	A5	41	ODD0	38	E9	D6	85	20	10	OA	A2	OF	38	A9	00	E5	20	4C	E3
0C40	DO	02	A9	CC	85	21	A5	11	85	41	A9	EF	EA	20	59	OD	ODE0	OD	A2	OA	8E	02	08	A8	8A	09	40	8D	02	08	98	C9	64
0C50	A5	41	DO	02	A9	90	85	11	A5	21	C9	C9	10	OA	A9	FF	ODFO	30	06	38	E9	64	4C	EE	OD	C9	OA	30	08	38	E9	OA	E6
0C60	20	1F	OE	A9	30	4C	7E	OC	38	65	11	20	1C	OE	A5	31	OE00	14	4C	F8	OD	8D	02	08	09	10	8D	02	08	A5	14	8D	02
0C70	85	41	A9	DF	EA	20	59	OD	A5	41	DO	OD	A9	10	85	32	OE10	08	09	20	8D	02	08	20	00	OF	60	EA	EA	38	E5	38	85
0C80	A9	00	85	33	85	34	EA	A9	C1	85	31	A5	35	FO	57	AD	OE20	1B	30	41	A5	21	C9	C9	FO	37	A5	3A	29	OC	FO	19	A9
0C90	00	08	29	80	DO	26	A5	32	29	08	FO	10	AO	03	66	33	OE30	FB	8D	00	08	A5	1B	DO	08	A9	FF	8D	00	08	4C	48	OE
0CA0	66	34	88	DO	F9	A5	34	85	37	4C	2A	OC	18	A5	34	65	OE40	20	BO	OE	C6	1B	4C	36	OE	A5	3B	10	04	A9	00	85	3B
0CB0	31	85	34	A5	33	69	00	85	33	4C	OE	OD	A9	00	85	35	OE50	E6	3B	A5	3B	C9	05	DO	08	A9	FF	85	3A	A9	00	85	3B
0CC0	38	A5	36	65	38	85	38	A5	37	38	E9	C6	18	65	36	C9	OE60	60	EA	EA	EA	A5	3A	29	OC	FO	19	A9	F7	8D	00	08	A5
0CD0	EA	10	02	A9	EA	85	36	EA	EA	EA	EA	EA	EA	EA	EA	EA	OE70	1B	DO	08	A9	FF	8D	00	08	4C	83	OE	20	BO	OE	E6	1B
0CE0	EA	A9	09	4C	2C	OC	A5	32	C9	08	FO	2D	29	F8	FO	59	OE80	4C	71	OE	A5	3B	30	04	A9	00	85	3B	C6	3B	A5	3B	C9
0CF0	AD	00	08	29	80	DO	17	A9	01	85	35	A5	36	49	FF	18	OE90	EC	DO	05	A9	F3	85	3A	60	C9	BA	DO	08	A9	FF	85	3A
0D00	65	38	85	38	EA	EA	EA	EA	EA	EA	EA	4C	2A	OC	E6	32	OEAO	A9	EF	85	3B	60	A1	E6	A4	A7	8C	AF	1B	AF	E2	1E	8F
0D10	A5	32	29	3F	85	32	4C	34	OC	AO	03	66	33	66	34	88	OEB0	A5	1F	45	1B	29	80	FO	03	20	BB	OE	A5	1B	85	1F	A9
0D20	DO	F9	EA	EA	EA	A5	34	38	E9	C2	49	FF	18	65	38	C9	OECO	OA	85	1C	A9	FF	85	1D	A9	FF	85	1E	C6	1E	DO	FC	C6
0D30	7A	10	05	A9	7A	4C	3E	OD	C9	86	30	02	A9	86	85	38	OEDO	1D	DO	F4	C6	1C	DO	EC	60	EA	EA	EA	EA	EA	EA	EA	EA
0D40	A9	00	85	33	85	34	4C	OE	OD	18	A5	34	65	31	85	34	OEE0	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA
0D50	A5	33	69	00	85	33	4C	OE	OD	25	3A	85	30	8D	00	08	OEFO	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA	EA
0D60	A9	40	05	1A	A9	00	85	42	85	43	AD	00	08	29	40	FO	OF00	A5	30	29	20	FO	10	A5	30	29	10	FO	05	A9	01	4C	18
0D70	F9	A2	06	E6	41	FO	2D	CA	DO	F9	A6	41	AO	30	A5	30	OF10	OF	A9	02	4C	18	OF	A9	03	A6	35	FO	03	18	69	03	8D
0D80	49	01	85	30	8D	00	08	86	10	C6	10	DO	FC	A9	10	85	OF20	02	08	09	80	8D	02	08	60	FF	FO	11	A9	OE	85	FD	88
0D90	09	C6	09	DO	FC	AD	00	08	29	40	FO	11	88	DO	DF	CA	OFFC	00	OC														

Bild 6. Hex-Dump des EPROM-Inhaltes

Herbert Müller

EMUF lernt sprechen

Die Kombination eines erschwinglichen Sprachsynthese-Bausteines mit dem EMUF ermöglicht den Aufbau eines intelligenten Sprachausgabegerätes zu einem wohl unschlagbar günstigen Preis. Dabei kann sich der EMUF-Talker in seinen Fähigkeiten durchaus mit entsprechenden Fertigergeräten [1] messen.

Das Sprachsynthese-IC SC-01 der Firma Votrax wurde bereits in [2] vorgestellt. Es erzeugt 61 verschiedene Laute, die durch einen 6-Bit-Parallelcode abgerufen werden können. Aus diesen Phonemen kann jedes beliebige Wort aufgebaut werden, so daß der Sprachschatz nicht auf ein festes Vokabular beschränkt ist. Zur Umsetzung eines Klartextes in Phoneme bedarf es allerdings einer gewissen „Intelligenz“ – eine Aufgabe, die der EMUF-Minimalcomputer [3] übernimmt. Er analysiert die seriell eingegebenen Worte, sendet die passenden Phonemenkombinationen an den

Sprachsynthesizer und steuert Stimmlage und Betonung. Der Benutzer kann den zu sprechenden Klartext direkt in ASCII über eine serielle Schnittstelle an den EMUF-Talker wie an einen Drucker ausgeben. Da auch zusätzliche Steuerzeichen in ASCII übertragen werden, dürfte sich der EMUF an nahezu allen Mikrocomputersystemen betreiben lassen.

Die Schaltung

Die zur Ansteuerung des Phonemensynthesizers erforderliche Schaltung ist in Bild 1 dargestellt. Sie enthält die Puffe-

rung und Pegelwandlung der Ein- und Ausgänge des SC-01, die Taktaufbereitung sowie Nf-Verstärker und Spannungsstabilisierung für 5 V.

Der 6-Bit-Phonemcode wird über die EMUF-Portleitungen PB0 bis PB5 ausgegeben und vom SC-01 bei einer negativen Flanke an PB6 in den internen Speicher übernommen. Der angewählte Laut erscheint am Nf-Ausgang, wobei nach Ablauf der zugeordneten Zeitdauer das Ready-Signal am Porteingang PB7 auf logisch null springt, um den nächsten Phonemcode anzufordern. Da der Status der Datenleitungen nur im Augenblick der Strobe-Flanke abgefragt wird, kann in der übrigen Zeit hierüber die Betonung des zu sprechenden Phonemes ausgegeben werden, ohne daß ein zusätzlicher Zwischenspeicher benötigt wird.

Der SC-01 besitzt zwei Betonungsbits, an denen sich vier verschiedene Tonlagen einstellen lassen. Der praktische Einsatz zeigte jedoch, daß der Unterschied der Stimmhöhen für eine Betonung innerhalb des Satzes zu groß erscheint. Daher wurden die eigentlichen Stimmhebungen und -senkungen durch Veränderung der SC-01-Taktfrequenz realisiert [4]. Diese wird aus dem 1-MHz-Takt des EMUF durch einen Binäraten-Multiplizierer (IC 2) abgeleitet, der mit vier Bit der Datenleitungen programmiert wird.

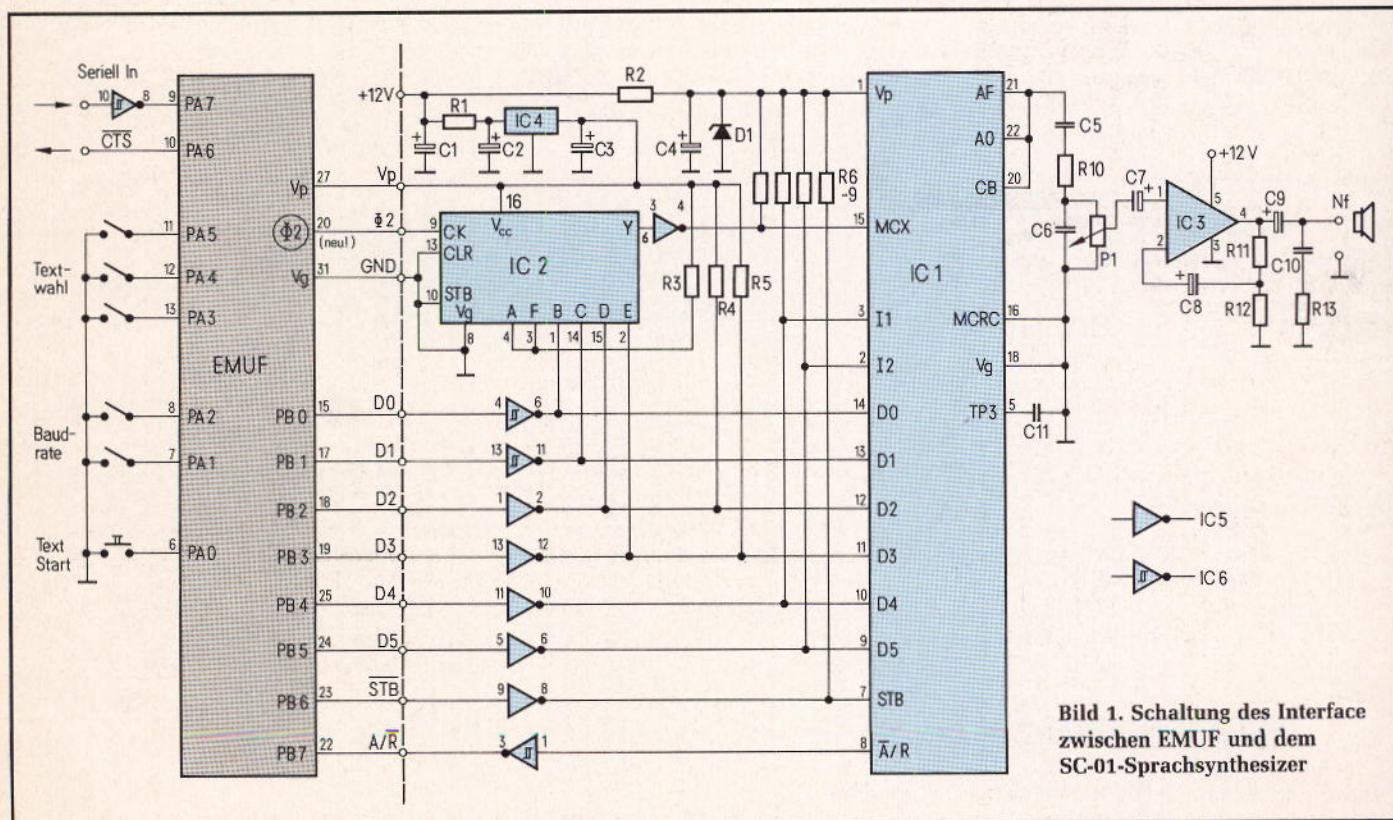


Bild 1. Schaltung des Interface zwischen EMUF und dem SC-01-Sprachsynthesizer

Hierdurch kann die Taktfrequenz in Stufen von 31,25 kHz um den mittleren Wert von etwa 700 kHz variiert werden. Zugleich ergibt sich daraus die Möglichkeit, die Sprachgeschwindigkeit zu beeinflussen. Die Betonungsbits selbst werden über die Datenleitungen PB4 und PB5 gesetzt und erlauben die Auswahl verschiedener Stimmen.

Mini-Platine

Für die beschriebene Schaltung wurde von Christoph Kessler eine kleine Zusatzplatine entworfen, die mit drei Schrauben und Abstandshaltern oberhalb des freien Raumes auf der EMUF-Karte montiert werden kann (Bild 2). Bild 3 zeigt den Bestückungsplan, die Tabelle nennt die erforderlichen Bauelemente. Um mit einer einzigen Betriebsspannung von +12 V auszukommen, ist ein +5-V-Spannungsregler vorgesehen, der auch den EMUF versorgt. Spannungsregler und Nf-Verstärker sitzen auf einem gemeinsamen Kühlkörper, ein zusätzlicher 5-W-Widerstand übernimmt einen Teil der Verlustwärme und kann ebenfalls mit dem Kühlblech verbunden werden. Lautstärkeregel und Lautsprecherbuchse sitzen direkt auf der Platine, um den Aufbau zu erleichtern. Für den

SC-01 sollte eine 22polige Fassung eingelötet und das IC erst zum Schluß eingesetzt werden. Die Verbindungen zum EMUF werden über Flachbandkabel an

Stückliste für den EMUF-Talker

IC 1	Votrax SC-01-A
IC 2	SN 7497
IC 3	TDA 2002, TDA 2003
IC 4	7805
IC 5	SN 7406
IC 6	MC 1489 oder SN 75189
D 1	ZF 15
R 1	12 Ω , 5 W
R 2	33 Ω
R 3-R 5	2,2 k Ω
R 6-R 9	10 k Ω
R 10	3,3 k Ω
R 11	220 Ω
R 12	2,2 Ω
R 13	1 Ω
P 1	10 k Ω log.
C 1, C 2, C 4	100 μ F, 16 V
C 3	22 μ F
C 5, C 10	100 nF
C 6, C 11	10 nF
C 7	10 μ F, 16 V
C 8	470 μ F, 16 V
C 9	1000 μ F, 16 V
Kühlkörper:	97 $\times$ 37,5 $\times$ 25 mm, z. B. Alutronic PR 90
Lautsprecherbuchse:	Hirschmann Lb 2 h

dessen Steckerleiste geführt. Der CPU-Takt des EMUF ist über eine Drahtverbindung vom Pin 6 des 7400 auf der EMUF-Karte an einen freien Stift der Steckerleiste zu legen (z. B. Stift 20).

Die serielle Schnittstelle

Die Eingabe der zu sprechenden Texte erfolgt über eine serielle Schnittstelle als ASCII-Großbuchstaben, wobei zur Parallel-Umwandlung eine ähnliche Routine wie in [5] Verwendung findet. Der serielle Eingang wird über einen Pegelwandler MC 1489 angesteuert, um auch V.24-Signale anlegen zu können. Der Rückmeldeausgang PB6 signalisiert, ob der EMUF bereit ist, das nächste Zeichen zu empfangen, oder noch mit der Sprachausgabe beschäftigt ist. Die serielle Ausgaberroutine muß daher in einer Warteschleife diese Leitung abfragen und darf erst bei logisch null das nächste ASCII-Zeichen senden. Die Polarität ist hierbei so gewählt, daß zur Umwandlung in ein V.24-gerechtes CTS-Signal bei Bedarf ein Pegelwandler (MC 1488, SN 75188) eingesetzt werden kann.

Die Übertragungsgeschwindigkeit wird an den beiden Portbits PA1 und PA2 auf einen der folgenden Werte eingestellt:

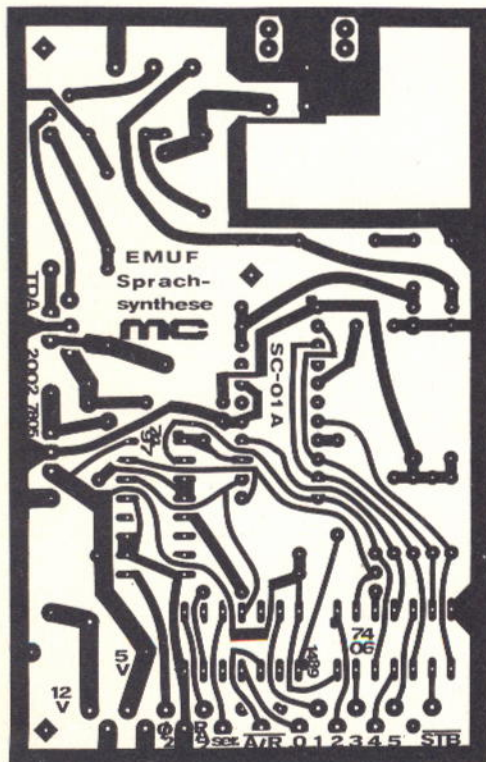


Bild 2. Platinenlayout der Zusatzplatine. Die Platine kann direkt auf die EMUF-Karte aufgesetzt werden

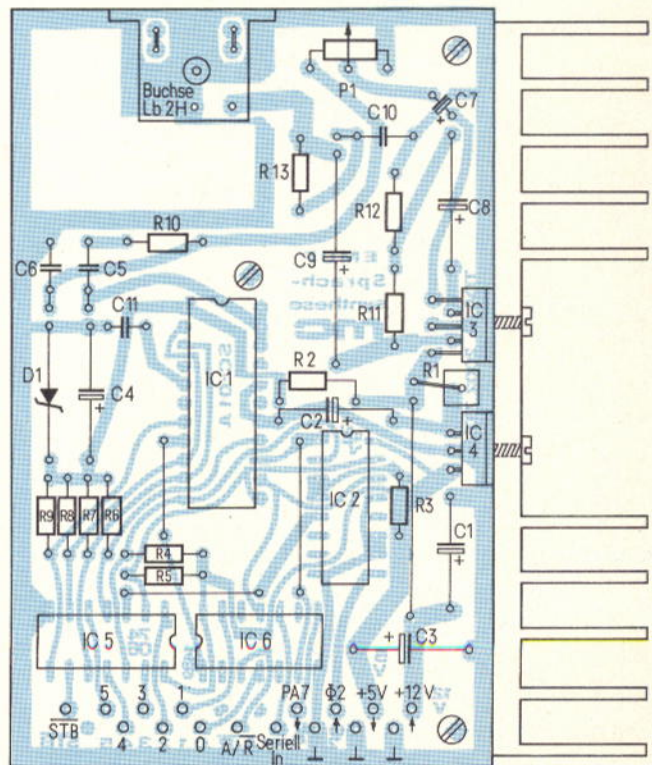


Bild 3. Bestückungsplan der Zusatzplatine

PA2	PA1	
0	0	1200 Baud
0	1	2400 Baud
1	0	4800 Baud
1	1	9600 Baud

Die Baudrate wird intern bei Reset und nach jeder Ausgabe eines gespeicherten Standardtextes durch Abfrage der beiden Eingänge festgelegt. Der Programmteil für die serielle Eingabe steht an den Adressen 0F80...0FC3 im EPROM, die zugehörige Baudraten-Tabelle bei 0FC4...0FCB.

Die Software im EMUF

Die empfangenen ASCII-Zeichen speichert der EMUF zunächst im eigenen RAM-Bereich, bis er anhand eines Zwischenraumes oder Wagenrücklaufes das Wortende erkennt. Daraufhin werden Sonderfunktionen ausgefiltert, das gesamte Wort analysiert, in Phoneme gewandelt und diese „gesprochen“. Für bestimmte Buchstabenkombinationen sind in einer Tabelle des Programmes (0E32...0EA2) besondere Ausspracheregeln vorgegeben, wobei die zugehörigen Phonem-Codes durch das gesetzte achte Bit gekennzeichnet sind. Ein Bindestrich am Anfang oder Ende der Buchstabenfolge bedeutet, daß die Aussprache nur am Wortanfang oder -Ende bzw. bei Silbentrennung Anwendung findet. Erst wenn das Programm hier nicht fündig wird, setzt es die Buchstaben einzeln anhand einer weiteren Tabelle (0E12 bis

0E31) in Phoneme um. Der Phonem-Code für die zu hex 5B...5F gehörenden ASCII-Zeichen ist in der Tabelle bei 0E2D...0E31 einzusetzen, z. B.: hex 2E = 'Ä', hex 3A = 'Ö', hex 36 = 'Ü', hex 03 = 47 ms Pause, hex 3E = 155 ms Pause.

Die Aussprache für Ziffern und Sonderzeichen sowie zum Buchstabieren ist in der dritten Tabelle (0EA3...0F7F) bereits im Klartext aufgeführt, wobei hier die umzuwandelnden ASCII-Zeichen durch das achte Bit gekennzeichnet sind. Die Tabellen wurden so strukturiert, daß sie leicht abgeändert oder erweitert werden können.

Die Texteingabe

Prinzipiell lassen sich vom EMUF-Talker beliebige deutsche Texte sprechen. Der einfache Umsetzungsalgorithmus ergibt bereits eine recht beachtliche „Tref-ferquote“, stolpert aber natürlich an einigen Stellen. Sofern eine Überarbeitung des Originaltextes möglich ist, lassen sich durch geringe Veränderungen der orthografischen Schreibweise die meisten Aussprachefehler leicht beheben. Häufig genügt schon ein Silbentrennungszeichen wie etwa bei „be-inhalten“ oder bei „be-stellen“.

Nicht ganz vermeidbar ist der amerikanische Akzent, da die Phoneme der amerikanischen Sprache angepaßt sind. Um ihn etwas zu reduzieren, wird bei der Umwandlung teilweise auf dem Deutschen ähnliche Ersatzlaute zurückge-

griffen und beispielsweise statt „arbeiten“ ein „aebeiten“ gesprochen.

Die richtige Betonung

Als Steuerzeichen zur Einführung einer Betonung dienen die üblichen Satzzeichen. Hierdurch erhält auch ein unveränderter Text zumindest am Satzende eine Stimmhebung oder -Senkung und klingt nicht zu monoton. Zusätzliche Betonungen lassen sich leicht durch das Einfügen weiterer Satzzeichen, auch innerhalb eines Wortes, nach eigenem Geschmack definieren. Die einzelnen Zeichen bewirken folgende Betonung der letzten vorangegangenen (!) Vokalkombination:

- . 2 Stufen tiefer
- ,: 1 Stufe tiefer
- ? 1 Stufe höher
- ! 2 Stufen höher

Beispiel: „DAS IST, SYN.THETISCHE SPRA?CHE“

Eine Betonung erfolgt nur, wenn zwischen Wortende und Satzzeichen kein Leerraum eingefügt ist. Die nötigen Betonungs-Datentabellen stehen bei 0DFF...0E11.

Sonderfunktionen

#: Buchstabieren

Enthält ein Wort das Zeichen # oder eine Ziffer, wird es in seiner Gesamtheit buchstabiert. Auf diese Weise können Eigennamen und andere wichtige Worte bzw. Amateurfunk-Rufzeichen übermit-

OC00	78	D8	A2	FF	9A	A9	7F	8D	03	08	A9	40	8D	01	08	A9
OC10	15	85	00	85	01	A9	3E	20	E4	0D	20	C7	0D	4C	3D	0C
OC20	20	B3	0F	A9	00	8D	00	08	AD	00	08	29	01	F0	0E	2C
OC30	00	08	30	F4	20	80	0F	20	79	0C	4C	23	0C	A9	40	8D
OC40	00	08	A9	CC	85	0A	A9	0F	85	0B	A0	00	AD	00	08	4A
OC50	4A	4A	29	07	AA	F0	10	B1	0A	F0	C5	08	0D	02	E6	0B
OC60	C9	0D	D0	F3	CA	D0	F0	B1	0A	F0	B5	20	79	0C	C8	D0
OC70	02	E6	0B	C9	0D	D0	F0	F0	A7	85	09	8A	48	98	48	A5
OC80	09	29	7F	A6	05	E0	40	90	03	4C	1F	0D	C9	21	90	2D
OC90	95	0E	E6	05	C9	24	F0	1F	C9	40	F0	1B	C9	23	F0	08
OCA0	C9	3A	B0	0C	C9	30	90	08	24	04	30	04	A9	80	85	02
OCB0	68	A8	68	AA	A5	09	60	A9	80	85	04	D0	F3	A9	2D	95
OCC0	0E	E0	02	F0	04	24	02	10	56	86	06	B5	0E	95	1E	CA
OCDO	D0	F9	20	C7	0D	A2	01	B5	1E	09	80	A0	00	D9	A3	0E
OCE0	F0	05	C8	D0	F8	F0	0C	C8	B9	A3	0E	30	06	20	79	0C
OCFO	4C	E7	0C	A9	20	20	79	0C	E8	E4	06	90	DA	B0	B1	A5
OD00	03	49	80	85	03	4C	87	0D	E8	B5	0E	29	03	0A	0A	0A
OD10	0A	85	00	E8	B5	0E	29	0F	05	00	85	00	4C	87	0D	A2
OD20	00	B5	0E	C9	40	F0	D8	C9	24	F0	DD	86	07	A5	00	18
OD30	6D	06	0E	85	01	CA	20	9E	0D	30	FB	20	9E	0D	10	FB
OD40	20	9E	0D	30	FB	A6	07	B5	0E	24	03	10	06	20	DD	0D
OD50	4C	87	0D	A0	00	84	08	B9	32	0E	30	36	D5	0E	D0	04
OD60	E8	C8	D0	F3	A6	07	C8	B9	32	0E	10	FA	C8	B9	32	0E
OD70	30	FA	84	08	C9	00	D0	E4	B5	0E	C9	41	90	09	29	3F
OD80	A8	B9	12	0E	20	DD	0D	E8	E4	05	90	95	20	C7	0D	4C
OD90	B0	0C	CA	B9	32	0E	10	EF	20	DD	0D	C8	D0	F5	E8	B5
ODA0	0E	C9	2D	F0	1D	A0	06	D9	FF	0D	F0	0E	88	10	F8	A0
ODB0	04	D9	0D	0E	F0	03	88	10	F8	60	B9	06	0E	18	65	00
ODC0	85	01	68	08	4C	45	0D	A9	03	20	DD	0D	A9	00	85	02
ODD0	85	03	85	04	A9	2D	85	0E	A9	01	85	05	60	29	3F	2C
ODE0	02	08	30	FB	09	40	49	3F	8D	02	08	49	40	8D	02	08
ODFO	09	40	8D	02	08	A5	01	09	40	49	3F	8D	02	08	60	20
OE00	2E	3B	2C	3A	3F	21	02	00	01	01	01	03	04	41	45	49
OE10	4F	55	03	15	0E	19	1E	06	1D	1C	1B	3C	22	19	18	0C
OE20	0D	35	25	19	2B	1F	2A	28	0F	2D	1F	3C	12	03	3E	3E
OE30	3E	3E	45	49	88	A9	41	49	88	A9	41	45	55	B5	A9	41
OE40	45	AE	4F	45	BA	55	45	B6	B6	45	55	B5	A9	49	45	AC
OE50	41	55	B0	B7	2D	53	54	91	AA	53	43	48	91	53	50	91
OE60	A5	43	48	53	99	9F	43	48	90	4E	47	94	9B	50	48	9D
OE70	51	55	99	9B	AD	41	43	48	95	9B	9B	4F	43	48	96	9B
OE80	9B	55	43	48	B7	9B	9B	45	52	57	AF	88	8F	45	52	85
OE90	88	58	99	9F	5A	AA	9F	41	52	95	80	4C	4C	98	4D	4D
OEA0	8C	00	00	B0	4E	55	4C	B1	45	49	4E	5A	B2	5A	57	4F
OEB0	48	B3	54	52	45	49	B4	56	49	45	52	B5	46	55	45	4E
OEC0	46	B6	53	45	43	48	53	B7	53	49	45	42	45	4E	B8	41
OED0	43	48	54	B9	4E	45	55	4E	C1	41	41	48	C2	42	45	45
OEE0	48	C3	54	53	45	45	48	C4	44	44	45	45	48	C5	45	45
OEF0	48	48	C6	45	46	46	C7	47	47	47	45	45	48	C8	48	41
OF00	41	C9	49	49	48	CA	4A	4F	54	54	48	CB	4B	41	41	48
OF10	CC	45	4C	4C	CD	45	4D	4D	CE	45	4E	4E	CF	4F	4F	48
OF20	D0	50	45	45	48	D1	4B	55	48	D2	45	52	52	D3	45	53
OF30	53	D4	54	45	45	48	D5	55	48	D6	56	41	55	48	D7	57
OF40	45	45	48	D8	49	4B	53	53	D9	55	45	50	53	49	4C	4F
OF50	4E	DA	5A	45	54	54	48	AA	4D	41	4C	AB	55	4E	4A	AC
OF60	4B	4F	4D	41	AD	53	54	52	49	43	48	AE	50	55	4E	4B
OF70	54	AF	53	54	52	49	43	48	BD	47	4C	45	49	43	48	FF
OF80	A2	09	A5	0C	4A	20	A9	0F	A9	00	48	20	A2	0F	AD	00
OF90	08	0A	68	6A	CA	D0	F3	A0	40	8C	00	08	48	20	A2	0F
OFA0	68	60	2C	15	08	10	FB	A5	0C	86	07	A6	0D	9D	14	08
OFB0	A6	07	60	AD	00	08	29	06	AA	BD	C4	0F	85	0C	BD	C5
OFD0	0F	85	0D	60	65	01	31	01	B5	00	51	00	4F	4B	45	2D
OFD0	49	5D	0D	45	4D	55	46	5D	0D	53	50	52	41	43	48	45
OFE0	5D	0D	54	45	53	54	5D	0D	45	52	4F	52	5D	0D	50	41
OFF0	55	53	45	5D	0D	52	49	53	45	54	5D	0D	00	0C	FF	FF

Bild 4. Listing des EMUF-Programms im EPROM

telt werden. In dieser Betriebsart spricht der EMUF-Talker auch einige Sonderzeichen, die sonst einfach ausgeblendet werden. Die Datentabelle mit Phonemen aller Zeichen steht an den Adressen 0EA3...0F7F.

@: Phonemcode

Zur Synthese der deutschen Sprache findet nur ein Teil der Phoneme aus dem SC-01-Repertoire Verwendung. Die übrigen Laute sind jedoch über einen Sonderbefehl ebenfalls anwählbar. Auf diese Weise kann der Anwender selbst experimentieren und insbesondere auch englische Worte zusammensetzen. Das Masterspace-Zeichen bewirkt, daß die darauffolgenden ASCII-Zeichen als 6-Bit-Phonem-Codes interpretiert werden, wobei sich die Zuordnung aus der Tabelle des SC-01-Datenblatts entnehmen läßt. Ein zweites Masterspace im Wort oder ein Zwischenraum beendet die Phonemsequenz.

Ein Beispiel: „The Talker“ wird „@81 T@SYA+“.

\$XX: Einstellung der Stimmlage

Der Klang der synthetischen Stimme und die Sprachgeschwindigkeit lassen sich ebenfalls über die serielle Schnittstelle programmieren. Hierzu dient das Dollarzeichen \$, auf das zwei Ziffern als Steuercode folgen müssen. Der bei Reset initialisierte Wert ist \$15. Die erste Ziffer ergibt die an den beiden Betonungseingängen des SC-01 angelegte Bitkombination und kann von 0 bis 4 variiert werden. Mit der zweiten Ziffer wird die aus dem CPU-Takt heruntergeteilte Taktfrequenz des Sprachsynthesizers eingestellt. Man kann damit sowohl Tonhöhe als auch die Sprachgeschwindigkeit einstellen. In bescheidenem Rahmen besitzt der EMUF-Talker also auch musikalische Fähigkeiten, wie ein kleines Beispiel zeigt: Den Anfang von „Happy Birthday“ singt er, wenn man ihm diese Textfolge sendet: \$33 HAAEPP\$25II \$34BOEOER@99@25DEE-I \$37T@677777 \$36J@677777

Vorprogrammierte Standardtexte

Neben der seriellen Eingabe besteht auch die Möglichkeit, bis zu acht verschiedene Standardtexte abzurufen, die fest im EPROM ab 0FCC gespeichert sind. Diese kann man in der gleichen Weise im Speicher ablegen wie bei serieller Eingabe, so daß eine umständliche Übersetzung in Phoneme entfällt. Die einzelnen Texte werden durch ein Wagenrücklauf-Zeichen voneinander getrennt (hex 0D), das Tabellenende ist mit hex 00 gekennzeichnet. Die Nummer

des zu sprechenden Textes (0...7) legt man als 3-Bit-Code an die EMUF-Eingänge PA 3...5. Die Sprachausgabe wird über eine Taste gestartet, der den Port PA0 mit Masse verbindet. Ebenso meldet sich der EMUF-Talker beim Einschalten der Betriebsspannung bzw. bei Reset mit dem gerade gewählten Text. Das Listing enthält ab 0FCC die Texte für „Okey“, „EMUF“, „Sprache“, „Test“, „Error“, „Pause“ und „Reset“. Da aufgrund der Länge des Programmes in einem KByte nur noch Platz für einige Worte bleibt, kann bei Bedarf der EPROM-Adressenraum des EMUF leicht auf 2 KByte erweitert werden [6]. Hierzu muß man nur die Adressenleitung A 11 der EMUF-CPU (Pin 16) von Pin 9 des 7400 abtrennen und statt dessen direkt mit Pin 19 eines 2716-EPROMs verbinden. Den Programmstart legt man in diesem Fall an die Adresse *=\$400 (EPROM-Adresse \$000), den Anfang der Texttabelle auf *=\$C00 (EPROM-Adresse \$400), und den Resetvektor bei *=\$FFC (EPROM-Adresse \$7FC) richtet man auf \$0400.

Nach dieser Änderung steht nun ein ganzes Kilobyte für fest gespeicherte

Texte zur Verfügung, wodurch der EMUF-Talker auch für Anwendungen ohne Fremdtexteingabe interessant ist. Allerdings ist eine Neuassemblierung des Programms für den neuen Adressenbereich erforderlich.

Die Bauelemente, insbesondere der Sprachsynthese-Baustein SC-01 von Votrax, können von der Fa. Heninger Digital-Service, Landwehrstraße 39, 8000 München 2, bezogen werden. Der SC-01 ist darüber hinaus auch bei der Votrax-Vertretung Dr. Otto Sosluty, Reissstraße 16, 6200 Wiesbaden, erhältlich.

Literatur

- [1] Hofer, R.: Sprachausgabe mit Komfort. mc 1982, Heft 11, S. 40.
- [2] Plate, J.: Sprachausgabe für Tischcomputer. mc 1982, Heft 11, S. 44.
- [3] Feichtinger, H.: Mädchen für alles. mc 1981, Heft 2, und mc-EMUF-Sonderheft.
- [4] Ciarcia, S.: Build the Microvox Text-to-Speech-Synthesizer. Byte Sept./Okt. 1982.
- [5] Hofer, R.: Standardschnittstellen für Schreibmaschinendrucker. EMUF-Sonderheft 1982, S. 18.
- [6] Müller, K.: Kompatible EMUF-Erweiterung. mc 1982, Heft 6, S. 59.

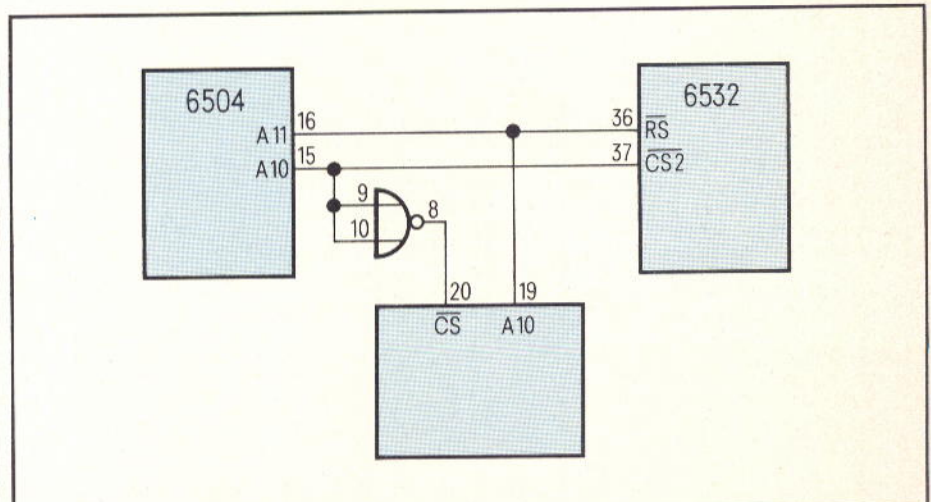
Kompatible EMUF-Erweiterung

Nach Durchführung der vorgeschlagenen Änderung im Bild belegt das EPROM die Adressen 400...7FF und C00...FFF. Beim „Schießen“ des EPROMs wird der Bereich C00...FFF auf den EPROM-Adressen 400...7FF abgelegt.

Die Realisierung dieser Änderung ist sehr einfach. An dem NAND-Gatter, dessen Eingänge auf A10 und A11 liegen,

wird die Verbindung A11 nach Pin 9 des NAND aufgetrennt. Pin 9 wird stattdessen mit einer kurzen Blankdrahtbrücke mit Pin 10 verbunden. Das NAND-Gatter arbeitet jetzt als Inverter für A10. Jetzt ist mit einer kurzen isolierten Drahtbrücke nur noch A11 (Pin 16 am 6504) mit Pin 19 (A10) des 2716 zu verbinden.

Karlheinz Müller



Erweiterung des EPROM-Bereichs auf 2 KByte beim mc-EMUF

Fritz-B. Raab

TRS-80 steuert Typenrad-EMUF

Das im EMUF-Sonderheft auf Seite 46 von Rudolf Hofer vorgestellte Interface für die Typenradschreibmaschine Praxis 30/35 von Olivetti arbeitet prinzipiell mit jedem Rechner zusammen, der ein V24-Interface besitzt. Der TRS-80 kann mit einer Standard-RS-232-C-Karte (ohne Expansion-Interface mit einer „TRS-232“-Karte und anderer Software) ausgerüstet werden, die im wesentlichen dieselben Signale zur Verfügung stellt. Nur die Pegel sind etwas anders, können jedoch problemlos angepaßt werden.

Der Autor besitzt einen TRS-80 mit RS-232-Schnittstelle und hat, nachdem er das EMUF-Interface aufgebaut hatte, den von Tandy im Handbuch der Karte vorgeschlagenen Printer-Treiber mit einem Handshake versehen sowie einige an-

dere Änderungen durchgeführt. Das in (Bild 1) abgedruckte Listing wurde mit Hilfe des Treibers selbst angefertigt. Die Zeilen, welche mit ;* gekennzeichnet sind, wurden zusätzlich zum Originaltreiber eingefügt.

Für die Handshake-Leitung kann Pin 5, 6, 20 oder 22 des Standard-DP25-Stekkers verwendet werden. Diese entsprechen den Bits 7, 6, 5, 4 des „Modem Status Register“ 0E8H. Der Autor verwendete die Leitung „Data Send READY“ (Bit 6, siehe Zeile 460). Die Schleife LOOP ist nötig, da das Handshake-Signal zu schnell für den USART auf Low zurückfällt und dadurch jedes zweite Zeichen verloren ging. Die Schleifendauer ist etwa 13 ms. Am Anfang des Programms wird der Printer-Device-Control-Block auf die Startadresse des neuen Treibers gesetzt. Die Break-Routine (Zeile 510) führt nach jedem gedruckten Zeichen zum Basic-Interpreter zurück.

In Bild 2 ist dargestellt, wie der Pegel der Datenleitung Pin 2 bzw. 3 (je nach eingestelltem Mode) an den EMUF angepaßt wird. Dabei wird das Signal gleichzeitig invertiert. Die Handshake-Leitung kann direkt an die RS-232-Schnittstelle gelegt werden, da sich nach entsprechenden Versuchen herausgestellt hat, daß der EIA-TTL-Wandler MC 1489 der RS-232-Karte, der -15...+15V verträgt, sauber bei 1,13 V schaltet, und sich dabei wie ein gewöhnliches TTL-Gatter verhält.

Bild 1. Listing des Printertreibers zur Steuerung des EMUF-Interface mit einem TRS-80 Modell I

```

00100 ;*****
00110 ;PRINTER-TREIBER FÜR TRS80 RS232-EMUF-PRAXIS 30/35
00120 ;FRITZ-B. RAAB DF2RF/A BOESSNERSTR.9/129 8400 REGENSBURG
00130 ;V 1.1 10.MAI 1982 /48K
00140 ;*****
4025 00150 ORG 04025H ;* ÄNDERUNG DES DRUCKER DCB
4025 02 00160 DEFB 02H ;*
4026 00FF 00170 DEFW INIT ;* AUF JEWEILIGEN ENTRY POINT
FF00 00180 INIT EQU OFF00H ;FÜR 32K:OBFOOH; FÜR 16K:07FOOH
FF00 00190 ORG INIT
FF00 E5 00200 PUSH HL ;REGISTER ALLE
FF01 C5 00210 PUSH BC ;IN STACK
FF02 F5 00220 PUSH AF ;RETEN
FF03 3A63FF 00230 LD A,(FLAG) ;IST DER USART SCHON
FF06 FE01 00240 CP 01H ;INITIALISIERT ?
FF08 2820 00250 JR Z,RESTOR ;WENN JA, ÜBERSPR. START
FF0A 3E01 00260 LD A,01H ;START FLAG WIRD
FF0C 3263FF 00270 LD (FLAG),A ;GESETZT.
FF0F D3E8 00280 OUT (0E8H),A ;USART RESET !
FF11 DBE9 00290 IN A,(0E9H) ;DIL-SCH. AUS RS-BOARD LESEN
FF13 E6F8 00300 AND 0F8H ;EIN KLEIN WENIG
FF15 F604 00310 OR 04H ;HEX RECHNEN.
FF17 3262FF 00320 LD (SWTIMG),A ;DAS ERGEBNIS AUFHEBEN
FF1A D3EA 00330 OUT (0EAH),A ;USART CONTROL REG. LADEN
FF1C DBE9 00340 BAUDST IN A,(0E9H) ;NOCHMAL SCHALTER ABLESEN
FF1E E607 00350 AND 07H ;OBERE 5 BITS SIND EGAL
FF20 215AFF 00360 LD HL,BDTABL ;AUF BAUDRATE TAB ZEIGEN
FF23 0600 00370 LD B,00H ;DAS B REGISTER 0 SETZEN
FF25 4F 00380 LD C,A ;SPRUNGWEITE IN C LADEN
FF26 09 00390 ADD HL,BC ;UND ZUM ZEIGER ADDIEREN.
FF27 7E 00400 LD A,(HL) ;WAS DORT STEHT
FF28 D3E9 00410 OUT (0E9H),A ;INS BAUDRATE REG. SENDEN
FF2A F1 00420 RESTOR POP AF ;FERTIG MIT START
FF2B C1 00430 POP BC ;REGISTERINHALTE AUS STACK
FF2C E1 00440 POP HL ;ZURÜCKHOLEN.
FF2D DBE8 00450 STATIN IN A,(0E8H) ;* MODEM STAT. REG. LESEN

```



```

FF2F CB77      00460      BIT      6,A      ;* OB BUSY HIGH IST
FF31 28FA      00470      JR        Z,STATIN ;* WENN JA,ZURÜCK !
FF33 DBEA      00480      IN        A,(OEAH) ;USART STAT.REG. LESEN
FF35 CB77      00490      BIT      6,A      ;OB LETZTES ZEICHEN
FF37 28F4      00500      JR        Z,STATIN ;SCHON WEG IST ?
FF39 3A4038    00510      LD        A,(3840H) ;* IST BREAK
FF3C CB57      00520      BIT      2,A      ;* GEDRÜCKT ?
FF3E C2191A    00530      JP        NZ,1A19H ;* WENN JA,ZUM BASIC
FF41 79        00540      LD        A,C      ;A MIT ZEICHEN LADEN
FF42 D3EB      00550      OUT      (OEBH),A ;ZEICHEN ZUM USART
FF44 060F      00560      LD        B,OFH    ;* SCHLEIFE FÜR EMUF
FF46 21FF00    00570      LD        HL,OFFH   ;* ERZEUGEN,
FF49 2B        00580      DEC      HL        ;* DAMIT DAS RICHTIGE
FF4A 7C        00590      LD        A,H      ;* TIMING ZUSTANDE
FF4B B5        00600      OR        L        ;* KOMMT.
FF4C C249FF    00610      JP        NZ,LP2    ;*
FF4F 10F5      00620      DJNZ     LOOP      ;*
FF51 FE0D      00630      CP        0DH      ;IST DAS ZEICHEN EIN CR ?
FF53 2004      00640      JR        NZ,RETRN  ;RETURN WENN NICHT
FF55 0E0A      00650      LD        C,0AH    ;EIN LF NACHSCHICKEN
FF57 18D4      00660      JR        STATIN   ;UND ZUR SENDESUBROUTINE
FF59 C9        00670      RETR      RET      ;ANSONSTEN ZURÜCK !
FF5A 22        00680      BDTABL  DEFB     22H ;110 BAUD
FF5B 44        00690      DEFB     44H      ;150 BAUD
FF5C 55        00700      DEFB     55H      ;300 BAUD
FF5D 66        00710      DEFB     66H      ;600 BAUD
FF5E 77        00720      DEFB     77H      ;1200 BAUD
FF5F AA        00730      DEFB     0AAH     ;2400 BAUD
FF60 CC        00740      DEFB     0CCH     ;4800 BAUD
FF61 EE        00750      DEFB     0EEH     ;9600 BAUD
FF62 00        00760      SWTIMG  DEFB     00H ;PLATZ F. DIL-SCH. IMAGE
FF63 00        00770      FLAG    DEFB     00H ;UND HIER IST DAS START-FLAG
FF00          00780      END      INIT
00000 TOTAL ERRORS
32684 TEXT AREA BYTES LEFT

```

```

BAUDST FF1C 00340
BDTABL FF5A 00680 00360
FLAG   FF63 00770 00230 00270
INIT   FF00 00180 00170 00190 00780
LOOP   FF46 00570 00620
LP2    FF49 00580 00610
RETRN  FF59 00670 00640
STATIN FF2D 00450 00470 00500 00660
SWTIMG FF62 00760 00320

```

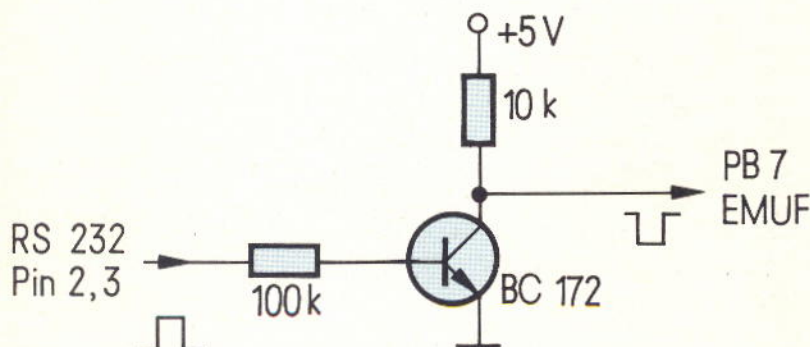


Bild 2. Die zusätzlich notwendige Schaltung nimmt die Pegelwandlung und Invertierung der RS-232-Datenleitung vom TRS-80 zum EMUF-Interface vor. Ohne diese Pegelwandlung könnte der EMUF-Baustein 6532 beschädigt werden

Alles über Mikro-Computer



ELO-Sonderheft
Vom Bit zum Beispiel
Anwendungsbeispiele mit dem Mikrocomputer-System UMS-85.
Gesammelte Beiträge der in der ELO bereits erschienenen Fortsetzungsreihe.
Für hardwareorientierte Mikrocomputer-Anwender (8085-CPU), auch Anfänger.
56 Seiten, 12,- DM



mc-Sonderheft
Ihr Weg zum Computer
Schafft fundiertes Grundlagenwissen über die Computertechnik, Computersprache, Selbstprogrammieren und Anwendungsmöglichkeiten.
Überwiegend neue Beiträge.
Für Computerneulinge.
80 Seiten, 9,80 DM



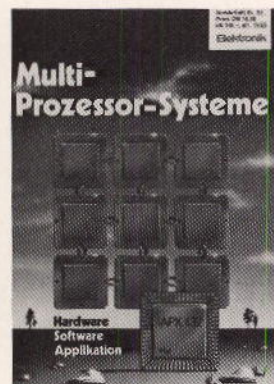
mc-Sonderheft
DAS EMUF-Sonderheft
Aufbau, Programmierung und Anwendung eines Einplatinen-Mikrocomputers für universelle Festprogramm-Anwendung.
Überwiegend neue Beiträge.
Für Anfänger und Fortgeschrittene, 6502-Assembler-Programmierung.
88 Seiten, 18,- DM



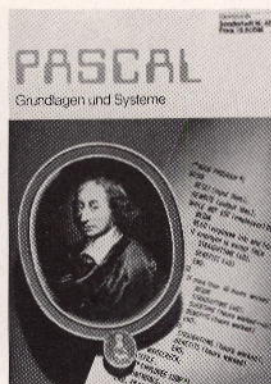
FUNKSCHAU-Sonderheft
mikrocomputer-anwendungen
Programme in Basic, Maschinensprache, Pascal und für Taschenrechner der HP- und TI-Serien.
Die Beiträge sind sonst nirgendwo erschienen.
Für alle Computer-Besitzer und solche, die es werden wollen.
80 Seiten, 15,60 DM



ELEKTRONIK-Sonderheft
Daten-Kommunikation
Alle wesentlichen Aspekte der Datenkommunikation einschließlich der Datenfernübertragung.
Überarbeitete Beiträge aus der ELEKTRONIK.
Für Einsteiger und „alte Hasen“
123 Seiten, 18,- DM



ELEKTRONIK-Sonderheft
Multi-Prozessor-Systeme
z. B.: Grundlagen der Multi-Mikroprozessor-Systeme, Technischer Stand, Leistungsmerkmale, Entwicklungssysteme.
Überwiegend bewährte Beiträge aus der ELEKTRONIK.
Für Entwickler, Programmierer und OEM-Anwender.
112 Seiten, 19,50 DM



ELEKTRONIK-Sonderheft
PASCAL
Grundlagen, Programmier-technik, Unterschiede zu anderen Programmiersprachen, Beschreibung von Systemen.
Überarbeitete Beiträge aus der ELEKTRONIK.
Für Mikrocomputer-Entwickler und -Anwender.
64 Seiten, 18,50 DM



ELEKTRONIK-Sonderheft
Software-Werkzeuge
Werkzeuge, mit deren Hilfe die Produktion von Software in das Stadium der Automatisierung treten kann.
Überwiegend bewährte Beiträge aus der ELEKTRONIK.
Für Mikrocomputer-Entwickler.
112 Seiten, 21,- DM

Sonderhefte zum Thema Mikro-Computer

Für Einsteiger, Elektroniker und Programmierer sind diese Sonderhefte eine kompakte Informationsquelle zu einzelnen Spezialbereichen.

Bezugsmöglichkeiten

Bei allen Bahnhofsbuchhandlungen, beim Elektronik-Fachhandel, bei größeren Zeitschriftenverkaufsstellen, in Buchhandlungen oder direkt beim Franzis-Verlag gegen

- Voreinzahlung des genannten Betrages zzgl. 2,- DM Porto auf unser Postscheckkonto München Nr. 813 75-809 mit genauer Nennung des jeweiligen Titels oder
- Zusendung eines Schecks

Franzis-Verlag

Karlstraße 37,
8000 München 2,
Tel. 0 89/51 17-2 39/-3 80
In der Schweiz:
Verlag Thali AG,
CH-6285 Hitzkirch.
In Österreich:
Fachbuch Center Erb,
Amerlingstraße 1,
A-1061 Wien.

Peter Engels

IEC-Centronics-Interface

Will man einen Drucker mit Centronics-Schnittstelle (Parallel-Schnittstelle) an einem Computer betreiben, der, wie etwa der CBM, den IEC-Bus als Ausgang zur Verfügung stellt, so ist ein geeignetes Interface erforderlich. Der mc-Einplatinencomputer EMUF ist mit einem Materialaufwand von rund 100 DM dafür eine preisgünstige Lösung.

Die hier beschriebene EMUF-Applikation ermöglicht es, Drucker von Epson, Itoh (Binder) oder Centronics mit Parallelschnittstelle und Strobe-Eingang an CBM-Computern zu betreiben. Der EMUF wertet die IEC-Bus-Sekundäradressen aus und erlaubt so eine einfache Software-Schriftsteuerung des Druckers.

Bild 1 zeigt die nötige Beschaltung der EMUF-Platine [1]. Zusätzlich ist noch eine kleine TTL-Logik für die Anpassung von Pegeln und Impulslängen vorgesehen (Bild 2), die sich bequem auf der Lochrasterfläche verdrahten läßt, die auf der EMUF-Platine noch frei ist. Der „Trick“ zur Einsparung von EMUF-I/O-Ports besteht darin, daß die Centronics-Datenleitungen mit den Datenleitungen des IEC-Bus parallelgeschaltet werden. Wenn Daten vom IEC-Bus kommen, liegen sie sowohl am EMUF-Port A als auch am Centronics-Bus an. Sendet der EMUF Zeichen zum Drucker, so wird der IEC-Bus über zwei TTL-Schalter-ICs von Drucker und EMUF getrennt. Dazu

wird PB7 auf Low gelegt. Ein Monoflop verlängert ferner das Acknowledge-Signal des Druckers auf den zur Auswertung durch die EMUF-Routine nötigen Wert von 30 µs. PB6 gibt schließlich das Strobe-Signal an den Drucker aus, sobald die Daten zu übernehmen sind.

Die IEC-Bus-Adresse des EMUF wird beim Einschalten fest auf 4 gelegt; dies ist die übliche Druckeradresse beim CBM. Bild 3 zeigt das vollständige Assemblerlisting der EMUF-Software.

Wenn Bit 3 der Sekundäradresse gesetzt ist, wandelt die Routine CONVERT den CBM-Code in ASCII um. SECADR1 prüft bei gesetztem Bit 0 der Sekundäradresse, ob es sich um ein Cursor- oder CBM-

Steuerzeichen handelt und gibt dieses dann als Klartext an den Drucker aus. SECADR2 gibt je nach Sekundäradresse entsprechende Schriftsteuerzeichen an den Drucker:

- Sek.-Adr. +2: Condensed Mode (132 Zeichen/Zeile)
- Sek.-Adr. +8: Enlarged Mode (40 Zeichen/Zeile)
- Sek.-Adr. +0: eingest. Mode beibehalten
- Sek.-Adr. +10: Normalmodus (60 Zeichen/Zeile)

CHR\$ (24) löscht die im Interface eingestellten Schriftflags und stellt Normal-schrift ein.

Drucker-Steuerzeichen sollten nur bei gelöschtem Bit 0 (also Sekundäradressen 0 oder 4) übergeben werden, da diese sonst als Cursortexte erscheinen. Bild 4 zeigt ein Basic-Demonstrationsprogramm.

Programmierte EPROMs mit dem EMUF-Programm können vom Autor (Kreisstraße 29, 5308 Rheinbach) bezogen werden. EMUF-Bausätze und Platinen liefern die Firmen Wirth, r+r und Elektronikladen.

Literatur

- [1] Mädchen für alles: Der Einplatinen-Computer EMUF. mc 1981, Heft 2, oder EMUF-Sonderheft, Franzis-Verlag.
- [2] Klein, R.-D.: EMUF bringt Strichcode zum IEC-Bus. mc 1981, Heft 3, oder EMUF-Sonderheft, Franzis-Verlag.
- [3] Epson: MX-80/82 Operation Manual (Control Codes).

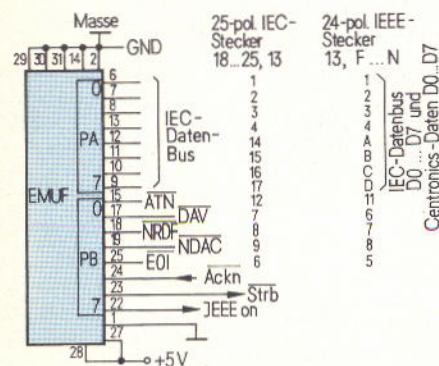


Bild 1. Beschaltung der EMUF-Platine und Verbindungen zum Drucker

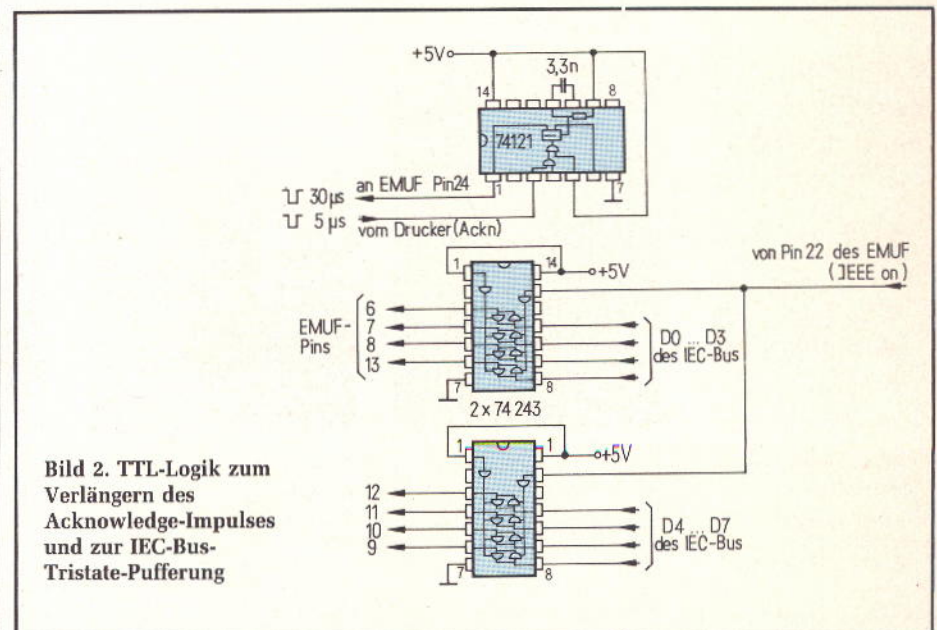


Bild 2. TTL-Logik zum Verlängern des Acknowledge-Impulses und zur IEC-Bus-Tristate-Pufferung

0300 PA	.DE \$800	0C55- 49 FF	0910	EDR #211111111	; INVERTIEREN
0310 PAD	.DE \$801	0C57- 48	0920	PHA PB	; RETTEN
0320 PB	.DE \$802	0C58- AD 02 08	0930	LDA PB	
0330 PBD	.DE \$803	0C5B- B5 0A	0940	STA \$FLAG	
0340 ;		0C5D- A9 C8	0950	LDA #211001000	; DAC
0350 FLAG	.DE \$A	0C5F- 8D 02 08	0960	STA PB	
0360 ZEICH	; IEC FLAG: ATN, EOI	0C62- AD 02 08	0970	LDA PB	
0370 COUNT	; ZMSPEICHER	0C65- 29 02	0980	AND #200000010	
0380 PADR	; HILFSZAEHLER	0C67- F0 F9	0990	BEG LOPB	; DAV HIGH ?
0390 SEC	; DEVICE-NR.	0C69- A9 C0	1000	LDA #211000000	
0400 ;	; SECUNDAER-ADRESSE	0C6B- BD 02 08	1010	STA PB	
0410 ;		0C6E- 68	1020	PLA	
0420 ;		0C6F- 60	1030	RTS	
0430	.DS	0C70- A5 0B	1040 ;	LDA \$ZEICH	
0440	.BA \$0C00	0C72- 29 0F	1050	AND \$0F	
0450	.MC \$7C00	0C74- C5 0D	1060	CMP \$PADR	; TEST OB DRUCKER-DEVICE
0460 ;		0C76- D0 B3	1080	BNE MAIN	; DRUCKER NICHT ANGESPROCH
0470 START	LDA #200000000	0C78- A5 0E	1090	LDA \$SEC	
0480	TXS	0C7A- 29 0A	1100	AND #200001010	
0490	LDA STA PAD	0C7C- B5 0E	1110	STA \$SEC	
0500	LDA #211001100	0C7E- 20 41	1120	JSR GETCHA	
0510	OC08- A9 CC	0C81- B5 0B	1130	STA \$ZEICH	
0520	OC0A- 8D 03	0C83- A5 0A	1140	LDA \$FLAG	
0530	OC0D- A9 C0	0C85- 29 01	1150	AND #200000001	
0540	OC0F- 8D 02 08	0C87- D0 0B	1160	BNE DATEN	
0550	OC12- A9 00	0C89- A5 0B	1170	LDA \$ZEICH	
0560	OC14- B5 0E	0C8B- 05 0E	1180	ORA \$SEC	
0570	OC16- D8	0C8D- B5 0E	1190	STA \$SEC	
0580	OC17- 78		1200 ;		
0590	OC18- A9 04		1210 ;		
0600	OC1A- B5 0D		1220	JSR GETCHA	
0610	OC1C- A9 07		1230	STA \$ZEICH	; ZWSFEICH
0620	OC1E- 20 3A 0D		1240	LDA \$FLAG	
0630	OC21- A9 0A		1250	AND #200000001	
0640	OC23- 20 3A 0D		1260	BEG ERROR	; TEST ATN
0650	OC26- A9 0D		1270	LDA \$ZEICH	
0660	OC28- 20 3A 0D		1280 ;		
0670			1290	CMP #18	
0680	OC2B- 20 41 0C		1300	BNE WORK	; CAN-CODE ?
0690	OC2E- B5 0B		1310	LDA \$SEC	; NEIN
0700	OC30- A5 0A		1320	AND #200000101	
0710	OC32- 29 01		1330	STA \$SEC	; SCHRIFTFLAG LOESCHEN
0720	OC34- D0 F5		1340	JMP LOPMAI	; WEITER
0730	OC36- A5 0A		1350 ;		
0740	OC38- 29 60		1360	LDA \$ZEICH	
0750	OC3A- C9 20		1370	JSR SECADR2	
0760	OC3C- F0 32		1380	JSR PRINT	; SEC-ADR AUSWERTEN
0770	OC3E- 4C 2B 0C		1390	LDA \$FLAG	; AN DRUCKER AUSGEBEN
0780 ;			1400	AND #200010000	
0790 ;			1410	BEG FINA	; EOI ?
0800 ;			1420	BNE LOPMAI	; ENDE
0810 ;			1430 ;		; HAUPTSCHLEIFE
0820 ;			1440	JMP MAIN1	
0830	OC41- A9 C4		1450	LDA #14	; ATN GESETZT
0840	OC43- BD 02 0B		1460	JSR PRINT	; DC 4 CODE
0850	OC46- AD 02 0B		1470	LDA #12	; LOESCHT ENLARGED MODE
0860	OC49- 29 02		1480	JSR PRINT	; DC 2 CODE
0870	OC4B- D0 F9		1490	JMP MAIN	; LOESCHT CONDENSED MODE
0880	OC4D- A9 C0		1500 ;		
0890	OC4F- BD 02 0B		1510 ;		
0900	OC52- AD 00 0B				UNTERSUCHT SEC-ADRESSE AUF GESETZTE BITS

Bild 3. Assemblerlisting der EMUF-Software, hier für die Kombination CBM und Epson-Drucker

0CC9- A5 0E	1520 ; UND GIBT ENTSPRECHENDE STEUERZEICHEN	0D31- 90 FB	2150	BCC CONV1
0CCB- 29 02	1530 ; AN DRUCKER AUS.	0D33- C9 E0	2160	CMP #224
0CCD- C9 02	1540 ;	0D35- B0 F7	2170	BCC CONV1
0CCF- D0 05	1550 SECADR2	0D37- 29 7F	2180	AND #7F
0CD1- A9 0F	1560 LDA #SEC	0D39- 60	2190	RTS
0CD3- 20 3A 0D	1570 AND #2		2200 ;	CHARACTER IM ACCU AUF CENTRONICS PARALLEL
0CD6- A5 0E	1580 CMP #2		2210 ;	MIT STROBE AUSGEBEN, AUF ACKN WARTEN
0CDB- 29 08	1590 BNE SECADR8		2220 ;	
0CDA- C9 0B	1600 LDA #0F		2230 ;	
0CDC- D0 05	1610 JSR PRINT		2240 PRINT	PHA
0CDE- A9 0E	1620 AND #8	0D3A- 4B	2250	LDA #201000000 ; IEEE-BUS TRENNEN
0CE0- 20 3A 0D	1630 BNE SECADR4	0D3B- A9 40	2260	STA PB
0CE3- A5 0E	1640 LDA #0E	0D3D- BD 02 0B	2270	LDA #FF
0CE5- 29 04	1650 JSR PRINT	0D40- A9 FF	2280	STA PAD
0CE7- C9 04	1660 AND #04	0D42- BD 01 0B	2290	PLA
0CE9- D0 03	1670 BNE SECADR1	0D45- 68	2300	STA PA
0CEB- A5 0B	1680 LDA #ZEICH	0D46- BD 00 0B	2310	LDA #2000000000
0CED- 60	1690 RTS	0D49- A9 00	2320	STA PB
	1700 ;	0D4B- BD 02 0B	2330	STA PB
	1710 ;	0D4E- AD 02 0B	2340	WAITACKN
	1720 ;	0D51- 29 20	2350	AND #2001000000
	1730 ;	0D53- C9 20	2360	CMP #2001000000
	1740 ;	0D55- F0 F7	2370	BEG WAITACKN
	1750 ;	0D57- A9 C0	2380	LDA #2110000000
	1760 ;	0D59- BD 02 0B	2390	STA PB
	1770 ;	0D5C- A9 00	2400	LDA #00
0CEE- A5 0E	1780 SECADR1	0D5E- BD 01 0B	2410	STA PAD
0CF0- 29 01	1790 AND #1	0D61- AD 02 0B	2420	WAITACKN1
0CF2- C9 01	1800 CMP #1	0D64- 29 20	2430	AND #2001000000
0CF4- D0 2B	1810 BNE CONVERT	0D66- C9 20	2440	CMP #2001000000
0CF6- A2 FF	1820 SECADRO	0D68- D0 F7	2450	BNE WAITACKN1
0CFB- 86 0C	1830 CLOOP	0D6A- 60	2460 ;	
0CFA- E8	1840 INX		2470 ;	
0CFB- BD 00 0F	1850 LDA CURSTAB,X		2480 ;	TABELLE DER CURSOR-ZEICHEN
0CFE- F0 21	1860 BEQ CONVERT		2490 ;	UND DER CURSOR-ABKUEZUNGS-TEXTE
0D00- C5 0B	1870 CMP #ZEICH		2500 ;	
0D02- D0 F6	1880 BNE CLOOP		2510 ;	
0D04- A0 FF	1890 LDY #FF		2520	.BA \$0F00
0D06- C8	1900 CLOOP1		2530	.MC \$7F00
0D07- B9 1D 0F	1910 LDA TEXTTAB,Y	0F00- 07 09 0E	2540 ;	.BY \$07 \$09 \$0E ; BELL TAB TEXT
0D0A- C9 00	1920 CMP #00	0F03- 0F 11 12	2550	.BY \$0F \$11 \$12 ; TOP CDOWN RVS
0D0C- D0 F8	1930 BNE CLOOP1	0F06- 13 14 15	2560	.BY \$13 \$14 \$15 ; HOME DEL DELINE
0D0E- E6 0C	1940 INC #COUNT	0F09- 16 19 1B	2570	.BY \$16 \$19 \$1B ; ERSEND SCRP ESC
0D10- E4 0C	1950 CPX #COUNT	0F0C- 1D	2580	.BY \$1D ; CRIGHT
0D12- D0 F2	1960 BNE CLOOP1		2590	
0D14- C8	1970 CLOOP2	0F0D- 87 89 BD	2600 ;	.BY \$87 \$89 \$8D ; BELL TABC SHRTN
0D15- B9 1D 0F	1980 LDA TEXTTAB,Y	0F10- BE BF 91	2610	.BY \$8E \$8F \$91 ; GRAPHIC BOT CUP
0D18- C9 00	1990 CMP #00	0F13- 92 93 94	2620	.BY \$92 \$93 \$94 ; OFF CLR INST
0D1A- F0 12	2000 BEQ CONV1	0F16- 95 96 99	2630	.BY \$95 \$96 \$99 ; INSLNE ERSBEG SCRDWN
0D1C- 20 3A 0D	2010 JSR PRINT	0F19- 9B 9D FF	2640	.BY \$9B \$9D \$FF ; SHESC CLEFT P1
0D1F- F0 F3	2020 BEQ CLOOP2	0F1C- 00	2650	.BY \$00
	2030 ;	2660	2660	
	2040 ;	2670 ;	2670 ;	
	2050 ;	2680 ;	2680 ;	
	2060 CONVERT	0F1D- 00 42 45	2690	.BY \$00 'BELL'
0D21- A5 0B	2070 LDA #ZEICH	0F20- 4C 4C	2700	.BY \$00 'TAB'
0D23- C9 40	2080 CMP #64	0F22- 00 54 41	2710	.BY \$00 'TEXT'
0D25- 90 07	2090 BCC CONV1	0F25- 42	2720	.BY \$00 'TOP'
0D27- C9 06	2100 CMP #96	0F26- 00 54 45	2730	.BY \$00 'DOWN'
0D29- B0 04	2110 BCS CONV2	0F29- 5B 54		
0D2B- 18	2120 CLC	0F2B- 00 54 4F		
0D2C- 69 20	2130 ADC #32	0F2E- 50		
0D2E- 60	2140 RTS	0F2F- 00 44 4F		
0D2F- C9 C0	2150 CONV2	0F32- 57 4E		


```

OF34- 00 52 56 2740      .BY $00 'RVS'
OF37- 53
OF38- 00 48 4F 2750      .BY $00 'HOME'
OF3B- 4D 45
OF3D- 00 44 45 2760      .BY $00 'DEL'
OF40- 4C
OF41- 00 44 45 2770      .BY $00 'DELNE'
OF44- 4C 49 4E
OF47- 45
OF48- 00 45 52 2780      .BY $00 'ERSEND'
OF4B- 53 45 4E
OF4E- 44
OF4F- 00 53 43 2790      .BY $00 'SCRUP'
OF52- 52 55 50
OF55- 00 45 53 2800      .BY $00 'ESC'
OF58- 43
OF59- 00 52 49 2810      .BY $00 'RIGHT'
OF5C- 47 48 54
OF5F- 00 42 45 2820      .BY $00 'BELL'
OF62- 4C 4C
OF64- 00 54 41 2830      .BY $00 'TABC'
OF67- 42 43
OF69- 00 53 48 2840      .BY $00 'SHRTN'
OF6C- 52 54 4E
OF6F- 00 47 52 2850      .BY $00 'GRAPH'
OF72- 41 50 48
OF75- 00 42 4F 2860      .BY $00 'BOT'
OF78- 54
OF79- 00 55 50 2870      .BY $00 'UP'
OF7C- 00 4F 46 2880      .BY $00 'OFF'
OF7F- 46
OF80- 00 43 4C 2890      .BY $00 'CLR'
OF83- 52
OF84- 00 49 4E 2900      .BY $00 'INST'
OF87- 53 54
OF89- 00 49 4E 2910      .BY $00 'INSLNE'
OF8C- 53 4C 4E
OF8F- 45
OF90- 00 45 52 2920      .BY $00 'ERSBEG'
OF93- 53 42 45
OF96- 47
OF97- 00 53 43 2930      .BY $00 'SCRDNW'
OF9A- 52 44 57
OF9D- 4E
OF9E- 00 53 48 2940      .BY $00 'SHESC'
OFA1- 45 53 43
OFA4- 00 4C 45 2950      .BY $00 'LEFT'
OFA7- 46 54
OFA9- 00 50 49 2960      .BY $00 'PI' $00 ; ENDE TEXTTAB
OFAC- 00
2970 ;
2980 ; RESETVECTOR
2990 ;
3000      .BA $OFFC
3010      .MC $7FFC
3020 ;
3030 RSTVECT      .BY $00 $0C
3040 ;
3050      .EN
ENDE VON MAE PASS !

```

```

10 rem emuf interface demoprogramm
15 :
20 open 1,4,1 : rem umrechnung mit cursor zeichen
30 open 2,4,3 : rem condensed mode
40 open 3,4,9 : rem enlarged mode
50 open 4,4,11: rem condensed & enlarged mode
60 cl$=chr$(24) : rem can-code
70 a$= "the quick brown fox jumps over the lazy dog"
100 for x=1to4
110 print#x,a$:print#x,cl$ : rem teststring und can code drucken
120 next x
130 :
140 b$="DOWNDOWNHOMEINSTINSTRIGHTRIGHTCLRP": rem 2*down;ho
me;3*inst;2*right;clear;up
150 c$="u=2*PI-2+PI/(PI*2/PI)": rem pi-string
160 for x=1to4
170 print#x,b$:print#x,c$:print#x,cl$ :rem b$ & a$ an drucker geben
180 nextx
190 close1:close2:close3:close4
ready.

```

Bild 4. Basic-Demonstrationsprogramm. Es wurde mit dem Interface und einem MX-80 ausgedruckt, indem am CBM die Befehlszeile OPEN 1,4,1:CMD1:LIST eingetippt wurde

Erweiterter EMUF

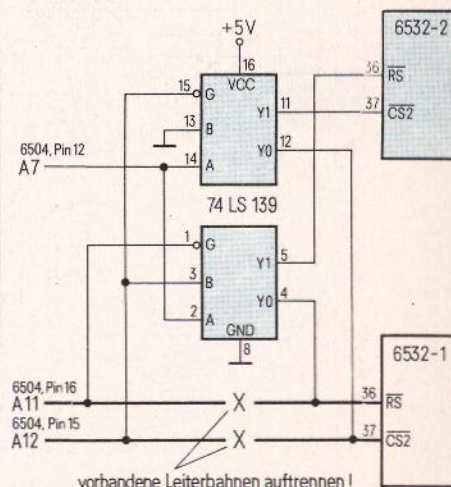
Zwei 8-Bit-Ports mehr und einen zusätzlichen Timer gewinnt man, wenn man den Einplatinencomputer EMUF (siehe mc 1981, Heft 2, und EMUF-Sonderheft) mit einem weiteren 6532-Baustein ausrüstet. Außerdem hat man 128 Byte mehr RAM zur Verfügung.

Die zusätzlichen E/A-Adressen

880	Port A
881	Port-A-Richtungsregister
882	Port B
883	Port-B-Richtungsregister
894	Timer 1 µs
895	Timer 8 µs
896	Timer 64 µs
897	Timer 1024 µs
89C	Timer wie oben, jedoch mit IRQ
89D	
89E	
89F	Timer lesen
896	
897	Timer testen (N-Flag)

Die Tabelle nennt die Adressenbelegung, die sich ergibt, wenn man die zwei Decoder eines einzigen Bausteins 74 LS 139 zur Gewinnung der Chipselect- und RAM-Select-Signale der nun zwei 6532-ICs heranzieht. Alle hier nicht gezeichneten Anschlüsse des neuen 6532 werden mit den entsprechenden des ersten verbunden. Die Verdrahtung erfolgt am einfachsten in Fädertechnik mit dem direkt verlötbaren Kupferlackdraht (0,2 mm).

Walter Ochs



Decodierung der RAM-Select- und Chipselect-Signale für die nunmehr zwei 6532-Bausteine auf der EMUF-Platine

Rudolf Hofer

EMUF als Schaltuhr, Thermometer, Stoppuhr und Frequenzmesser

Will man den EMUF als eigenständiges Gerät einsetzen, dann braucht man in vielen Fällen Tastatur und Anzeigeeinheit. Daß dies nicht gleichbedeutend mit großem Zusatzaufwand an Hardware ist, zeigt dieser Beitrag. Auch die beschränkte Zahl von Ein-/Ausgabeleitungen stellt kein grundsätzliches Hindernis für zahlreiche Anwendungen dar. Oft kann man sich, wie im folgenden gezeigt wird, durch Mehrfachausnutzung von Portanschlüssen oder Seriell/Parallel-Umsetzung behelfen.

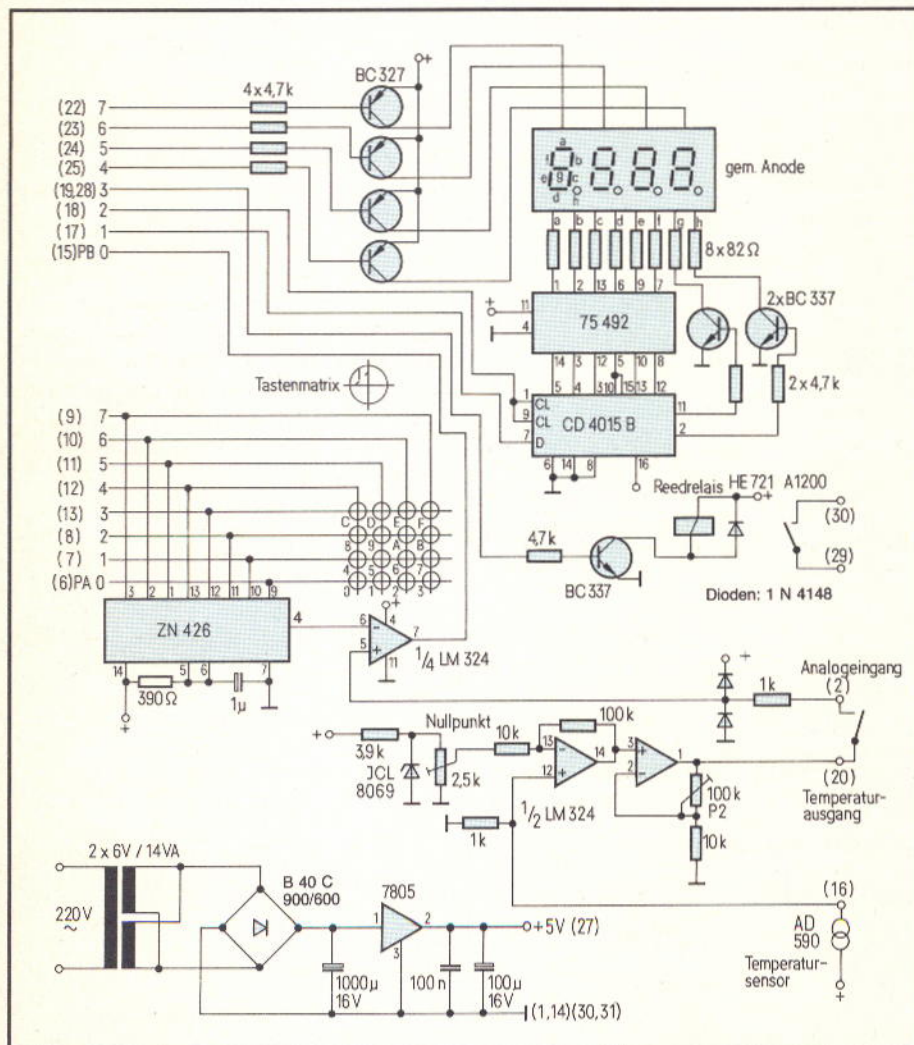


Bild 1. Tastaturanschluß, Anzeigeeinheit, A/D-Umsetzer und Schaltausgang für den EMUF. Als Frequenzmeßeingang wird PA 7 verwendet.

Die Schaltung von Bild 1 wurde als Peripheriekarte für den EMUF (siehe mc 1981, Heft 2, oder EMUF-Sonderheft) entwickelt. Sie enthält die Funktionsblöcke „Anzeige“ und „4x4-Tastatur“. Für die spezielle Anwendung als Hilfsgerät im Labor wurde parallel zur Tastatur ein D/A-Umsetzer (ZN 426) angeschlossen, der im Zusammenspiel mit der Software als A/D-Umsetzer betrieben wird. Ein Vierfach-Operationsverstärker bereitet das Signal eines Temperaturfühlers (z. B. AD 590) auf, der einen Strom von $1 \mu\text{A/K}$ liefert. Zusätzlich ist ein Schaltausgang vorhanden, mit dem man Lampen, Heizungspumpen oder ähnliches ansteuern kann. Anschluß PA7 wird außerdem zur Frequenzmessung (0...10 kHz) verwendet. Mit dem unbenutzten Verstärker kann man das Signal auf den richtigen Pegel bringen, falls es sich nicht ohnehin um ein TTL-Signal handelt.

Das in Bild 2 abgedruckte Programm enthält zahlreiche Routinen, mit denen häufig vorkommende Standardaufgaben erledigt werden, z. B. A/D-Umsetzung und Tastaturabfrage. Eine Besonderheit ist die Art, wie die 4stellige 7-Segment-Anzeigeeinheit betrieben wird: Die Interrupt-Routine IRQ schiebt jeweils ein Anzeigemuster (gespeichert in $W...W+3$) in das Schieberegister CD4015 und schaltet den Transistor der entsprechenden Stelle durch. Vier aufeinanderfolgende Interrupts ergeben somit einen kompletten Anzeigezyklus. Ist Bit 7 von DFLAG null, dann wird automatisch der Inhalt der Speicherzellen $DREG...DREG+3$ in den 7-Segment-Code umgewandelt und ins 7-Segment-Register übernommen. Führende Nullen werden dann unterdrückt. Im Hauptprogramm braucht man also lediglich das Anzeigeregister (DREG) mit den Dezimal- oder Hexadezimalzahlen zu laden, die angezeigt werden sollen (Bit 7 von DFLAG = 0). Will man – soweit das mit 7-Segment-Anzeigen überhaupt möglich

Bild 2. Dieses Programm sorgt dafür, daß der EMUF zum nützlichen Hilfsgerät im Labor wird

0800	1	;*****	0C3A 205F0F	68	JSR SCAN
0800	2	;SCHALTUHR, THERMOMETER,	0C3D C900	69	CMP #0
0800	3	;STOPPUHR UND FREQUENZ-	0C3F F003	70	BEQ FRQ
0800	4	;MESSER	0C41 4C690E	71	JMP UHR
0800	5	;*****	0C44 4CE70C	72	FRQ JMP FREQU
0800	6	C1024 EQU \$B17	0C47	73	;*****
0800	7	IFLAG EQU \$B15	0C47	74	; INTERRUPT FUER
0800	8	CL64 EQU \$B1E	0C47	75	; ANZEIGE
0800	9	PBD EQU \$B03	0C47 48	76	IRQ PHA
0800	10	PB EQU \$B02	0C48 861A	77	STX XTEMP
0800	11	PA EQU \$B00	0C4A 841B	78	STY YTEMP
0800	12	PAD EQU \$B01	0C4C A940	79	LDA #\$40
0800	13	RESV EQU \$FFC	0C4E 8D1E0B	80	STA CL64
0800	14	IRGV EQU \$FFE	0C51 2417	81	ANZG BIT DFLAG
0800	15	XTEMP EPZ \$1A	0C53 3006	82	BMI DIR
0800	16	YTEMP EPZ \$1B	0C55 20AD0C	83	JSR CONV
0800	17	YTEM1 EPZ \$1C	0C58 20D50C	84	JSR HIDE
0800	18	YTEM2 EPZ \$1D	0C5B 206C0C	85	DIR JSR SEND
0800	19	YTEM3 EPZ \$1E	0C5E C608	86	DEC DIGIT
0800	20	;7-SEG.-REGISTER	0C60 1004	87	BPL RTN
0800	21	W EPZ 0	0C62 A903	88	LDA #3
0800	22	D EPZ 4	0C64 8508	89	STA DIGIT
0800	23	DIGIT EPZ 8	0C66 A61A	90	RTN LDX XTEMP
0800	24	;ANZEIGEREG.	0C68 A41B	91	LDY YTEMP
0800	25	DREG EPZ 9	0C6A 68	92	PLA
0800	26	;VERGLEICHREG.	0C6B 40	93	RINT RTI
0800	27	VREG1 EPZ \$8	0C6C	94	;-----
0800	28	VREG2 EPZ \$D	0C6C AD020B	95	SEND LDA PB
0800	29	;ZEITTAKT	0C6F 09F0	96	ORA #\$FO
0800	30	CNT EPZ \$F	0C71 8D020B	97	STA PB
0800	31	SEK EPZ \$10	0C74 A608	98	LDX DIGIT
0800	32	MIN EPZ \$11	0C76 B500	99	LDA W,X
0800	33	STD EPZ \$12	0C78 A008	100	LDY #8
0800	34	;STD/SEK-FLAG	0C7A 2A	101	SCHBW ROL
0800	35	FLAG EPZ \$13	0C7B 208A0C	102	JSR BITA
0800	36	;IRQ-VEKTOR	0C7E 88	103	DEY
0800	37	INTR EPZ \$14	0C7F D0F9	104	BNE SCHBW
0800	38	;DISPLAY-FLAG	0C81 AD020B	105	LDA PB
0800	39	;BIT 7=1:STATT DREG	0C84 3504	106	AND D,X
0800	40	;W ANZEIGEN	0C86 8D020B	107	STA PB
0800	41	;BIT 6=1:ZEIT NICHT	0C89 60	108	RTS
0800	42	;LAUFEND IN ANZEIGE	0C8A	109	;-----
0800	43	;UEBERNEHMEN	0C8A	110	;BIT AUSGEBEN (C)
0800	44	DFLAG EPZ \$17	0C8A 4B	111	BITA PHA
0800	45	U EPZ \$18	0C8B AD020B	112	LDA PB
0800	46	KEY EPZ \$19	0C8E 900C	113	BCC NULL
0800	47	;-----	0C90 0902	114	ORA #2
0C00	48	ORG \$C00	0C92 8D020B	115	STA PB
0C00 A2FF	49	LDX #\$FF	0C95 0906	116	ORA #6
0C02 9A	50	TXS	0C97 8D020B	117	STA PB
0C03 D8	51	CLD	0C9A D00A	118	BNE TKTN
0C04 4C200C	52	JMP START	0C9C 29FD	119	NULL AND #111111101
0C07 000000	53	TBL HEX 00000000EFDFFB	0C9E 8D020B	120	STA PB
		7F03000000000000	0CA1 0904	121	ORA #4
		00000000000004C	0CA3 8D020B	122	STA PB
0C0A 00EFDFF			0CA6 29F9	123	TKTN AND #111111001
0C0D BF7F03			0CA8 8D020B	124	STA PB
0C10 000000			0CAB 68	125	PLA
0C13 000000			0CAC 60	126	RTS
0C16 000000			0CAD	127	;-----
0C19 00004C			0CAD	128	;D-REGISTER IN ANZEIGE
0C1C 470C	54	ADR IRQ	0CAD A003	129	CONV LDY #3
0C1E 0020	55	HEX 0020	0CAF A50A	130	LDA DREG+1
0C20 A219	56	START LDX #\$19	0CB1 48	131	PHA
0C22 BD070C	57	ZPT LDA TBL,X	0CB2 20C70C	132	JSR STOR
0C25 9500	58	STA W,X	0CB5 88	133	DEY
0C27 CA	59	DEX	0CB6 68	134	PLA
0C28 10F8	60	BPL ZPT	0CB7 20CB0C	135	JSR STOR1
0C2A A90F	61	LDA #\$F	0CBA 88	136	DEY
0C2C 8D010B	62	STA PAD	0CBB A509	137	LDA DREG
0C2F A940	63	LDA #\$40	0CBD 48	138	PHA
0C31 8D1E0B	64	STA CL64	0CBE 20C70C	139	JSR STOR
0C34 A9FE	65	LDA #11111110	0CC1 88	140	DEY
0C36 8D030B	66	STA PBD	0CC2 68	141	PLA
0C39 58	67	CLI	0CC3 20CB0C	142	JSR STOR1

OCC6 60	143	RTS	OD4C D0FC	221	BNE LOP
OCC7	144	;	OD4E 68	222	PLA
OCC7 4A	145	STOR LSR	OD4F E901	223	SBC #1
OCC8 4A	146	LSR	OD51 D0F6	224	BNE LO
OCC9 4A	147	LSR	OD53 60	225	RTS
OCCA 4A	148	LSR	OD54	226	*****
OCCB 290F	149	STOR1 AND #0F	OD54	227	REG. AUF NULL
OCCD AA	150	TAX	OD54 A900	228	REGNL LDA #0
OCCD BD630D	151	LDA TAB7, X	OD56 8509	229	STA DREG
OCD1 990000	152	STA W, Y	OD58 850A	230	STA DREG+1
OCD4 60	153	RTS	OD5A 8500	231	STA W
OCD5	154	;	OD5C 8501	232	STA W+1
OCD5	155	; VORNULLEN-UNT.	OD5E 8502	233	STA W+2
OCD5 A003	156	HIDE LDY #3	OD60 8503	234	STA W+3
OCD7 B90000	157	HIDE2 LDA W, Y	OD62 60	235	RTS
OCD8 C93F	158	CMP #03F	OD63	236	*****
OCD8 D008	159	BNE HIDE1	OD63 3F065B	237	TAB7 HEX 3F065B4F666D
OCD8 A900	160	LDA #0			7D077F6F777C
OCE0 990000	161	STA W, Y			395E7971
OCE3 88	162	DEY	OD66 4F666D		
OCE4 D0F1	163	BNE HIDE2	OD69 7D077F		
OCE6 60	164	HIDE1 RTS	OD6C 6F777C		
OCE7	165	*****	OD6F 395E79		
OCE7	166	; FREQUENZ MESSEN	OD72 71		
OCE7 A204	167	FREQU LDY #4	OD73	238	*****
OCE9 78	168	SEI	OD73	239	UHR STARTEN
OCEA 20540D	169	JSR REGNL	OD73 78	240	STRT SEI
OCEB AD1508	170	LDA IFLAG	OD74 A989	241	LDA #IRQU
OCEC A902	171	TIME4 LDA #2	OD76 8515	242	STA INTR+1
OCE2 20480D	172	JSR DELAY	OD78 A90D	243	LDA /IRQU
OCE5 20060E	173	JSR RN1	OD7A 8516	244	STA INTR+2
OCE8 20060E	174	JSR RN1	OD7C A99B	245	LDA #155
OCEB A9F4	175	LDA #244	OD7E 8D1E08	246	STA CL64
OCEB 8D1708	176	STA C1024	OD81 A900	247	LDA #0
OD00 2C1508	177	CHECK BIT IFLAG	OD83 8517	248	STA DFLAG
OD03 300A	178	BMI TIMER	OD85 58	249	CLI
OD05 50F9	179	BVC CHECK	OD86 4C690E	250	JMP UHR
OD07 202D0D	180	JSR INKRT	OD89	251	;
OD0A B01B	181	BCS UEBL	OD89	252	INTERRUPT FUER
OD0C 4C000D	182	JMP CHECK	OD89	253	ZEITAKT
OD0F CA	183	DEX	OD89 48	254	IRQU PHA
OD10 D0DE	184	BNE TIME4	OD8A 861A	255	STX XTEMP
OD12 A90A	185	UEBL1 LDA #10	OD8C 841B	256	STY YTEMP
OD14 8D1E08	186	STA CL64	OD8E EA	257	NOP
OD17 58	187	CLI	OD8F EA	258	NOP
OD18 203D0D	188	JSR SEK1	OD90 20060E	259	JSR RN1
OD1B 78	189	SEI	OD93 20060E	260	JSR RN1
OD1C AD0208	190	LDA PB	OD96 20060E	261	JSR RN1
OD1F 09F0	191	ORA #0F0	OD99 20060E	262	JSR RN1
OD21 8D0208	192	STA PB	OD9C 78	263	SEI
OD24 4CE70C	193	JMP FREQU	OD9D A99B	264	LDA #155
OD27	194	*****	OD9F 8D1E08	265	STA CL64
OD27 A940	195	UEBL LDA #40	ODA2 F8	266	SED
OD29 8500	196	STA W	ODA3 A000	267	LDY #0
OD2B D0E5	197	BNE UEBL1	ODA5 38	268	SEC
OD2D	198	*****	ODA6 A50F	269	LDA CNT
OD2D	199	DEZIMALREG. INKR.	ODA8 6900	270	ADC #0
OD2D F8	200	INKRT SED	ODAA 850F	271	STA CNT
OD2E 38	201	SEC	ODAC C900	272	CMP #0
OD2F A900	202	LDA #0	ODAE D02A	273	BNE FLG
OD31 6509	203	ADC DREG	ODB0 38	274	SEC
OD33 8509	204	STA DREG	ODB1 A510	275	LDA SEK
OD35 A900	205	LDA #0	ODB3 6900	276	ADC #0
OD37 650A	206	ADC DREG+1	ODB5 8510	277	STA SEK
OD39 850A	207	STA DREG+1	ODB7 C960	278	CMP #060
OD3B D8	208	CLD	ODB9 D01F	279	BNE FLG
OD3C 60	209	RTS	ODBB 8410	280	STY SEK
OD3D	210	*****	ODBD 38	281	SEC
OD3D	211	SEKUNDE VERZÜGERUNG	ODBE A511	282	LDA MIN
OD3D A204	212	SEK1 LDY #4	ODC0 6900	283	ADC #0
OD3F A9FF	213	DEL1 LDA #255	ODC2 8511	284	STA MIN
OD41 20480D	214	JSR DELAY	ODC4 C960	285	CMP #060
OD44 CA	215	DEX	ODC6 D012	286	BNE FLG
OD45 D0F8	216	BNE DEL1	ODC8 8411	287	STY MIN
OD47 60	217	RTS	ODCA 38	288	SEC
OD48 38	218	DELAY SEC	ODCB A512	289	LDA STD
OD49 48	219	LD PHA	ODCD 6900	290	ADC #0
OD4A E901	220	LOP SBC #1	ODCF 8512	291	STA STD

0DD1 C924	292	CMP ##24	0E69 205F0F	370	UHR	JSR SCAN
0DD3 D005	293	BNE FLG	0E6C C90C	371		CMP ##C
0DD5 8412	294	STY STD	0E6E F0ED	372		BEQ STR
0DD7 20540D	295	JSR REGNL	0E70 C90B	373		CMP ##B
0DDA D8	296	CLD	0E72 F046	374		BEQ SWCH
0DDB 2417	297	BIT DFLAG	0E74 C907	375		CMP ##7
0DDD 7013	298	BVS ANZG1	0E76 F0EE	376		BEQ TAK
0DDF 2413	299	BIT FLAG	0E78 0617	377		ASL DFLAG
0DE1 3009	300	BMI SEKUN	0E7A 38	378		SEC
0DE3 200B0E	301	JSR VGLST	0E7B 6617	379		ROR DFLAG
0DE6 20F50D	302	JSR STANZ	0E7D C908	380		CMP ##8
0DE9 4C510C	303	JMP ANZG	0E7F F017	381		BEQ TIMA
0DEC 20070E	304	SEKUN JSR VGLSK	0E81 C909	382		CMP ##9
0DEF 20FE0D	305	JSR ZSANZ	0E83 F0DB	383		BEQ TION
0DF2 4C510C	306	ANZG1 JMP ANZG	0E85 C90A	384		CMP ##A
0DF5	307	;	0E87 F0DA	385		BEQ TIOFF
0DF5	308	;STD ANZEIGEN	0E89 C90E	386		CMP ##E
0DF5 A511	309	STANZ LDA MIN	0E8B F0DC	387		BEQ UHR
0DF7 8509	310	STA DREG	0E8D 0617	388		ASL DFLAG
0DF9 A512	311	LDA STD	0E8F 18	389		CLC
0DFB 4C040E	312	JMP ANZ1	0E90 6617	390		ROR DFLAG
0DFE	313	;SEK ANZEIGEN	0E92 C90D	391		CMP ##D
0DFE A50F	314	ZSANZ LDA CNT	0E94 F031	392		BEQ STOPP
0E00 8509	315	STA DREG	0E96 D0D1	393		BNE UHR
0E02 A510	316	LDA SEK	0E98	394		;
0E04 850A	317	ANZ1 STA DREG+1	0E98	395		;ZEIT EINGEBEN
0E06 60	318	RN1 RTS	0E98 A978	396	TIMA	LDA ##78
0E07	319	;	0E9A 8503	397		STA W+3
0E07	320	;ZEITVERGLEICH	0E9C A930	398		LDA ##30
0E07	321	;UND SCHALTFUNKTION	0E9E 8502	399		STA W+2
0E07 A906	322	VGLSK LDA #6	0EAO A937	400		LDA ##37
0E09 D002	323	BNE VGL1	0EA2 8501	401		STA W+1
0E0B A908	324	VGLST LDA #8	0EA4 A979	402		LDA ##79
0E0D 851E	325	VGL1 STA YTEM3	0EA6 8500	403		STA W
0E0F A204	326	LDX #4	0EAB A906	404		LDA #6
0E11 A002	327	LDY #2	0EAA 851D	405		STA YTEM2
0E13 20490E	328	JSR COMP	0EAC 2413	406		BIT FLAG
0E16 F010	329	BEQ OFF	0EAE 3004	407		BMI SEKD
0E18 B017	330	BCS VGL2	0EB0 E61D	408		INC YTEM2
0E1A A61E	331	LDX YTEM3	0EB2 E61D	409		INC YTEM2
0E1C 20490E	332	JSR COMP	0EB4 20D90E	410	SEKD	JSR TRANS
0E1F B01E	333	BCS ON	0EB7 4C690E	411		JMP UHR
0E21 A004	334	LDY #4	0EBA	412		;
0E23 20490E	335	JSR COMP	0EBA	413		;STUNDEN/SEKUNDEN-UMSCHALT.
0E26 9017	336	BCC ON	0EBA 2613	414	SWCH	ROL FLAG
0E28 AD0208	337	OFF LDA PB	0EBC 2613	415		ROL FLAG
0E2B 29F7	338	AND #%11110111	0EBE E613	416		INC FLAG
0E2D BD0208	339	STA PB	0EC0 6613	417		ROR FLAG
0E30 60	340	VGL3 RTS	0EC2 6613	418		ROR FLAG
0E31 A61E	341	VGL2 LDX YTEM3	0EC4 4C690E	419		JMP UHR
0E33 20490E	342	JSR COMP	0EC7	420		;
0E36 90F0	343	BCC OFF	0EC7	421		;ZEIT ANHALTEN
0E38 A004	344	LDY #4	0EC7 78	422	STOPP	SEI
0E3A 20490E	345	JSR COMP	0EC8 A947	423		LDA #IRQ
0E3D B0E9	346	BCS OFF	0ECA 8515	424		STA INTR+1
0E3F AD0208	347	ON LDA PB	0ECC A90C	425		LDA /IRQ
0E42 0908	348	ORA #%00001000	0ECE 8516	426		STA INTR+2
0E44 BD0208	349	STA PB	0ED0 A93E	427		LDA #62
0E47 D0E7	350	BNE VGL3	0ED2 8D1E08	428		STA CL64
0E49	351	;	0ED5 58	429		CLI
0E49 F8	352	COMP SED	0ED6 4C690E	430		JMP UHR
0E4A 38	353	SEC	0ED9	431		;
0E4B B50A	354	LDA DREG+1,X	0ED9	432		;REG. LADEN
0E4D F90A00	355	SBC DREG+1,Y	0ED9	433		;YTEM2 BESTIMMT REG.
0E50 F002	356	BEQ FORTS	0ED9 203D0D	434	TRANS	JSR SEK1
0E52 D8	357	BACK CLD	0EDC 203D0D	435		JSR SEK1
0E53 60	358	RTS	0EDF 38	436	TRAN1	SEC
0E54 38	359	FORTS SEC	0EE0 6617	437		ROR DFLAG
0E55 B509	360	LDA DREG,X	0EE2 4617	438		LSR DFLAG
0E57 F90900	361	SBC DREG,Y	0EE4 A41D	439		LDY YTEM2
0E5A 4C520E	362	JMP BACK	0EE6 B90900	440		LDA DREG,Y
0E5D	363	;	0EE9 8509	441		STA DREG
0E5D 4C730D	364	STR JMP STRT	0EEB B90A00	442		LDA DREG+1,Y
0E60 4C080F	365	TION JMP TON	0EEE 850A	443		STA DREG+1
0E63 4C220F	366	TIOFF JMP TIOFF	0EFO 203A0F	444	TMA1	JSR INDEZ
0E66 4CC30F	367	TAK JMP TAKE	0EF3 C90F	445		CMP ##F
0E69	368	;	0EF5 D0F9	446		BNE TMA1
0E69	369	;EINGABESCHLEIFE	0EF7 A509	447		LDA DREG

0EF9 A41D	448	LDY YTEM2	0F75 A900	516	SCAN3	LDA #0
0EFB 990900	449	STA DREG,Y	0F77 8519	517		STA KEY
0EFE A50A	450	LDA DREG+1	0F79 A203	518		LDX #3
0F00 990A00	451	STA DREG+1,Y	0F7B A90E	519		LDA #0E
0F03 A900	452	LDA #0	0F7D 8D000B	520		STA PA
0F05 8517	453	STA DFLAG	0F80 A004	521	SCAN1	LDY #4
0F07 60	454	RTS	0F82 A910	522		LDA #10
0F08	455	;	0F84 2C000B	523	SCAN2	BIT PA
0F08	456	;EINSCHALTZEIT	0F87 F00F	524		BEQ FND
0F08 A97B	457	TON LDA #78	0F89 E619	525		INC KEY
0F0A 8503	458	STA W+3	0F8B 0A	526		ASL
0F0C A95C	459	LDA #5C	0F8C 8B	527		DEY
0F0E 8502	460	STA W+2	0F8D D0F5	528		BNE SCAN2
0F10 A954	461	LDA #54	0F8F 0E000B	529		ASL PA
0F12 8501	462	STA W+1	0F92 EE000B	530		INC PA
0F14 A900	463	LDA #0	0F95 CA	531		DEX
0F16 8500	464	STA W	0F96 10E8	532		BPL SCAN1
0F18 A902	465	LDA #2	0F98 60	533	FND	RTS
0F1A 851D	466	STA YTEM2	0F99	534	;	*****
0F1C 20D90E	467	JSR TRANS	0F99 A9FF	535	ADU	LDA #FF
0F1F 4C690E	468	JMP UHR	0F9B 8D010B	536		STA PAD
0F22	469	;	0F9E A900	537		LDA #0
0F22	470	;AUSSCHALTZEIT	0FA0 8D000B	538		STA PA
0F22 A97B	471	TOFF LDA #78	0FA3 A901	539	TEST	LDA #1
0F24 8503	472	STA W+3	0FA5 20480D	540		JSR DELAY
0F26 A95C	473	LDA #5C	0FAB A901	541		LDA #1
0F28 8502	474	STA W+2	0FAA 2C020B	542		BIT PB
0F2A A971	475	LDA #71	0FAD F00B	543		BEQ RET
0F2C 8501	476	STA W+1	0FAF 202D0D	544		JSR INKRT
0F2E 8500	477	STA W	0FB2 202D0D	545		JSR INKRT
0F30 A904	478	LDA #4	0FB5 EE000B	546		INC PA
0F32 851D	479	STA YTEM2	0FB8 D0E9	547		BNE TEST
0F34 20D90E	480	JSR TRANS	0FBA AD000B	548	RET	LDA PA
0F37 4C690E	481	JMP UHR	0FBD A00F	549		LDY #F
0F3A	482	;	0FBF 8C010B	550		STY PAD
0F3A	483	;DEZ.-EINGABE	0FC2 60	551		RTS
0F3A 205F0F	484	INDEZ JSR SCAN	0FC3	552	;	*****
0F3D C90A	485	CMP #A	0FC3	553	;	AD-UMSETZUNG UND
0F3F 100C	486	BPL INRT	0FC3	554	;	MITTELWERT BILDEN
0F41 3003	487	BMI INCH1	0FC3 A20A	555	TAKE	LDX #10
0F43 205F0F	488	INCH JSR SCAN	0FC5 A9C0	556		LDA #C0
0F46 204E0F	489	INCH1 JSR ROT	0FC7 0517	557		ORA DFLAG
0F49 0509	490	ORA DREG	0FC9 8517	558		STA DFLAG
0F4B 8509	491	STA DREG	0FCB 20540D	559		JSR REGNL
0F4D 60	492	INRT RTS	0FCE 20990F	560	NCH	JSR ADU
0F4E	493	;	0FD1 CA	561		DEX
0F4E 0609	494	ROT ASL DREG	0FD2 D0FA	562		BNE NCH
0F50 260A	495	ROL DREG+1	0FD4 A9F0	563		LDA #F0
0F52 0609	496	ASL DREG	0FD6 2509	564		AND DREG
0F54 260A	497	ROL DREG+1	0FDB 8509	565		STA DREG
0F56 0609	498	ASL DREG	0FDA 0617	566		ASL DFLAG
0F58 260A	499	ROL DREG+1	0FDC 18	567		CLC
0F5A 0609	500	ASL DREG	0FDD 6617	568		ROR DFLAG
0F5C 260A	501	ROL DREG+1	0FDF 203D0D	569		JSR SEK1
0F5E 60	502	RTS	0FE2 A980	570		LDA #80
0F5F	503	;	0FE4 8D000B	571		STA PA
0F5F	504	;TASTATURABFRAGE	0FE7 2C000B	572		BIT PA
0F5F 841C	505	SCAN STY YTEM1	0FEA D0D7	573		BNE TAKE
0F61 20750F	506	LKEY JSR SCAN3	0FEC 4617	574		LSR DFLAG
0F64 F0FB	507	BEQ LKEY	0FEE 4C690E	575		JMP UHR
0F66 20750F	508	NKEY JSR SCAN3	0FF1	576	;	*****
0F69 D0FB	509	BNE NKEY	0FFC	577		ORG RESV
0F6B 20750F	510	JSR SCAN3	0FFC 000C	578		HEX 000C
0F6E D0F6	511	BNE NKEY	0FFE	579		ORG IRQV
0F70 A519	512	LDA KEY	0FFE 1400	580		ADR INTR
0F72 A41C	513	LDY YTEM1	1000	581		PAU
0F74 60	514	RTS				
0F75	515	;				

ist – Buchstaben darstellen, dann setzt man Bit 7 von DFLAG auf eins und schreibt das Muster in die Zellen W...W+3. Dabei entspricht Segment a dem Bit 0, Segment b dem Bit 1 usw.

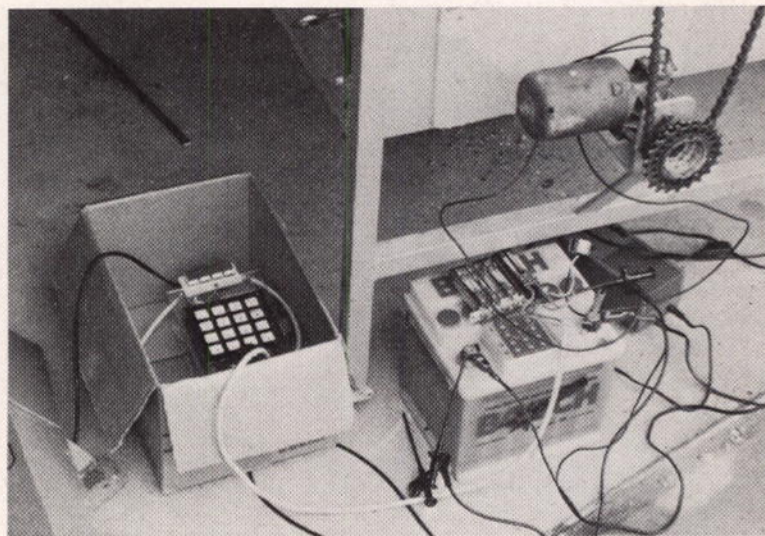
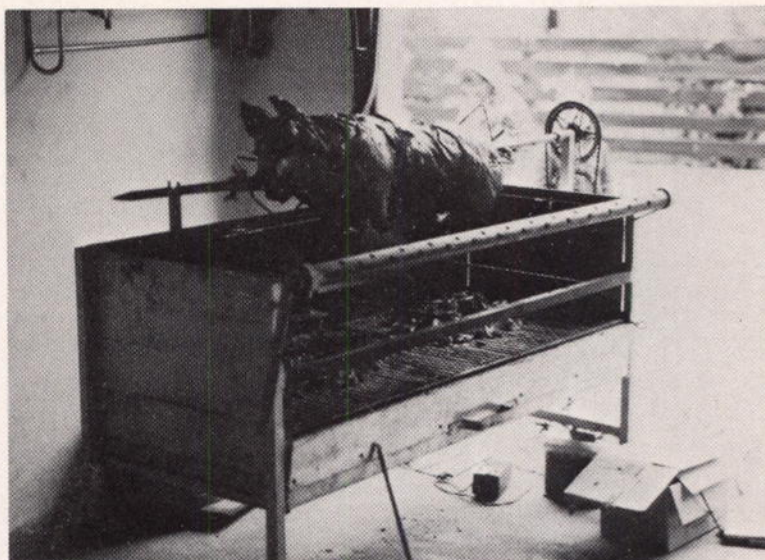
dargestellt (Bild 3). In die Betriebsart „Frequenzmessung“ gelangt man nur, indem man nach dem Reset die Null-Taste betätigt. Zu beachten ist, daß zu diesem Zeitpunkt an PA7 kein Signal

eingestellt werden. Nach dem Tastendruck erscheint als Bestätigung das entsprechende Wort in etwas getrickster 7-Segment-Schrift etwa eine Sekunde in der Anzeige. Danach erscheint die einge-

Eine ungewöhnliche Anwendung

Den Beweis, daß er wirklich universell verwendbar ist, lieferte unser Labor-EMUF bei einer ungewöhnlichen Gelegenheit. In der Nachbarschaft des Verfassers sollte ein Spanferkel gegrillt werden. Da knapp 30 Personen davon satt werden sollten, mußte eine ausgewachsene Sau daran glauben. Alles war vorbereitet; der Grill – von einem befreundeten Metzger ausgeliehen – stand bereit und sollte seine erste Bewährungsprobe bestehen. Doch der Antriebsmechanismus schleuderte das arme Schwein derart durch die Gegend, daß es sich in seine Bestandteile auflösen drohte. Einige schnell herbeigeholte Vorwiderstände, die den Lauf des Motors etwas beruhigen sollten, erwiesen sich als nutzlos, da sich das Schwein – wegen der ungleichmäßigen Belastung des Motors – jetzt gar nicht mehr drehte. Als sich die männlichen Festteilnehmer schon damit abgefunden hatten, daß sie sich wohl auf einen mehrstündigen Schichtdienst im Schweinedrehen einzustellen hätten, kam die rettende Idee. Die Zeitschaltfunktion des Labor-EMUF wurde dazu ausgenutzt, den Motor periodisch ein- und auszuschalten. Da dies normalerweise nur im 1-Minuten-Takt geht, mußte noch der 60-s-Vergleich im EPROM auf drei Sekunden geändert werden (1 Byte). Jetzt konnte auf eine Hundertstelsekunde genau eingestellt werden, wie lange sich das Schwein alle drei Sekunden drehen sollte. Das Ergebnis: fünf Stunden manueller „Dreharbeiten“ eingespart und ein wunderbar gleichmäßig gegrilltes „Spanferkel“, das, nebenbei bemerkt, ausgezeichnet geschmeckt hat.

Rudolf Hofer



Das abgedruckte Programm und die gezeigte Schaltung machen den EMUF zur Schalt- und Stoppuhr, zum Thermometer und zum Frequenzmesser. Die Bedienung ist in Form eines Flußdiagramms

anliegen darf. Wegen der Mehrfachausnutzung dieser Leitung muß man das in Kauf nehmen. Mit den Tasten „TIME“, „T ON“ und „T OFF“ können die aktuelle Zeit, eine Ein- und eine Ausschaltzeit

stellte Zeit. Sie kann unverändert übernommen (Enter) oder neu eingetippt werden. Ziffern größer als Neun nimmt das Gerät nicht an.

Da die Anzeige nur 4stellig ist, ist eine

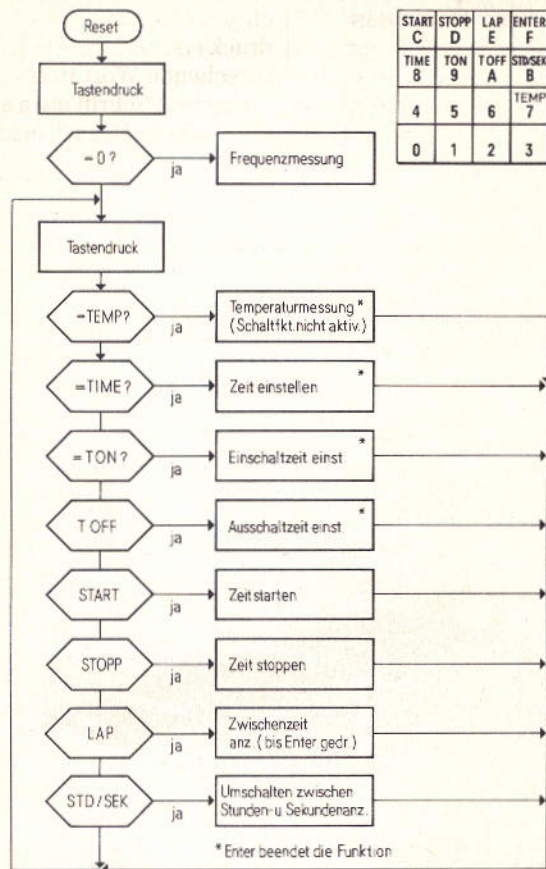


Bild 3. Das Gerät ist auf einfache Weise mit einer 4x4-Tastatur zu bedienen

Umschaltmöglichkeit zwischen Stunden/Minuten und Sekunden/Hundertstelsekunden vorgesehen. Intern läuft die Uhr immer mit voller Auflösung weiter – und zwar ohne Unterbrechung ab „START“, bis „STOPP“ gedrückt wird. Der Schaltausgang ist allerdings vom eingestellten Anzeigemodus abhängig. Dadurch kann man Schaltzeiten im 24-Stunden-Rhythmus mit einer Minute Auflösung oder im Minuten-Rhythmus mit einer Hundertstelsekunde Auflösung einstellen.

Mit der angegebenen Beschaltung können Temperaturen von 0...51,2 °C gemessen werden. Die Genauigkeit ist bei kleinen Werten nicht besonders hoch, da die Operationsverstärker nur mit +5 V versorgt werden. Beim Abgleich stellt man mit P1 null Grad Celsius ein und mit P2 eine bekannte Temperatur, die möglichst nahe am Maximalwert liegen sollte. Spannungsmessungen kann man durchführen, wenn man statt des Temperatursignals 0...2,56 V anlegt.

Die Platine und das Programm im EPROM sind bei der Fa. Wirth, Mühlstr. 25, 7064 Remshalden, Tel. (0 71 51) 7 12 26, erhältlich. Unbedingt zu beachten ist, daß die Anschlüsse 30, 29, 28 und 2 nicht mit dem EMUF-Stecker verbunden werden dürfen – die restlichen Anschlüsse können durchverbunden werden.

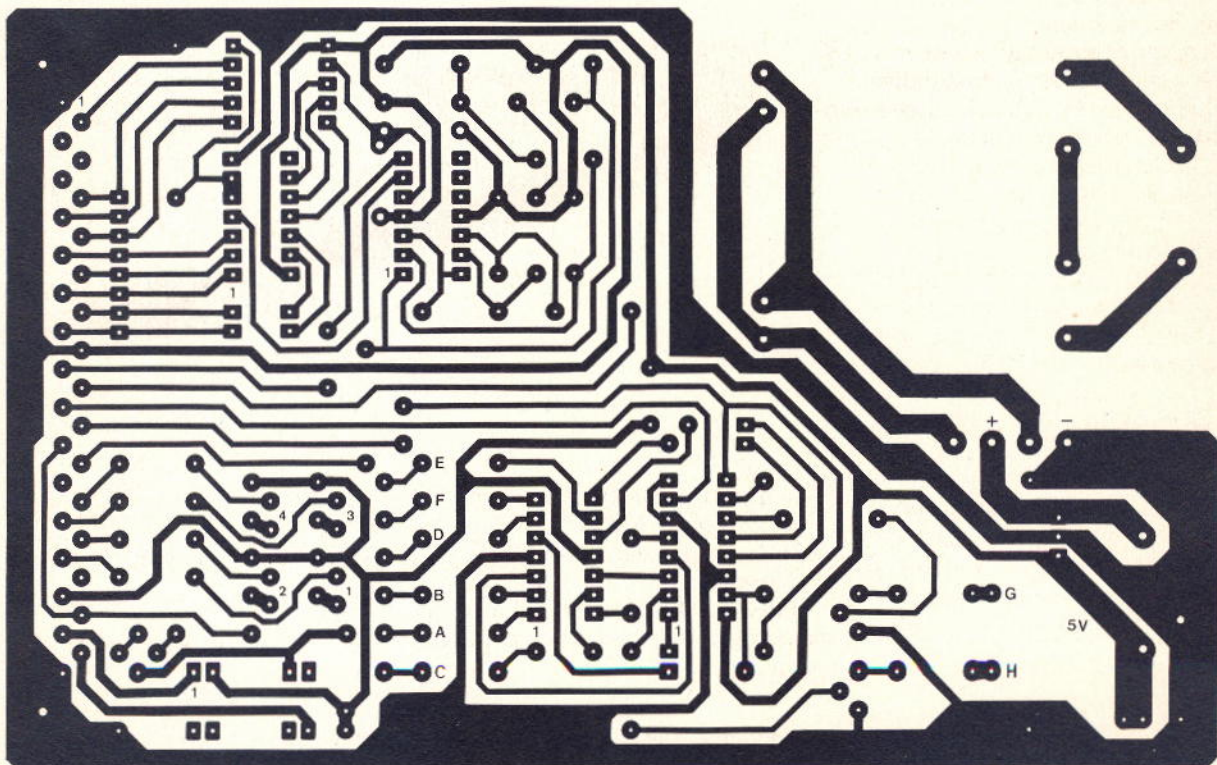


Bild 4. Bei der Peripherieplatine wurden einige Anschlüsse anders belegt als beim EMUF; die Anschlüsse 30, 29, 28 und 2 dürfen deshalb nicht durchverbunden werden (Lieferquelle der Platine siehe Text)

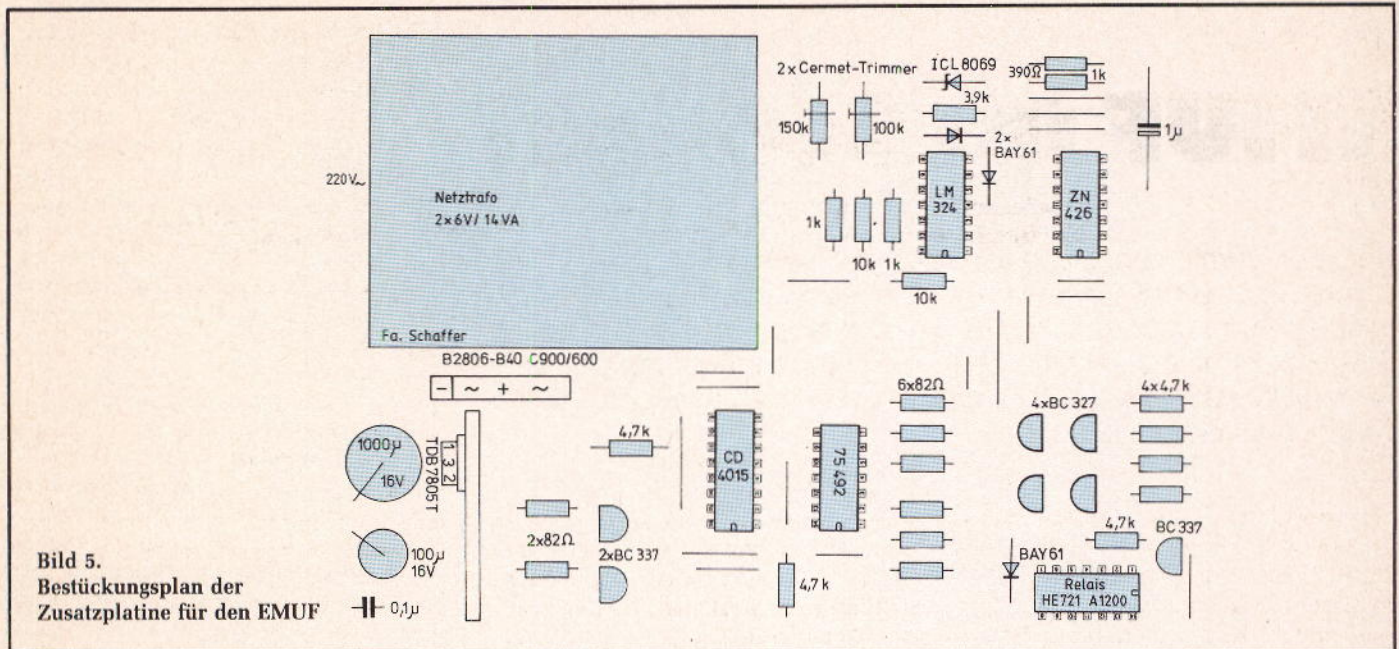


Bild 5.
Bestückungsplan der
Zusatzplatine für den EMUF

Interrupts mit dem 6532-Timer

Der im Einplatinen-Computer EMUF (mc 1981, Heft 2) enthaltene Multifunktionsbaustein 6532 enthält 128 Byte RAM, zwei 8-Bit-I/O-Ports und einen programmierbaren Interrupt-Timer. Hier sei kurz erklärt, wie dieser Timer zu programmieren ist, nachdem das aus der Applikationsliteratur nicht immer eindeutig hervorgeht.

Will man eine einfache Verzögerungsschleife mit dem Timer realisieren, so kann diese für eine Zeit von etwa 5 ms so aussehen:

```
LDA #05
STA 0817
WAIT BIT 0817
BPL WAIT
```

Das Schreiben an die Adresse 0817 lädt den Wert 05 in den 8-Bit-Timer und stellt gleichzeitig den Vorteiler auf 1024 ein, d.h. ein Timer-Zyklus entspricht dann 1,024 ms. Ein Interrupt wird nicht erzeugt, sondern es wird lediglich das höchstwertige Bit an der Adresse 0817 gesetzt, das mit dem BIT-Befehl überprüft werden kann. Diese Prüfung kann immer bei 0817 erfolgen, unabhängig davon, ob man die gewünschte Zeit bei 0814, 0815, 0816 oder 0817 einschreibt

und damit einen Timer-Zyklus von 1, 8, 64 oder 1024 µs wählt. An der Adresse 0816 könnte man auch den Timer-Inhalt auslesen, wovon hier allerdings kein Gebrauch gemacht wird. Der Timer zählt ab dem Einschreiben der gewünschten Zeit mit dem eingestellten Vorteiler-Faktor abwärts; sobald er Null erreicht, wird Bit 7 in 0817 gesetzt, und der Timer zählt mit dem Vorteiler-Faktor 1 weiter abwärts.

Möchte man zwei Vorgänge unabhängig voneinander vom EMUF bearbeiten lassen, z. B. eine Frequenz an einem Port erzeugen, während ein anderes Hauptprogramm läuft, so kann man dazu – vom Timer gesteuert – periodisch ein Interrupt-Programm anspringen, das den Portzustand umschaltet. Für eine Ausgangsfrequenz von 100 Hz müßte man dazu 200mal pro Sekunde das Interrupt-Programm durchlaufen.

Die Tonerzeugung per Interrupt kann dann z. B. so aussehen:

```
Hauptprogramm:
LDA # 08 ;PA3=
STA 0801 ;Ausgang
LDA #05 ;ersten Timer –
STA 081F ;Interrupt
CLI ;erzeugen
weiter im Programm
```

Interrupt-Routine:

```
PHA ;Akku retten
LDA #05 ;Timer neu
STA 081F ;starten
LDA 0800 ;PA3=Ton-
EOR #08 ;ausgang
STA 0800 ;invertieren
PLA
RTI
```

Dem Leser mag auffallen, daß in der Interrupt-Routine weder das Prozessor-Statusregister mit PHP gerettet werden muß (das tut die CPU nämlich bei einem Interrupt automatisch) noch das Interrupt-Flag von CPU oder Timer beeinflußt werden muß – auch das erfolgt automatisch.

Die Adresse 081F entspricht 0817, bis auf einen Unterschied: der Timer wird veranlaßt, nach 5 x 1024 CPU-Zyklen, also wieder nach rund 5 ms, die Interrupt-Leitung $\overline{\text{IRQ}}$ auf Low-Pegel zu ziehen. Dies veranlaßt die CPU zu einem Sprung ins Interrupt-Programm. Die $\overline{\text{IRQ}}$ -Leitung bleibt so lange auf Low, bis dem Timer irgendwie mitgeteilt wird, daß der Timer-Interrupt von der CPU akzeptiert wurde, hier bis zu einem erneuten Schreiben in ein Timer-Register. Auch eine Leseoperation wäre geeignet. Da in unserem Beispiel der Timer in der Interrupt-Routine selbst immer wieder neu gestartet wird, entsteht ein periodischer Interrupt und damit ein konstanter Ton am Port-Ausgang PA3 – unabhängig davon, was das Hauptprogramm inzwischen tut.

Fe.

Jens Grysbjerg

EMUF im Senegal

Seit der vernichtenden Dürrewelle der siebziger Jahre in den Ländern der Sahel-Zone wurde eine Reihe von Maßnahmen zur Bewässerung und zum Anbau von Getreide eingeleitet. Der mc-EMUF trägt sein Teil dazu bei: Im Rahmen eines französischen Entwicklungshilfe-Projekts dient er zu automatischen Untersuchung des Bodenzustandes.

Eines der Entwicklungshilfe-Projekte der Sahel-Zone ist die Verbesserung der Reisernte im nördlichen Senegal, das regelmäßig von Überschwemmungen heimgesucht wird. Die französische Organisation ORSTOM (Office de la Recherche Scientifique et Technique Outre-Mer), die sich um dieses Projekt kümmert, nimmt langfristige Untersuchungen der Boden-Parameter vor (Na, Ca, K, pH, Redox u. a.). Die Meßstationen sind abgelegen, und die Geräte müssen die gesammelten Daten für lange Zeiträume speichern können. Früher wurde dafür eine etwas altmodische Technik verwendet: Die analogen Ausgänge der Meßwert-Aufnehmer werden mit einem motorgetriebenen Schalter als Multiplexer auf einem kleinen Kurven-

schreiber aufgezeichnet. Ein solches System ist aber teuer, verbraucht viel Strom und ist nicht gerade zuverlässig. Schlimmer noch ist, daß bei einer späteren Datenerfassung mit dem Computer alle Daten von der Aufzeichnung abgelesen und von Hand eingegeben werden müssen – eine langwierige, frustrierende und fehleranfällige Prozedur. Es ist naheliegend, die Aufgabe einem Mikrocomputer zu übertragen, der alle Daten vorübergehend speichert und dann zu einem zentralen Computer übertragen kann. Allerdings durften die Entwicklungshilfsmittel nicht allzu teuer sein; die Arbeit stand unter Zeitdruck und mußte lokal in Dakar durchgeführt werden. Das führte zu folgendem Konzept:

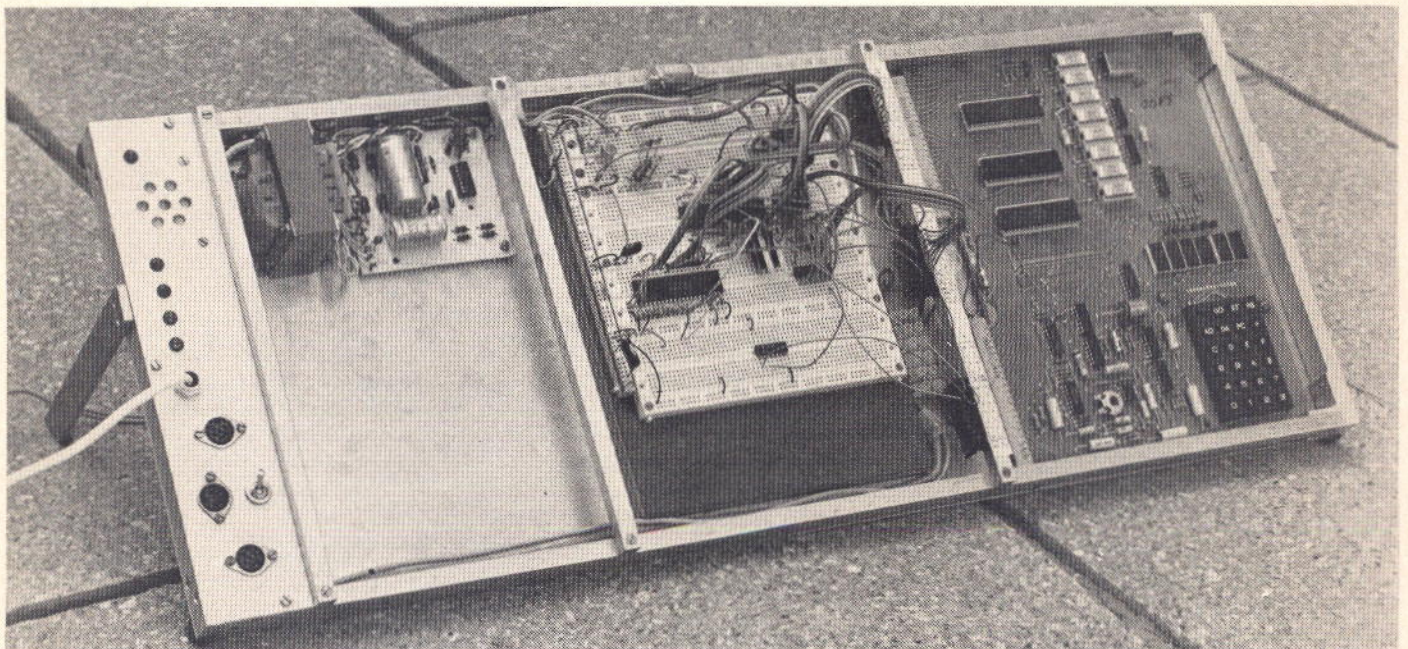
- Die Meßwertaufnehmer des alten Systems wurden übernommen.
- Als Multiplexer und A/D-Wandler dient ein IC (ADC 0817).
- Als Mikrocomputer wird der mc-EMUF verwendet (Preis ca. 100 DM).
- Die Daten werden auf gewöhnlichen Tonbandkassetten gespeichert.

Während eines Meßzyklus liest der EMUF 16 Analogkanäle und speichert die Werte im RAM. Dann werden die Meßwerte auf einen Kassettenrecorder ausgegeben, wobei das Format dem des bekannten KIM-1 entspricht – langsam, aber sicher (135 Bit/s).

Das Programm wurde auf einem AIM-65 entwickelt und teilweise getestet. Der endgültige Test wurde mit einem KIM-1 durchgeführt (*Bild*), der in seiner Hardware-Struktur dem EMUF recht ähnlich ist.

Der A/D-Wandler fand schließlich auf der Lochrasterfläche des EMUF Platz. Diese EMUF-Applikation ist ein Beispiel dafür, wie ein einfaches und preiswertes Mikrocomputer-System Probleme lösen kann, für die sonst eine komplexe Logik nötig wäre. Sie zeigt, daß man bei der Entwicklung elektronischer Meßschaltungen immer den Mikroprozessor in seine Überlegungen einbeziehen sollte.

Der Autor, ein dänischer Ingenieur, hat die EMUF-Applikation für ORSTOM entwickelt und ist Experte der UNESCO.



Versuchsaufbau während der Entwicklungsphase; links ein Netzteil, in der Mitte der A/D-Wandler und Multiplexer, rechts der KIM-1, der dann durch den EMUF ersetzt wurde

RPB 6
Antennen für Rundfunk- und Fernseh-Empfang. Theoretische Überlegungen und gebräuchliche Ausführungsformen.
(Mende)
DM 9.80 ISBN 3-7723-0066-9

RPB 20
Methodische Fehlersuche in Rundfunkempfängern. Meßmethoden und Fehlersuche durch Signalverfolgung und Signalzuführung. (Renardy/Lummer)
DM 9.80 ISBN 3-7723-0205-X

RPB 30
UHF-Amateurfunk-Antennen. Theorie, Dimensionierung und praktischer Nachbau für das 70-, 23- und 13-cm Amateurfunkband. (Reithofer)
DM 12.80 ISBN 3-7723-0302-1

RPB 33
Elektronische Voltmeter. Grundlagen und Praxis der elektronischen Voltmeter und Multimeter. (Limann/Pelka)
DM 12.80 ISBN 3-7723-0339-0

RPB 40
Fachwörter der Elektronik. Heiße Definitionen neuester Elektronik-Begriffe. (Franz)
DM 6.80 ISBN 3-7723-0403-6

RPB 50
Praktischer Antennenbau. Ein Ratgeber für Entwurf und Ausführung von Antennenanlagen aller Rundfunkwellenbereiche. (Mende)
DM 9.80 ISBN 3-7723-0508-3

RPB 51
Kleine Fernseh-Bildfehler-Fibel. Typische und häufige Bildfehler anhand von Schirmbildaufnahmen erklärt. (Gies/Kirsch)
DM 9.80 ISBN 3-7723-0513-X

RPB 64
Einführung in die Operationsverstärker-Technik. Ein Wegweiser, Aufbau, Arbeitsweise und Eigenschaften der Operationsverstärker besser zu verstehen. (Nüßmann)
DM 9.80 ISBN 3-7723-0643-8

RPB

electronic-taschenbücher bieten
die Summe des Elektronikwissens
für Beruf und Hobby.

RPB 90
Netzgeräte mit ICs. 21 praktische Schaltvorschlüsse ausführlich beschrieben. (Sehrig)
DM 6.80 ISBN 3-7723-0903-8

RPB 99
Wie arbeite ich mit dem Elektronenstrahl-Oszilloskop? Eine Fibel der Oszilloskoptechnik nebst einer umfangreichen und universellen Betriebsanleitung für Amateure und Praktiker. (Sutner/Wißler)
DM 12.80 ISBN 3-7723-0991-7

RPB 105
Lautsprecher und Lautsprechergehäuse für HiFi. Der Selbstbau hochwertiger Lautsprecheranlagen. (Klinger)
DM 12.80 ISBN 3-7723-1051-6

RPB 106
Netztransformatoren und Drosseln. Daten, Tabellen, Berechnungsbeispiele sowie neue Bauformen vom Kern bis zum Zusammenbau. (Klein)
DM 12.80 ISBN 3-7723-1065-6

RPB 112
Das Löten für den Praktiker. Beherrschenswerte Regeln für den Anfänger – nützliches Grundwissen für den Profi. (Strauß)
DM 9.80 ISBN 3-7723-1121-0

RPB 146
Halbleiterspeicher. Eine Kurz-Darstellung der Halbleiterspeicher von den Grundlagen bis zur Anwendung. (Bonerz)
DM 9.80 ISBN 3-7723-1461-9

RPB 149
Kondensatorenkunde für Elektroniker. Eine ausführliche Darstellung der Kondensatoren und ihrer Kennwerte, Bauformen und speziellen Eigenschaften, Anwendungsbeispiele und Kennzeichensysteme. (Leucht)
DM 12.80 ISBN 3-7723-1491-0

RPB 154
KW-Amateurbildfunk SSTV und FAX. Technische Grundlagen – Nachbaupraxis – Betriebstechnik. (Pietsch)
DM 12.80 ISBN 3-7723-1541-0

RPB 157
Meßgeräte und Meßverfahren für den Funkamateure. Auch einfache Meßgeräte bringen genügend genaue Meßergebnisse. (Link)
DM 9.80 ISBN 3-7723-1573-9

RPB 158
Sensible Sensoren. Elektronische Meßwertnehmer-Prinzipien und Anwendungsbeispiele. (Limann)
DM 9.80 ISBN 3-7723-1581-X

RPB 159
Die logisch gesteuerte Modelleisenbahn. Eine Großanlage wird mit neuartigen Bauelementen und Schaltungen sowie mit Mikroprozessoren durchautomatisiert. (Platerink)
DM 12.80 ISBN 3-7723-1591-2

RPB 160
Relais. Grundlagen, Bauformen und Schaltungstechnik. Relaiskunde für den praktischen Elektroniker. (Köhler)
DM 12.80 ISBN 3-7723-1602-6

RPB 161
HiFi-Lautsprecher-Kombination. Schaltbeispiele für Frequenzweichen mit 68 Lautsprecher-Kombinationen. (Klinger)
DM 9.80 ISBN 3-7723-1612-3

RPB 171
Halbleiter-Schaltungstechnik einfach dargestellt. Wie Halbleiter-Schaltungen theoretisch erkannt, praxisnah ausgewertet und auf Fehler untersucht werden können. (Benda)
DM 12.80 ISBN 3-7723-1714-6

RPB 173
Anwendungsbeispiele für den Mikroprozessor 6502. Hardware-Tips und nützliche Programmbeispiele in Maschinensprache. (Feichtinger)
DM 9.80 ISBN 3-7723-1731-6

RPB 175
Infrarot-Elektronik. Eine Einführung in die Infrarottechnik mit Hobbyschaltungen und Experimenten. (Schreiber)
DM 12.80 ISBN 3-7723-1751-0

RPB 177
Energiesparende Heizungsregelung. Planung und Aufbau einer witterungsgesteuerten Vorlauftemperaturregelung. (Rapp)
DM 12.80 ISBN 3-7723-1771-5

RPB 182
Aktive Antennen für DX-Empfang. Theorie, Selbstbau-Praxis. (Best)
DM 9.80 ISBN 3-7723-1821-5

RPB 185
VMOS-Schaltungen. VMOS-Bausteine im NF-Bereich, in Signalkreisen sowie bei Tongenerator- und Steuerschaltungen. (Penfold)
DM 12.80 ISBN 3-7723-1851-7

RPB 186
MOS-Leistungstransistortechnik. Aufbau – Schaltungen – Anwendung. (Schreiber)
DM 12.80 ISBN 3-7723-1861-4

RPB 300
Kfz-Elektronik im Selbstbau. Von Warn- und Schutzschaltungen, von elektronischen Zündungen und anderen interessanten Sachen rund um das Kraftfahrzeug. (Jansen)
DM 9.80 ISBN 3-7723-3003-7

RPB 302
Electronic-Pianos und Synthesizer. Nach industriellen Gesichtspunkten entworfene, jedoch für den Selbstbau geeignete Schaltungen. (Tünker)
DM 12.80 ISBN 3-7723-3022-3

RPB 311
Lautsprechergehäuse-Baubuch. Bauzeichnungen für 90 verschiedene Lautsprecherboxen. (Klinger)
DM 12.80 ISBN 3-7723-3114-9

RPB 336
Musikelektronik. Elektronische Schlagzeuge, Sound-Organen, Glockenspiele, Metronome selbstgebaut. (Tünker)
DM 12.80 ISBN 3-7723-3364-8



Sie sind modern, handlich und preiswert.

RPB 65
Operationsverstärker-Anwendung. Ein Wegweiser zur Verwirklichung eigener Ideen mit dem „Bauelement“ Operationsverstärker. (Hirschmann)
DM 12.80 ISBN 3-7723-0653-5

RPB 82
Was ist ein Mikroprozessor? Über die Arbeitsweise, Programmierung und Anwendung von Mikrocomputern. (Pelka)
DM 9.80 ISBN 3-7723-0825-2

RPB 83
Mikrofon-Aufnahmetechnik. Von Mikrofonen, ihren Qualitätsmerkmalen und Anschlußproblemen zur zweckmäßigsten Aufnahmetechnik. (Praetzel/Warneke)
DM 9.80 ISBN 3-7723-0832-5

RPB 84
Fernsehantennen-Praxis. Ein zuverlässiger Leitaden zum besten Fernsehempfang. (Mende)
DM 9.80 ISBN 3-7723-0844-9

RPB 87
Methodische Fehlersuche in der Industrie-Elektronik. Wie Fehler in elektronischen Geräten und Anlagen durch zielbewußte Systematik und Logik geortet werden können. (Benda)
DM 9.80 ISBN 3-7723-0872-4

RPB 127
Schaltungsanwendungen der Optoelektronik. Einfache Selbstbau-Schaltungen, die auch ein Anfänger nachvollziehen kann. (Oehmichen)
DM 9.80 ISBN 3-7723-1271-3

RPB 130
Solar-Zellen. Kennwerte, Schaltungen und Anwendung. (Juster)
DM 9.80 ISBN 3-7723-1301-9

RPB 134
Kleines Halbleiter-ABC. Ein kleines Nachschlagewerk über Aufbau, Eigenschaften und Funktion der wichtigsten Halbleiterbauteile. (Büscher/Wieglmann)
DM 12.80 ISBN 3-7723-1344-2

RPB 136
Transistorisierte Netzgeräte. Spannung und Strom geregelt durch Halbleiter. (Strobel)
DM 6.80 ISBN 3-7723-1366-3

RPB 138
Kleines HiFi-ABC. 1210 definierte Fachausdrücke aus der HiFi-Technik. (Fellbaum/Loos)
DM 12.80 ISBN 3-7723-1381-7

RPB 142
Kleine HiFi-Stereo-Praxis. Eine HiFi-Stereo-Fibel mit ausführlichen Bauanleitungen für jedermann. (Knobloch)
DM 9.80 ISBN 3-7723-1421-X



Sie sind modern, handlich und preiswert.

RPB 165
BIFET – BIMOS – CMOS in Feldeffekt-Operationsverstärkern. Die Eigenschaften der FET-Operationsverstärker und ihre zahlreichen Anwendungsmöglichkeiten. (Schreiber)
DM 12.80 ISBN 3-7723-1651-4

RPB 167
Diavertonung. Regie und Technik der elektronisch gesteuerten Tonbildschau. (Tollmien)
DM 12.80 ISBN 3-7723-1671-9

Rund 120 RPB electronic-taschenbücher bieten Ihnen Information, Tips und Ratschläge. Diese Anzeige zeigt eine Auswahl.

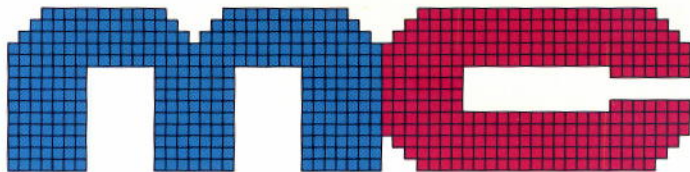
Bitte fordern Sie das kostenlose Gesamtverzeichnis unter der Bestellnummer P 254 an.



Franzis-Bücher erhalten Sie durch jede Buchhandlung sowie in den einschlägigen Fachhandlungen. Bestellungen auch an den Verlag.

Franzis

der große Fachverlag für angewandte Elektronik und Informatik



ist die Mikro- computer- Zeitschrift

Als professioneller Mikrocomputer-Anwender kommen Sie um mc nicht herum. Warum? Hier ein paar Themen aus dem April-Heft von mc:

Z80-Einplatinencomputer für weniger als 100 DM

Nach dem mc-6504-EMUF, dem bisher preiswertesten Einplatinencomputer, veröffentlicht mc jetzt eine Z80-Version, selbstverständlich mit Layout-Vorschlag: Bis 32 I/O-Leitungen, 2 KByte RAM, 4...8 KByte EPROM, ideal für Ihre Steuerungs- und Interface-Anwendungen.



Das mc-Netz

Von lokalen Netzwerken spricht heute jeder. Ethernet or not, könnte man manchmal meinen – aber es gibt preiswertere Methoden. mc beschreibt ein einfaches Zweidraht-Netzwerk für bis zu 254 Geräte, das sich mit geringem Aufwand realisieren läßt. Die Kollision von Daten, die andere Netzwerke krampfhaft vermeiden, wird hier zum obersten Prinzip erhoben.

Tests: IBM-PC, Commodore-64, CT-65

Drei völlig unterschiedliche, aber für ihre Klasse höchst interessante Computer testet mc im April: den Personal Computer von IBM, den „aufgebohrten Volkscomputer“ C-64 und den Einplatinen-Computer CT-65.



Außerdem...

...finden Sie im April-Heft von mc den nächsten Teil der CP/M-Einführungsserie, eine wissenschaftlich fundierte Abhandlung über die richtige Computerauswahl, eine Zusammenstellung der Hersteller von Einplatinen-Computern – und auch sonst eben genau die Mischung, die mc zur Zeitschrift für Mikrocomputer-Profis machte.

**Wenn Sie mc näher kennenlernen wollen, machen Sie bitte
von unserem „Kennenlern-Angebot“ Gebrauch.
Nebenstehend finden Sie die entsprechende Abrufkarte.**

Franzis-Verlag, Karlstraße 37, 8000 München 2, Telefon 0 89/51 17-2 79/-2 40

BASIC
für blutige

FORTH
Handbuch

GRUNDKURS

Programmiere mit dem CBM

Rund um den Atari

HOFACKER

Ing. W. Hofacker GmbH, Tegernseer Str. 18, 8150 Holzkirchen, Tel. (08024) 7331

Lieferung durch den Fach- und Buchhandel oder per Nachnahme oder Vorkasse, Postscheck-Kto. Mohn 15 994-807 oder Eurocheck, Eurocard, Preise inkl. MwSt., zuzügl. Porto und NN-Gebühr. Unverbindliche Preisempfehlung. Angebot freibleibend. Zwischenverkauf vorbehalten.

Nr. 139
19,80 DM

Nr. 108
29,80 DM

Fachbücher, Software

BASIC Bücher

139 BASIC für blutige Laien **NEU** 19,80

113 BASIC Handbuch für Anfänger 19,80

121 Microsoft BASIC HB 29,80

122 BASIC für Fortgeschrittene 39,-

31 57 Praktische BASIC Programme 39,-

34 TINY BASIC Handbuch 19,80

256 BASIC / BASIC 39,-

256 Stimulating Simulations 19,80

257 BASIC Computer Programs in Science and Engineering 39,-

260 BASIC Computer Progr. Business 1 39,-

284 BASIC Computer Progr. Business 2 49,-

266 Advanced BASIC Applications 39,-

151 Microsoft BASIC 9,80

270 BASIC with Style 39,-

27 Basic-M/Motorola 6800/09/68000 29,80

HOFACKER Bücher

1 Transistor-Berechn. u. Bauanl. HB 29,80

2 TB8, Band 2 19,80

284 i. Auto m. HB f. Polizei-Radar 9,80

4 IC-Handbuch (TTL, CMOS, Linear) 19,80

5 IC-Datenbuch 9,80

8 IC-Bauschaltungen-Handbuch 19,80

9 Feldeffekttransistoren 9,80

10 Elektronik und Radio, IV 9,80

11 IC-NF-Verstärker 9,80

12 Beispiele integrierter Schaltungen 19,80

13 Hobby-Elektronik-Handbuch 9,80

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 Digitaltechnik Grundkurs **NEU** 19,80

22 Mikroprozessoren 19,80

23 Elektronik Grundkurs 9,80

24 Progr. in Z80 Masch.-Spr. II **NEU** 29,80

26 Mikroprozessor Teil II 19,80

28 Microcomputer Lexikon 29,80

29 Microcomputer Datenbuch 49,-

137 FORTH Handbuch (deutsch) 49,-

132 CP/M-Handbuch 19,80

ELCOMP Books

150 Care a. Feeding of the Comm. PET 9,80

152 Expansion Handb. f. 6502 u. 6800 19,80

154 Complex Sound Gen. w. Microc. 9,80

155 The First Book of 80 US 29,80

156 Small Business Programs 29,80

157 The First Book of Ohio 19,80

158 The Second Book of Ohio 19,80

159 The Third Book of Ohio 19,80

160 The Fourth Book of Ohio 19,80

161 The Fifth Book of Ohio 19,80

162 Atari Games in BASIC 19,80

163 The Periph. Handbook 29,80

164 Atari Progr. - Learning by Using 19,80

170 FORTH on the Atari **NEU** 29,80

172 Hackerbook I (Atari) 29,80

173 Description Book, PD-Book 9,80

175 Astrologie with 48K, Atari 800 49,-

177 CP/M-MBASIC and the Osborne-I 29,80

TAB Bücher

952 Microcomp. Progr. f. Hobbyist 39,-

1015 Beginner's Guide to Microproc. 29,80

1055 The BASIC Cookbook 24,80

NEU * NEU *** NEU *** NEU**

1496 The IBM Personal Computer 49,80

1256 33 Games of Skill & Chance for the IBM PC 59,-

1556 Graphics Programs f. the IBM PC 69,-

1540 100 Ready-to-Run Progr. and Subroutines for the IBM PC 79,-

1545 Advanced Progr. Techniques for your Atari, Graphics & Voice Progr. 89,-

1453 Progr. your Atari Computer 49,-

1485 Atari Progr. w. 55 Programs 59,-

1513 APPLE II BASIC 39,-

1394 Progr. your Apple II Computer 39,-

1491 101 Projects for the Z80 69,-

1421 Handbook of Advanced Robotics 69,-

1391 Projects i. Machine Intelligence for your Home Computer 49,-

1455 The Art of Computer Progr. 49,-

1468 Verbal Control w. Microcomp. 49,-

1422 Writing Basic Adventure Progr. for TRS-80 39,-

1481 TRS-80 Color Programs 99,-

und Hardware Add-ons

IJG Bücher und Software

240 TRS-80 Disk & other Mysteries 69,00

245 Microsoft BASIC Decoded 89,00

246 BASIC Faster and Better 129,00

247 The Custom TRS-80 129,00

283 The Captain 80 Book of BASIC Adventures 79,00

681 Machine Language Disk I/O 129,00

5125 TRSDOS 2.3 Decoded 129,00

5126 How to do it on the TRS-80 129,00

680 The Custom APPLE 79,00

Weitere Neuerscheinungen von IJG sind:

5127 BASIC Disk I/O Faster and Better 129,00

5128 The TRS-80 Beginners Guide to Games and Graphics 99,00

Software von IJG

5208 Electric Pencil (D) 295,00

5209 Electric Pencil (C) 285,00

5210 Red Pencil - Roter Bleistift (D) 295,00

5203 Blue Pencil Dictionary (D) 295,00

5204 BASIC Faster and Better Demonstration Disk (BFDDEM) (D) 79,00

5205 BASIC Faster and Better Library Disk (BFLIB) (D) 79,00

5206 CYBERCHESS, Amateurs (D) 129,00

5207 CYBERCHESS, Professional (D) 129,00

Endlich ist er da!

150 S. vollgepackt m. neuen Büchern für Elektronik und Microcomputer. Software für Osborne, Commodore 64, PET/CBM, VC-20, SINCLAIR, TRS-80, GENIE, APPLE II

Heute noch bestellen!

DM 2,- in Briefmarken od. Vorkasse a. Postscheck-Kto. Mohn 15 994-807

für die bedeutenden Personalcomputer

ATARI 400 / 800

7051 Spielesammlung, 4 Spiele (C) 49,-

7052 Spielesammlung, 4 Spiele (D) 59,-

7223 Astrologie / Atari 800, 48k (D) 99,-

7209 Gunfight (Cowboykampf) (C) 79,-

7008 Roter Baron Luftkampf (C) 79,-

7007 Submarine Minefield (C) 49,-

7008 Down the Trench (C) 79,-

7009 Panzerkrieg (C) 49,-

7010 WUMPUS Adventure, 16K (C) 69,-

7011 WUMPUS Adventure, 24K (C) 79,-

7012 Schnuppercassette (8/16K) (C) 49,-

7019 Einfache Spiele in BASIC (C) 19,80

7004 3-D-Comp. Grafik (D) 159,-

7315 Biorythmus (D/C) 49,-

7309 Mondphasen (D) 49,-

7328 SARGON II - Schach (D) 129,-

7325 Flipper Buldog Pinball (C) 99,-

7326 GO-Spiel (D) 99,-

7343 Superdisk-Pack 8 Disk-Seiten voll Progr. (D) 699,-

Geschäftsprogramme

7212 ATEXT-1 (C) 148,-

7211 ATEXT-1 (D) 159,-

7210 ATEXT-1 (Rom-Modul) 199,-

7214 Lagerverwaltung (C) 49,-

7215 Lagerverwaltung (D) 59,-

7021 Adressverwaltung (C/D) 99,-

7020 Rechnungen schreiben (C) 99,-

7002 BASIC Texteditor (C/D) 99,-

7320 SUPERINVENTORY (D) 149,-

7312 SUPERMAILING **NEU** (D) 99,-

7312 BUSIPACK (D) 499,-

Maschinensprachenmonitore

168 How to Progr. i. Machine Lang. on the Atari (Book) 29,80

170 FORTH - Learn. b. Using (Book) 29,80

7022 ATMONA-1 (Ma-Monitor) D/C 49,-

7023 Progr. i. Maschinensprache (C) 49,-

7049 Supertrac (D/C) 49,-

7045 ATMONA-1 (Cartridge) 99,-

7098 ATAS-1, 32K RAM (C) 99,-

7998 ATAS-1, 48K RAM (C) 99,-

7099 ATMAS-1 Macroass., 48K (D) 299,-

7060 ATMAS-1 (Rom-Modul) 399,-

7050 ATAS-1 Macroass. m. Include 399,-

7053 Lern-FORTH (C/D) 79,-

7055 ELCOMP-FORTH (D) 199,-

VC-20

4883 Prof. Adressverw. (8/16k RAM) 99,-

4892 Prof. Textverarb. (16k RAM) 149,-

4896 Miniassembler (8k RAM) 49,-

4899 Kräuterprogramm 49,-

4884 BASIC-UTILITY-Program. BUTI 199,-

4894 Füllhorn-Spiel (8K) 19,80

4895 SNAKE-Fressen (8K) 19,80

4881 Tennis, Squash, Breakout (8K) 29,80

4901 Astrologieprogramm 49,-

4902 BUCKATTACK 29,80

478 VC-20 Games-Paket (engl.) 99,-

4903 Haushaltsfinanzen (engl.) 179,-

4897 VC-Mona (Grundversion) 19,80

4828 Spielesammlung f. VC-20 49,-

4843 16k Speichererw., o. Bauteile 149,-

4844 Universal Experimentierplatte 89,-

4846 Schalterinterface 199,-

4847 Stecker für USER PORT 19,80

4848 Stecker für Erweiterungsport 19,80

141 Programme für VC-20 (Buch) 29,80

31 57 Praktische Programme 39,-

128 Programmieren mit dem CBM 29,80

130 Programme für CBM 19,80

291 VIC Revealed 59,-

NEU SUPER-AKTION SPIELE

4913 APPLE PANIC (Rom-Modul) 149,-

4914 CHOLIFTER (Rom-Modul) 149,-

4915 ASTROBLITZ (Rom-Modul) 149,-

4916 SERPENTINE (Rom-Modul) 149,-

4917 TRASHMAN (Rom-Modul) 149,-

BASIC-Programmbibliothek, besteht aus insgesamt 7 Büchern (DIN A4), ca. 200 Progr. a. den Bereichen, Geschäftsprogr., Mathematik, Statistik, Spiele, u.v.a. nur 449,-

(Die Bände sind auch einzeln erhältlich!)

Hardware Zusätze für ATARI 400 / 800

7208 EPSON-Interface, Platine 49,-

Anleitung + Software (C/D) 59,-

7291 RS-232 Interface (C/D) 99,-

7293 EPROM-Platine, leer 19,80

7224 EPROM-Platine, Bausatz 49,-

7243 EPROM-Platine, fertig 65,-

7292 EPROM-Brenner, Platine mit Anleitung + Software 129,-

172 Hackerbook I, Util., Tips, Tricks 29,80

173 Program-Descriptions. Ein Büchlein m. Progr. Besch. unserer besten Progr. 9,80

APPLE II

The Custom Apple & other Mysteries

Dieses Buch braucht jeder APPLE-Besitzer. Ca. 190 Seiten Großformat voll mit Hardwareinformationen u. Platinenvorlagen, Data-Aquisition, I/O-Progr., EPROM-Burner, u.v.a. **Best.-Nr. 680 79,- DM**

6153 Lern FORTH (C) 79,-

6155 ELCOMP-FORTH (D) 199,-

6118 Schach - SARGON (D) 119,-

6126 Adressverwaltung (D) 199,-

6127 Datenverwaltung (D) 199,-

6136 Game Package (D) 69,-

Leerplatinen mit Anleitung

604 Universal Experimentierplatte 59,-

605 Ein-/Ausgabe Platine 89,-

606 Bus Expansion ELCOMP-1 129,-

607 EPROM Platine 149,-

608 MSK Platine 89,-

609 EPROM-Karte f. 2716 54,-

610 Analog-Digitalw.-Karte 149,-

611 6502 Rechnerkopplung 249,-

ELCOMP - Fachzeitschrift f. Microcomputer

Sonderpreis f. zurückl. ELCOMP-Hefte:

Sept. 79 - Sept. 79, außer Nr. 2, 4-79 33,-

Jahrgang 1981, außer Nr. 2, 3, 10 35,-

Jahrgang 1982 - komplett 49,-

ELCOMP 1/83 (über 200 Seiten) 29,80

SONDERPOSTEN

350 10 Creative Computing Hefte gem. 28,-

351 20 Creative Computing Hefte gem. 42,-

354 10 Dr. Dobbs Hefte gemischt 49,-

355 3 6502 User Notes Hefte 19,80

356 8048 Microcomputer Handbuch 9,80

420 Schach f. CBM + PET 2000/3000 79,-

426 Textverarbeitung CBM/PET 96,-

8020 Dr. Dobbs Sammelband, Vol. 1 29,80

SINCLAIR ZX81 / ZX-Spectrum

Progr. i. BASIC u. Maschinensprache m. d. ZX81

Endlich ein dt. Progr.-Handb. für den Sinclair ZX81. Viele Tricks, Tipps, Hinweise, Progr. in Maschinenspr., Hardware-Erweiterung, lustige Spielprogramme zum Eintippen. **Best.-Nr. 140 29,80 DM**

143 35 Programme für den ZX81 29,80

119 Progr. i. Masch.-Spr. m. Z80, I 39,-

24 Progr. i. Masch.-Spr. m. Z80, II 29,80

252 Z80 Reference Karte 5,-

8029 Z80 Assembler Handbuch Erkl. der Maschinenbefehle 29,80

2400 Adapterplatine f. ext. Experim. 39,-

604 Ext. Experimentierplatte nur zus. mit 2400 verwendbar 59,-

NEU * NEU *** NEU *** NEU**

108 Rund um den Spectrum 29,80

144 Mehr als 33 Progr. f. den Spectrum 29,80

2401 Externe I/O u. Experimentierpl. 89,-

2402 Alle Progr. aus Buch Nr. 108 auf Cassette (Spectrum) 79,-

WUNDERTÜTE

Diese besteht aus Büchern und Zeitschriften. Aus Lagerbeständen älterer Bücher u. Zeitschriften von mind. 50 versch. Publikationen (Compute, Elcomp, 80US, Sync ...). Bitte haben Sie Verständnis, daß wir Wünsche bzgl. d. Zusammensetzung nicht erfüllen können. Dafür stimmt der Preis!

Nur 35,- DM pro Paket

Lieferung p. NN od. Vorkasse auf Postscheckkonto m. Stichwort: Wundertüte 1/83.

Leercassetten - C 10 -

8089 1 Cassette 3,50

8100 10 Cassetten 29,80

8096 100 Cassetten 249,-