

# Schneiderware # 8

Unsere Aktivitäten der ersten Schneiderstunde gehen unvermindert weiter. Mittlerweile umfaßt die Hardwareserie der »SCHNEIDERWARE« die stattliche Anzahl von sieben Erweiterungskarten. Die Wartezeit seit der letzten Karte hat sich meiner Meinung nach sicher gelohnt. Ihre Geduld wird in diesem Beitrag mit einer Speichererweiterung mit ganz besonderen Features belohnt werden.

## Jede Menge Speicher

Einige Leser unter Ihnen werden sich beim Lesen des Wortes Speichererweiterung einen lustlosen Seufzer wohl nicht verkneifen können, der ausdrücken soll, daß es Speichererweiterungen für den Schneider wie Sand am Meer gibt. Das muß ich in jedem Falle mit einem »leisen« Ja beantworten. Aber was nützt eine Karte, die ich mit Eproms betreiben kann, kann mir aber keine Software dazu programmieren. Ich brauche beides nicht, werden Sie sagen – das mag auch bedingt richtig sein. Aber dann werden Sie nie in den äußerst befriedigenden Genuß kommen können, zu erleben, wie man ein Eprom dazu bringt, seinen Inhalt (Programm) nicht nur durch Auslesen herzugeben, sondern ihm während des Betriebs-Bytes »unterjubeln« zu können, was das Eprom auch anstandslos und ohne zu murren akzeptiert. Daß die dazu erforderliche Hardware die einer einfachen Piokarte bei weitem übersteigt, dürfte wohl jedem von Ihnen klar sein. Aus diesem Grunde müssen wir uns wohl wieder mit den unvermeidlichen Grundlagen befassen. Ich teile diesen Abschnitt wegen der breiten Fächerung in drei Bereiche ein: die Speicherbausteine, deren Ansteuerung im Schneider und die dazu benötigte Software.

## Was sind Speicher RAM oder ROM

Wohl jeder, der sich mit Computern beschäftigt, kann den Begriff Speicher erklären. Wie er allerdings funktio-

niert, wissen nur wenige. Ich möchte Ihnen das auf einfache Weise näherbringen. Ein Speicherelement ist ein elektronisches Bauteil, welches es fertig bringt, digitale Signale in Form von Spannungspegeln über einen längeren Zeitraum, von Steuersignale koordiniert, für sich zu behalten (WRITE-ZYKLUS). Ebenso muß es dem Computersystem diesen Inhalt auf Anforderung übergeben können (READ-ZYKLUS). Diese sogenannten »Statischen Speicher« besitzen im Inneren eine große Anzahl einzelner Speicherzellen (Flip-Flops), die matrixförmig angeordnet und verschaltet sind (siehe Bild 1). Durch diese Anordnung der einzelnen Elemente besteht die Möglichkeit, die Leitungen, die an X und Y angeschlossen sind, mittels Decoderelementen gemeinsam über wenige Leitungen anzusprechen. So wählt man jede einzelne Zelle aus dem schier unendlich scheinendem Angebot von Flipflops heraus. Die Information bleibt diesen Speichertypen

(statische Speicher) solange erhalten, wie die Versorgungsspannung eingeschaltet ist; es sei denn, man puffert die Versorgungsspannung mit einer Batterie. Eine andere Speicherart sind die dynamischen Speichertypen. Das Wort »dynamisch« verrät schon etwas vom Nachteil dieser Bausteine. Sie verlieren nicht nur nach dem Ausschalten der Versorgungsspannung ihren Inhalt, sondern tun dies fortwährend nach ein paar Millisekunden. Was nützt uns denn solch ein »Speicher«, der gar kein richtiger zu sein scheint? Hier werden die speichernden Elemente nicht durch Flipflops gebildet, sondern bestehen aus winzigen Kondensatorelementen. Die Information wird durch die Ladung des Kondensators repräsentiert, die sich aber nach einer kurzen Zeit wieder entlädt. Hier muß der Speicherinhalt durch zyklisches Lesen und Beschreiben stets aufgefrischt werden. Dieser Vorgang muß in einer sehr kurzen Zeit stattfinden, denn es dürfen die übrigen Ope-

```
&BCCE-KL INIT BACK :SUCHE UND INITIALISIERE ALLE HINTERGRUNDROMS
: AF UND BC SIND ZERSTOERT
&B90F-KL ROM SELECT :SELECTIERE EIN BESTIMMTES OBERES ROM
: C BEINNHALTET DIE ROMSELECTADRESSE
: AF ZERSTOERT
&B918-KL ROM Deselect:STELLE DIE VORIGE OBERE ROMAUSWAHL WIEDER HER
: C BEINNHALTET DIE ROM SEL. ADR. DES VORIGEN ROM
! _____:B BEINNHALTET DEN VORIGEN ROMSATE
!
! --> WURDE DURCH KL ROM SELECT UEBERGEHEN

DIESE FIRMWAREROUTINEN WERDEN IM VERWALTUNGSPROGRAMM VERWENDET
```

Tabelle 1: Kurzform der Bankingbefehle

```
: WRSROM, Romnr, Datenquelle, Ziel, Anzahl
Beispiel:
: wrsrom, &0f, &4000, &c000, &3fff :schreibt den Speicherbereich
: von &4000 bis &7fff in das ROM
: mit der Adresse &0f, beginnend
: an der Adresse &c000.
: screenup :verlegt den Bildspeicher an &4000
: screendown :verlegt den Bildspeicher an &c000
: peeklist, ROMnr, Adresse :listet ein Byte aus dem Speicher
: deren Leseadresse an ROMnr steht
: Speicher den WERT in das Ram, mit
: Romnr, und an die Adresse.
```

Tabelle 2: Vorhandene RSX-Befehle

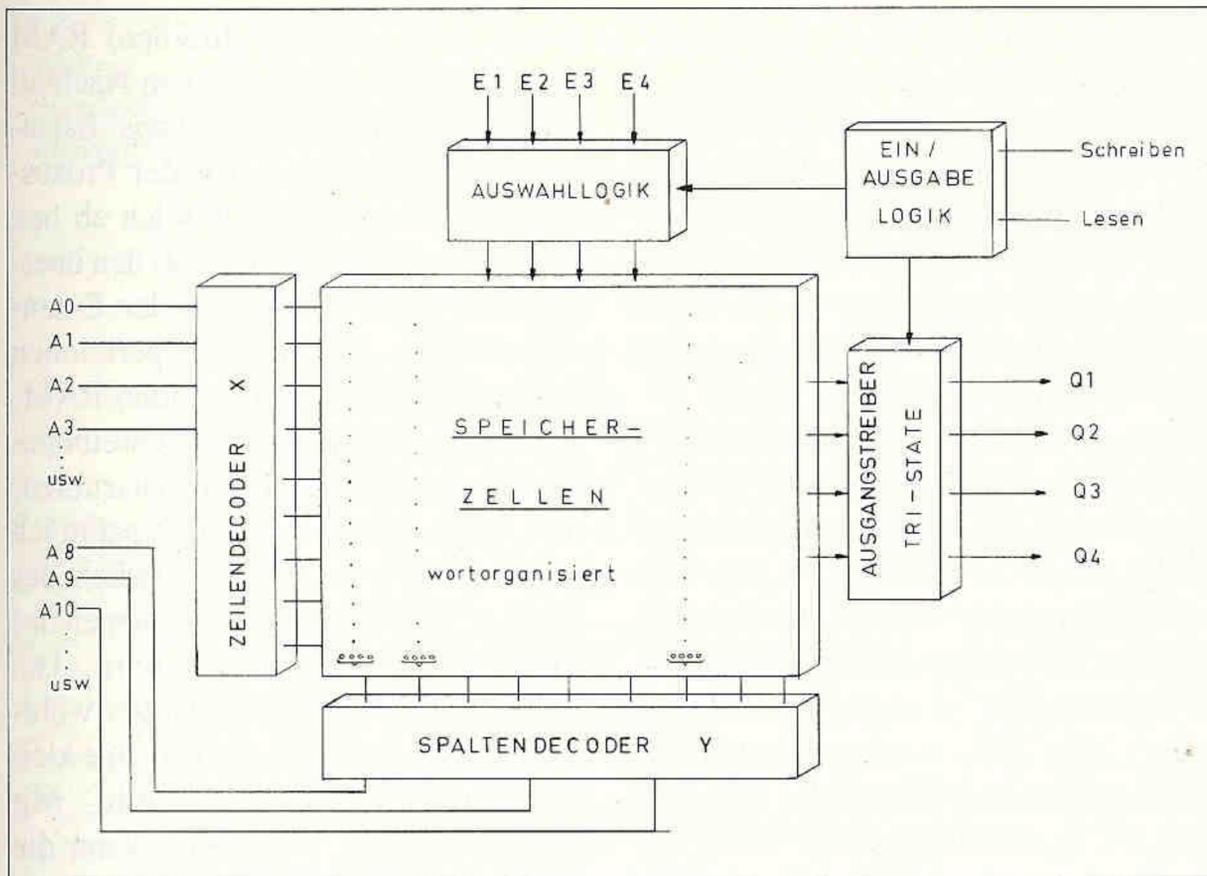


Bild 1: Schematische Speicherdarstellung

rationen der CPU nicht behindert werden. Derartige Probleme begegnen uns bei Eproms nicht. Dieses Speicherelement behält seinen einmal eingebrannten Inhalt über Jahre hinaus, was besonders für Programme, die nach dem Einschalten unseres Computers sofort zur Verfügung stehen sollen (z.B. RSX-Befehle), wichtig ist. Die Eproms (ERASABLE PROGRAMMABLE READ ONLY MEMORY), sind natürlich auch löscherbar. Dies geschieht mit ultraviolettem Licht, welches über das Quarzfenster auf den im Inneren sichtbaren Chip gelangt. Die näheren Eigenschaften dieser Bausteine werden wir in einem der nächsten Beiträge beleuchten.

### RAM und ROM – glücklich vereint!

Fassen wir nun einige gravierende Eigenschaften der für uns wichtigsten Speicherbausteine zusammen. Der Inhalt statischer Speicher ist leicht änderbar, der von Eproms nicht. Statische Speicher verlieren ihren Inhalt nach dem Abschalten der Versorgungsspannung, Eproms dagegen nicht. Der ideale Speicher für unsere Karte wäre deshalb eine Kombination aus beiden Speichertypen. Da man aber die Eigenschaften dieser Speicherelemente nicht einfach in einen

Topf werfen kann, bauen wir uns einfach eine Karte, die beide Typen enthält. Die Idee der Pseudoromkarte basiert auf der Tatsache, daß sich die Hersteller von Speicherbausteinen endlich dazu bewegen ließen, RAM-Speicher und Eproms (abgesehen von den Steueranschlüssen zum Programmieren der Eproms, die man bei RAM's ja nicht benötigt), mit gleicher Anschlußbelegung (Pinskompatibel) herzustellen. Das führte mich in Versuchung, einer Epromkarte einfach einen RAM-Baustein in den Sockel zu stecken. Nach Umverdrahten des WE (Write enable – Schreibfreigabe) – Anschlusses, der das Beschreiben der Ram's regelt und bei Eproms meistens auf 5 Volt liegt, konnte ich das RAM wie ein Eprom ansprechen. Diese Art der Speicherung klappt hervorragend.

### DECODIERUNG – anders als sonst

Die Adressdecodierung im üblichen Sinn haben wir in früheren Beiträgen schon sehr ausführlich besprochen. Da man mit einem Z 80 Prozessor, wie ihn der Schneider beherbergt, mit seinen 16 Adressleitungen nur  $2 \text{ hoch } 16 = 65536$  (64 KB) einzelner Speicherplätze adressieren kann, muß man dem System nun sagen, ob ein adressiertes Byte in einer anderen Speicher –

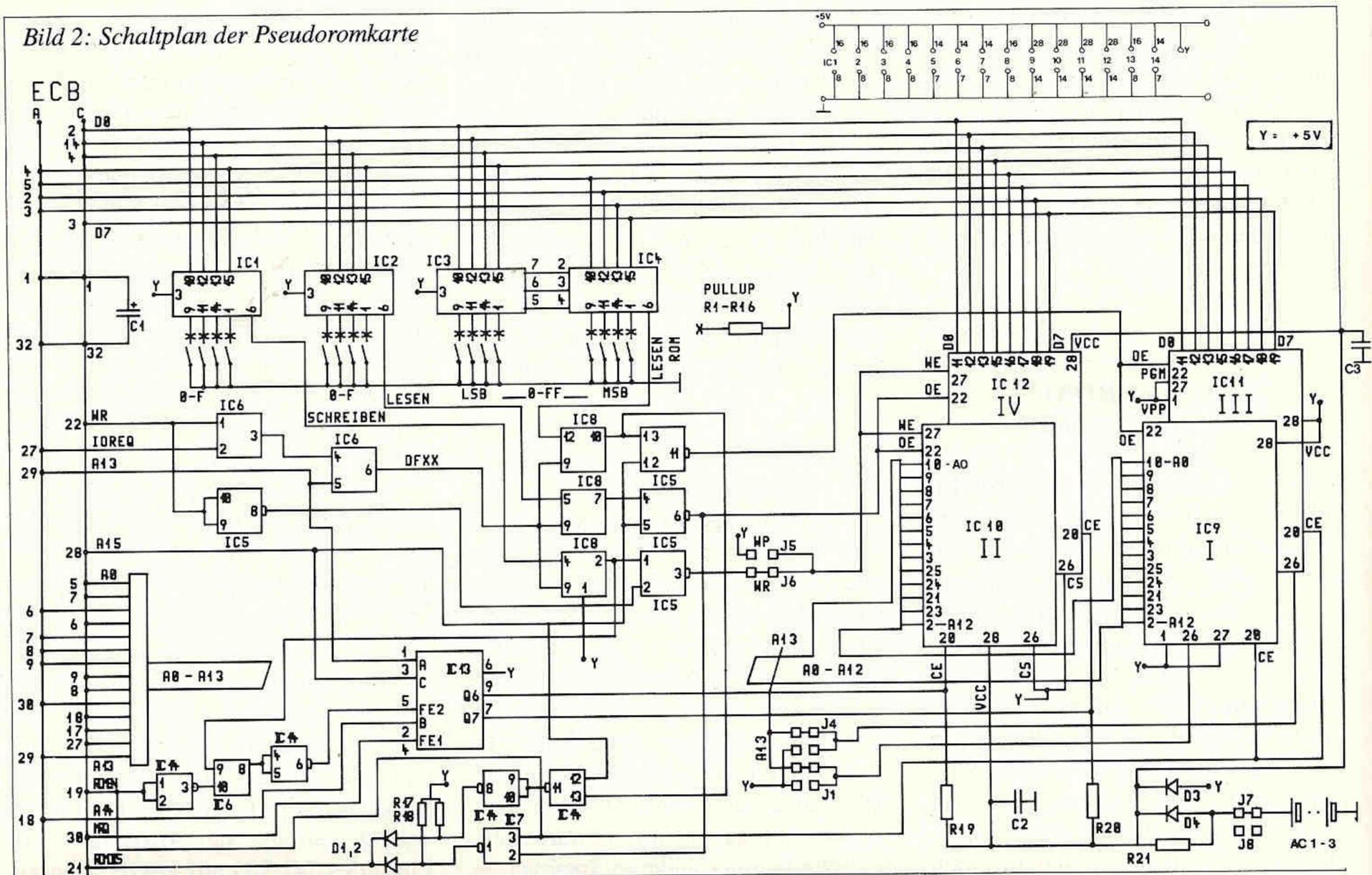
»BANK« Platz fand. Diese Methode könnte man als »Softbankingverfahren« bezeichnen. Hier wird die Freigabe der einzelnen Speicherbänke mittels Ausgabeoperationen der ext. Hardware über den Datenbus mitgeteilt. Eine andere Methode, den adressierbaren Speicherbereich zu erweitern, besteht darin, zusätzliche Adressbits hard- und softwaremäßig zu erzeugen (A17,18,19). Das erste Verfahren wurde von den »Amstrad« Designern auf die Schneider Computer angewandt. Das ROM im Floppy-interface ist hierzu das beste Beispiel. Wie Sie aus früheren Beispielen schon wissen, besitzt der Schneider parallel zum Video-RAM (Monitorspeicher) im Adressbereich von hex C000 bis hex FFFF maximal 255 adressierbare Speicherbänke zu je 16 KB. Leider wurde hier nur an die Möglichkeit des Lesens von Speichern gedacht. Die FIRMWARE (fest eingebautes Betriebsprogramm) der Schneidercomputer stellt eine Reihe von Programmchen bereit, die das Banking sehr elegant verwalten können. Tabelle 1 stellt Ihnen in Kurzform deren Vektoren (Einsprungsadressen) zusammen. Um nun z.B. die Bank #7 (gleichzeitig ROM-Nummer) ansprechen zu können (hier befindet sich AMSDOS), veranlaßt die Bankinglogik, daß ein Byte über den Datenbus mit dem Wert 7, der sogenannten SELECT-Adresse ausgegeben wird. Die Ausgabeadresse für dieses Byte ist vom Betriebsprogramm des Schneiders festgelegt und lautet &DFxx. Da sie wie alle Ausgabeadressen nicht vollständig decodiert ist, haben die niederwertigsten 8 Bits keine Bedeutung. Wie Sie aus dem Schaltbild (Bild 2) erkennen können, werden über den Datenbus D0-D7 die Selectadressen auf die 4 Bit Vergleicher vom Typ 74LS85 gegeben. Die jeweils gewünschte Adresse (IC 1 = SCHREIBEN/RAM; IC 2 = LESEN/RAM; es wird jeweils nur das untere Byte verglichen (&0-F)) kann über die DIL-Schalter oder Jumperfelder eingestellt werden, je nachdem, was Sie einbauen. Gelangt nun ein Wert an die entsprechenden Vergleicher, und dieser entspricht dem eingestellten Wert der Schalter (offen = High), so geht der Ausgang A=B Pin 6 des 74LS85 auf HIGH. Dieses

High-Signal wird allerdings nur an die Ausgänge der Flipflops in Ic 8 durchgeschaltet, wenn das entsprechende Byte über die entsprechende Ausgabeadresse &DFxx ausgegeben wurde. Als Taktsignal fungiert die decodierte Adresse &DFxx, die an IC 6=6 anliegt. Im weiteren Verlauf des Signalflusses muß noch unterschieden werden, ob eine Lese- oder Schreiboperation mit dem Speicher durchgeführt werden soll. Um das Ganze noch zu vereinfachen, bildet man das Lesesignal aus dem gesetzten Adressbit A15. Hier ist immer gewährleistet, daß der obere Adressbereich aktiv ist. Beim Schreiben genügt es, wenn das WR-Signal vom WR-Signal des Prozessors abgeleitet wird; hier allerdings über den Inverter mit umgekehrten Pegel. Über den Decoder 74 138 und den Adressbits A13 – 15 werden die Bereiche der beiden 8 KB Bausteine ausdecodiert. Da es leider noch keine 16 KB Statik-RAMs gibt, müssen wir uns mit dem Baustein 6264 (8k RAM) begnügen. Sie sind allerdings mit dem Epromtyp 2764 pinkompatibel. Über die Signale ROMEN und ROMDIS wird dem eingebautem Basicrom (das

sich ja ebenfalls in diesem Bereich befindet) gesagt, es solle sich still verhalten. Nachdem wir aber mehrere verschiedene Eproms in unserer Karte betreiben wollen, muß die Decodierung für die Eproms etwas anders aussehen. Hier leiten wir aus dem Freigabesignal ROMEN für das Interne BASIC-ROM das CE Signal ab. Immer, wenn das eingebaute BASIC-ROM selektiert wird, ist auch ein Externes aktiv. Da aber die Selectadresse des BASIC-ROMs über den Datenbus ausgegeben wird, ist der OE Anschluß der Roms noch inaktiv, d.h. die Ausgangstreiber sind gesperrt. Das ändert sich erst, wenn eine externe ROM-selectadresse über die Vergleicher (IC 3+4) ausgegeben wird, die einem angeschlossenen Eprom entspricht. Wenn an das NAND-Gatter Ic 5=13 nun ein High Signal gelangt und A15 High wird, wird dieses entsprechende Eprom selektiert. Nun kann gelesen werden. Gleichzeitig gelangt über den Deselectteil Ic 14 und über die Dioden ein Highsignal an den Anschluß ROMDIS, der das eingebaute BASIC-ROM kurzerhand ausblendet. Dies geschieht auch über Ic 7,

wenn (wie oben beschrieben) RAM gelesen wird. Einen kleinen Nachteil hat dieses Verfahren allerdings. Bei allen Leseoperationen, die der Prozessor im oberen Adressbereich ab hex &C000 ausführt, liest er aus den überlagerten Eproms (Basic oder Extension-ROM). Alle Schreiboperationen hingegen gehen in das Video-RAM. Alle Anstrengungen, bei Schreiboperationen den Bildspeicher auszublenden, schlugen fehl. Deshalb nahm ich in Kauf, daß beim Beschreiben des Pseudo-ROMs auch der Bildspeicher (Monitor) mit verändert wird. Die Leute von AMSTRAD halten wohlweislich auch für diesen Fall eine kleine Maschinenroutine bereit. Mit diesem kleinen Programm kann die Startadresse des Bildspeichers aus der Gefahrenzone transponiert werden. In diesem Fall beginnt er an der Adresse hex 4000. Dieses Verfahren wurde auch bei unserem kleinen Beispielprogramm angewandt. Die Belegung der Jumper für den Einsatz verschiedener Speicherbausteine dürfte sich durch einen Blick auf das Schaltbild von selbst erklären. In die Steckplätze der Eproms können die Typen

Bild 2: Schaltplan der Pseudoromkarte



*Listing 1:  
Hexdump des Uhrtreibers aus IO/86 auf Adresse &c000  
angepaßt mit neuer Titelzeile.*

```

1 DATA &01,&00,&00,&00,&12,&C0,&C3,&26,&C0,&C3,&33
,&C0,&C3,&16,&C1,&C3, 1679
2 DATA &43,&C1,&49,&4E,&49,&54,&B2,&57,&52,&54,&49
,&4D,&C5,&54,&49,&4D, 1580
3 DATA &C5,&44,&41,&54,&C5,&00,&F5,&C5,&D5,&E5,&CD
,&43,&C1,&E1,&D1,&C1, 2587
4 DATA &F1,&37,&C9,&FE,&07,&C0,&3E,&00,&DD,&66,&00
,&CD,&A9,&C0,&3E,&01, 1964
5 DATA &DD,&66,&01,&CD,&A9,&C0,&3E,&02,&DD,&66,&02
,&CD,&A9,&C0,&3E,&03, 1910
6 DATA &DD,&66,&03,&CD,&A9,&C0,&3E,&04,&DD,&66,&04
,&CD,&A9,&C0,&3E,&05, 1918
7 DATA &DD,&66,&05,&CD,&A2,&C0,&CD,&A9,&C0,&3E,&0B
,&DD,&66,&06,&CD,&A9, 2229
8 DATA &C0,&3E,&0C,&DD,&66,&07,&CD,&A9,&C0,&3E,&09
,&DD,&66,&08,&CD,&A9, 1938
9 DATA &C0,&3E,&0A,&DD,&66,&09,&CD,&A9,&C0,&3E,&07
,&DD,&66,&0A,&CD,&A9, 1938
10 DATA &C0,&3E,&0B,&DD,&66,&0B,&CD,&A9,&C0,&3E,&06
,&DD,&66,&0C,&CD,&A9, 1939
11 DATA &C0,&C9,&F5,&7C,&CB,&DF,&67,&F1,&C9,&00,&00
,&00,&07,&07,&07,&07, 1761
12 DATA &CB,&DF,&CB,&97,&CB,&87,&CB,&CF,&01,&E2,&FB
,&ED,&79,&7C,&01,&E3, 2716
13 DATA &FB,&ED,&79,&3E,&F4,&01,&E2,&FB,&ED,&79,&C9
,&21,&01,&A6,&16,&00, 2174
14 DATA &7A,&CD,&E1,&C0,&E6,&0F,&C6,&30,&77,&23,&14
,&3E,&10,&BA,&20,&F0, 1945
15 DATA &C9,&C5,&D5,&07,&07,&07,&07,&CB,&DF,&CD,&0C
,&C1,&CD,&0C,&C1,&CB, 2088
16 DATA &D7,&CB,&BF,&CB,&87,&01,&E2,&FB,&ED,&79,&AF
,&01,&E4,&FB,&ED,&78, 2747
17 DATA &57,&3E,&F4,&01,&E2,&FB,&ED,&79,&7A,&D1,&C1
,&C9,&C5,&F5,&3E,&0F, 2473
18 DATA &3D,&20,&FD,&F1,&C1,&C9,&F5,&3E,&00,&32,&00
,&A6,&F1,&FE,&03,&28, 2042
19 DATA &03,&38,&17,&D0,&DD,&6E,&04,&DD,&66,&02,&CD
,&75,&BB,&DD,&7E,&00, 1806
20 DATA &32,&00,&A6,&FE,&08,&28,&03,&CD,&B4,&BB,&CD
,&CB,&C0,&DD,&21,&01, 1948
21 DATA &A6,&18,&5C,&FE,&03,&28,&03,&38,&10,&D0,&DD
,&6E,&04,&DD,&66,&02, 1522
22 DATA &CD,&75,&BB,&DD,&7E,&00,&CD,&B4,&BB,&CD,&CB
,&C0,&DD,&21,&01,&A6, 2449
23 DATA &CD,&01,&C2,&CD,&D6,&C1,&DD,&7E,&06,&CD,&44
,&C2,&CD,&D6,&C1,&DD, 2665
24 DATA &7E,&08,&CD,&B5,&C2,&DD,&7E,&07,&CD,&B5,&C2
,&CD,&E6,&C1,&DD,&7E, 2623
25 DATA &0A,&CD,&B5,&C2,&DD,&7E,&09,&CD,&B5,&C2,&CD
,&E6,&C1,&CD,&20,&C2, 2585
26 DATA &DD,&7E,&0C,&CD,&B5,&C2,&DD,&7E,&0B,&CD,&B5
,&C2,&CD,&D6,&C1,&CD, 2694
27 DATA &EC,&C1,&DD,&7E,&05,&CD,&2B,&C2,&CD,&B5,&C2
,&DD,&7E,&04,&CD,&B5, 2540
28 DATA &C2,&CD,&E6,&C1,&DD,&7E,&03,&CD,&B5,&C2,&DD
,&7E,&02,&CD,&B5,&C2, 2681
29 DATA &CD,&E6,&C1,&DD,&7E,&01,&CD,&B5,&C2,&DD,&7E
,&00,&CD,&B5,&C2,&C9, 2684
30 DATA &3E,&20,&CD,&B5,&C2,&C9,&3E,&20,&CD,&B5,&C2
,&3E,&3A,&CD,&B5,&C2, 2249
31 DATA &3E,&20,&CD,&B5,&C2,&C9,&3E,&3A,&CD,&B5,&C2
,&C9,&3E,&55,&CD,&B5, 2309
32 DATA &C2,&3E,&48,&CD,&B5,&C2,&3E,&52,&CD,&B5,&C2
,&3E,&20,&CD,&B5,&C2, 2306
33 DATA &C9,&3E,&44,&CD,&B5,&C2,&3E,&41,&CD,&B5,&C2
,&3E,&54,&CD,&B5,&C2, 2344
34 DATA &3E,&55,&CD,&B5,&C2,&3E,&4D,&CD,&B5,&C2,&3E
,&20,&CD,&B5,&C2,&C9, 2321
35 DATA &3E,&31,&CD,&B5,&C2,&3E,&39,&CD,&B5,&C2,&C

```

```

9,&FE,&38,&28,&09,&FE, 2204
36 DATA &39,&28,&09,&FE,&3A,&28,&09,&C9,&3E,&30,&18
,&FB,&3E,&31,&18,&F7, 1435
37 DATA &3E,&32,&18,&F3,&FE,&30,&28,&19,&FE,&31,&28
,&21,&FE,&32,&28,&29, 1507
38 DATA &FE,&33,&28,&31,&FE,&34,&28,&39,&FE,&35,&28
,&41,&FE,&36,&28,&49, 1630
39 DATA &C9,&3E,&53,&CD,&B5,&C2,&3E,&4F,&CD,&B5,&C2
,&18,&F3,&3E,&4D,&CD, 2258
40 DATA &B5,&C2,&3E,&4F,&CD,&B5,&C2,&18,&E7,&3E,&44
,&CD,&B5,&C2,&3E,&49, 2196
41 DATA &CD,&B5,&C2,&18,&DB,&3E,&4D,&CD,&B5,&C2,&3E
,&49,&CD,&B5,&C2,&18, 2281
42 DATA &CF,&3E,&44,&CD,&B5,&C2,&3E,&4F,&CD,&B5,&C2
,&18,&C3,&3E,&46,&CD, 2194
43 DATA &B5,&C2,&3E,&52,&CD,&B5,&C2,&18,&B7,&3E,&53
,&CD,&B5,&C2,&3E,&41, 2158
44 DATA &CD,&B5,&C2,&18,&AB,&F5,&3A,&00,&A6,&FE,&08
,&28,&05,&F1,&CD,&5A, 2087
45 DATA &BB,&C9,&F1,&CD,&2B,&BD,&C9,&00, 1267
46 dat=0:sz=0:dz= 1
47 FOR adr = &4000 TO &42C7
48 READ byte : dat=dat+1
49   sz=sz+byte
50   POKE adr,byte
51   IF dat < 16 AND adr < &42C7 THEN 55
52   READ chksum
53   IF chksum<>sz THEN PRINT "Fehler in zeile :";d
z
54   dz=dz + 1 : sz=0:dat=0
55 NEXT adr
56 ' das File "UHRC000x.obj", ist fuer die Pseudoro
mvesion angepasst
57 ' der Speicherbereich ab &4000 dient nur als Zw
ischenbuffer ,und
58 ' Programm lauft an dieser Adresse nicht.Es ist
nur in einem Rom
59 ' oder Pseudoram lauffaehig ab adresse &c000.en
d

```

*Listing 2:  
Rommon.bas: Rommonitor mit ROM-Verwaltung*

```

10 MEMORY &3FFF:LOAD"vwsrom.obj",&9800:CAL [3316]
L &9800
11 !SCREENDOWN:MODE 2:PRINT" *** Pseudoro [5215]
m-Monitor ***"
20 LOCATE 20,10:PRINT"L E S E N [2778]
- 1 - "
30 LOCATE 20,12:PRINT"S C H R E I B E N [1875]
- 2 - "
35 LOCATE 20,14:PRINT"T E S T E N [1955]
- 3 - "
40 a$=INKEY$:IF a$="" THEN GOTO 40: [1743]
50 IF a$="1" THEN GOTO 1000 [1316]
60 IF a$="2" THEN GOSUB 2000 [988]
61 IF a$="3" THEN GOSUB 3000 [1651]
70 GOTO 20 [398]
1000 REM ***** [2031]
*****
1010 REM lesen Pseudoram [1674]
*
1020 REM ***** [2031]
*****
1030 ' [117]
1031 MODE 2:WINDOW #1,1,80,1,6:WINDOW #0,1 [6142]
,80,11,25:WINDOW#2,1,80,7,10
1040 CLS:PRINT " Programmteil Lesen " [3018]
1050 PRINT:PRINT: [757]
1060 LOCATE 10,10:INPUT "Startadresse HEX [10139]
ohne (& ) ";start$:LOCATE 50,10:PRINT"
":IF start$="" THEN LOCATE 50,10:
PRINT CHR$(24)"F E H L E R "CHR$(24):GOTO
1060

```

```

1070 LOCATE 10,12:INPUT"Endadresse  HEX o [6650]
hne (&)" ;ende$:LOCATE 50,12:PRINT"
":IF ende$="" THEN LOCATE 50,12:PRINT
CHR$(24)"F E H L E R "CHR$(24):GOTO 1070
1075 LOCATE 10,14:INPUT"Rom leseadresse & [8952]
xx ";rom$:LOCATE 50,14:PRINT"
:IF rom$="" THEN LOCATE 50,14:PRINT CHR$(2
4)"F E H L E R "CHR$(24):GOTO 1075
1080 CLS:CLS#1:CLS#2:start=VAL("&"+start$) [6144]
:ende=VAL("&"+ende$):rom=VAL("&"+rom$):adr
=start:an$=RIGHT$((start$),2):an=VAL("&"+a
n$)
1082 PRINT #2," ";:FOR s=an TO an+15: [2222]
PRINT#2,CHR$(24);HEX$((s),2);" ";CHR$(24)
;:NEXT
1090 ' Anfang Leseschleife [1117]
1100 CLS:PRINT#1, "lesen Speicher ";"# "; [3286]
HEX$(rom)
1110 PRINT HEX$(adr);" ";:FOR x=1 TO 16: [5488]
PEEKLIST,rom,adr:PRINT " ";:adr=adr+1:NEXT
:PRINT
1120 IF adr>ende THEN PRINT:PRINT "lesen b [5414]
eendet Taste druecken !";:CALL &BB18:GOTO
11
1130 GOTO 1110 [349]
2000 REM ***** [1992]
*****
2010 REM schreiben Pseudora [1122]
m *
2020 REM ***** [1992]
*****
2021 MODE 2:WINDOW #1,1,80,1,6:WINDOW #0,1 [6142]
,80,11,25:WINDOW#2,1,80,7,10
2030 CLS:PRINT" Programmteil Schreiben " [3199]
2040 PRINT:PRINT [743]
2050 LOCATE 10,10:INPUT "Startadresse HEX [10376]
ohne (&)" ;start$:LOCATE 50,10:PRINT"
":IF start$="" THEN LOCATE 50,10:
PRINT CHR$(24)"F E H L E R "CHR$(24):GOTO
2050
2060 LOCATE 10,12:INPUT"Ram schreibadresse [9415]
&xx ";rom$:LOCATE 50,14:PRINT"
":IF rom$="" THEN LOCATE 50,14:PRINT CHR
$(24)"F E H L E R "CHR$(24):GOTO 2060
2070 CLS:CLS#1:CLS#2:start=VAL("&"+start$) [4732]
:rom=VAL("&"+rom$):adr=start
2071 PRINT #2," ";:FOR s=an TO an +15 [2222]
:PRINT#2,CHR$(24);HEX$((s),2);" ";CHR$(24)
);:NEXT
2080 PRINT CHR$(24);HEX$(adr);CHR$(24);" [11743]
";:FOR x=1 TO 16:LOCATE#1,1,1:INPUT#1," EI
NGABE ",wert$:IF wert$="" THEN GOTO 11 ELS
E wert=VAL("&"+wert$):!POKE,rom,adr,wert:
PRINT HEX$((wert),2);" ";:adr=adr+1:NEXT
2081 PRINT " " [520]
2090 GOTO 2080 [347]
3000 '***** [2154]
*****
3001 'testen zweier Speicherbereiche in p [2772]
seudobereichen *
3002 '***** [2154]
*****
3010 MODE 2:WINDOW #1,1,80,1,6:WINDOW #0,1 [6142]
,80,11,25:WINDOW#2,1,80,7,10
3020 CLS:CLS#1:CLS#2:LOCATE#1,1,1:INPUT#1, [3410]
"eingabe Startadresse ohne & ";start$
3030 INPUT#1,"Orginial Romadresse ohne & [4921]
XX ";source$
3040 INPUT#1,"kopie Romadresse ohne & [2977]
XX ";destination$
3041 start=VAL("&"+start$):source=VAL("&"+ [5376]
source$):destination=VAL("&"+destination$)
3042 PRINT#1,:PRINT#1:PRINT#1, "Abruch mit [4435]
ESC und <ENTER>"
3050 ' vergleich anfang ***** [1250]
3060 LOCATE 1,10: [787]

```

```

3070 FOR x=start TO &FFFF:t$=INKEY$:IF T$= [13191]
CHR$(13) THEN GOTO 11 ELSE !WRSROM,source,
x,&8000,1:!WRSROM,destination,x,&8001,1:a=
PEEK(&8000):b=PEEK(&8001):LOCATE 1,1:PRINT
HEX$((x),4);" ";HEX$((a),2);" ";HEX$((b),
2):IF a<>b THEN GOTO 3500 ELSE NEXT
3500 CLS:CLS#1:LOCATE#1,1,1:PRINT#1,"F E H [5530]
L E R an A D R E S S E ":LOCATE 1,1:PR
INT HEX$((x),4);" ";HEX$((a),2);" ";HEX$((
b),2)
3600 CLS#2:LOCATE #2,1,1:PRINT#2, "w e i t [5641]
e r TASTE druecken ":CALL &BB18:start=x
+1:GOTO 3070

```

### Listing 3: Datalader von VWSROM.OBJ

```

5 MEMORY &97FF
10 DATA &00,&00,&00,&00,&01,&0D,&98,&21, 199
20 DATA &00,&98,&C3,&D1,&BC,&24,&98,&C3, 1127
30 DATA &54,&98,&C3,&82,&98,&C3,&88,&98, 1196
40 DATA &C3,&A4,&98,&C3,&B4,&98,&C3,&C8, 1433
50 DATA &98,&C3,&E1,&98,&43,&41,&4C,&CC, 1136
60 DATA &53,&43,&52,&45,&45,&4E,&55,&D0, 741
70 DATA &53,&43,&52,&45,&45,&4E,&44,&4F, 595
80 DATA &57,&CE,&53,&45,&4C,&45,&43,&D4, 869
90 DATA &57,&52,&53,&52,&4F,&CD,&50,&4F, 777
100 DATA &4B,&45,&CE,&50,&45,&45,&4B,&4C, 719
110 DATA &49,&53,&D4,&00,&FE,&06,&CO,&DD, 1041
120 DATA &66,&0B,&DD,&6E,&0A,&22,&7F,&98, 767
130 DATA &DD,&7E,&08,&32,&81,&98,&DD,&46, 977
140 DATA &05,&DD,&4E,&04,&DD,&56,&03,&DD, 839
150 DATA &5E,&02,&DD,&66,&01,&DD,&6E,&00, 751
160 DATA &DD,&7E,&06,&06,&DF,&7F,&98,&C9,&00, 1056
170 DATA &00,&00,&3E,&CO,&CD,&08,&BC,&C9, 856
180 DATA &3E,&40,&CD,&08,&BC,&C9,&DD,&66, 1051
190 DATA &05,&DD,&6E,&04,&DD,&56,&03,&DD, 871
200 DATA &5E,&02,&DD,&46,&01,&DD,&4E,&00, 687
210 DATA &DD,&7E,&06,&C9,&FE,&01,&CO,&DD, 1222
220 DATA &4E,&00,&2A,&7D,&AE,&11,&9B,&A6, 757
230 DATA &CD,&CE,&BC,&C9,&FE,&04,&CO,&DD, 1471
240 DATA &4E,&06,&CD,&OF,&B9,&C5,&CD,&8E, 1033
250 DATA &98,&ED,&B0,&C1,&CD,&18,&B9,&C9, 1373
260 DATA &FE,&03,&CO,&DD,&4E,&04,&CD,&OF, 972
270 DATA &B9,&C5,&DD,&7E,&00,&DD,&66,&03, 1055
280 DATA &DD,&6E,&02,&77,&C1,&CD,&18,&B9, 1059
290 DATA &C9,&FE,&02,&CO,&DD,&4E,&02,&CD, 1155
300 DATA &OF,&B9,&C5,&DD,&66,&01,&DD,&6E, 1052
310 DATA &00,&7E,&CD,&FF,&98,&3E,&20,&CD, 1037
320 DATA &5A,&BB,&C1,&CD,&18,&B9,&C9,&5F, 1180
330 DATA &OF,&OF,&OF,&OF,&E6,&OF,&CD,&10, 526
340 DATA &99,&7B,&E6,&OF,&CD,&10,&99,&C9, 1096
350 DATA &FE,&0A,&38,&02,&C6,&07,&C6,&30, 773
360 DATA &CD,&5A,&BB,&C9,&00, 683
370 dat=0:sz=0:dz= 10
380 FOR adr = &9800 TO &991C
390 READ byte : dat=dat+1
400 sz=sz+byte
410 POKE adr,byte
420 IF dat < 8 AND adr < &991C THEN 460
430 READ chksum
440 IF chksum<>sz THEN PRINT "Fehler in zeile :";
dz
450 dz=dz + 10 : sz=0:dat=0
460 NEXT adr
470 CALL &9800:NEW

```

2732, 2764 und 27128 gesteckt werden. Wer die kleinen 2 KB Eproms noch einsetzen möchte (sie kosten mittlerweile mehr als die großen Brüder 64 und 128), dem kann auch geholfen werden. Hier macht man an Pin 23 der Epromfassung eine Trennstelle und legt diesen Pin auf konstant 5 Volt, und schon ist auch ein 2716 lauffähig. Der Übersichtlichkeit wegen haben wir hier keinen Jumper spendiert. Wegen der einfachen Decodierung ist es natürlich nur möglich, ein Eprom einzusetzen. Das heißt, daß beim Typ 27128 der ganze 16 KB Bereich, beim 2764 nur die Hälfte und beim 2732 mit 4 KB nur ein Viertel des reservierten Bereiches benutzt werden kann. Wie Sie es bei der SCHNEIDERWARE schon immer vorfanden, können mehrere Karten parallel betrieben werden.

### Die SOFTWARE – klein aber fein

Jetzt haben Sie allerhand an Informationen über Speicher erhalten. Nun werden wir uns mit der benötigten Software befassen. Das Problem beim Auslesen eines Speichers, der sich in einem überlagerten Bereich befindet, ist beachtlich. Ich habe im Verwaltungsprogramm einige RSX-Befehle geschaffen, die diese Aufgabe zu unserer vollsten Zufriedenheit erfüllen. Wie Sie wissen, gehen alle Leseoperationen im oberen Adressbereich in das BASIC-ROM. Wollen wir nun ein Byte z.B. aus dem siebten (Floppy-ROM) lesen, müssen wir uns der Firmware bedienen. Während der Befehlsausführung muß mit der Routine

»KL L ROM SELECT« ein entsprechendes ROM selectiert werden. Mit der Routine »KL ROM DESELECT« wird der ursprüngliche Zustand wieder hergestellt. Möchte man nun ein Programm in einem externen ROM ausführen, so kann man diese Routine nicht einfach mit CALL ADRESSE anspringen, denn man würde unweigerlich ins Video-RAM gelangen. Da in diesem Speicherbereich für den Call-Befehl mit Sicherheit keine sinnvollen Werte stehen würden, befördert sich Ihr CPC ungeniert ins Jenseits. Hier können wir auf einen RSX-Befehl zugreifen, der uns erlaubt, jede beliebige Adresse im System anzuspringen. Eine Auflistung der vorhandenen RSX-Befehle zeigt Ihnen Tabelle 2. Sie möchten vielleicht auch eine Routine in Ihr neugebautes Pseudo-ROM schreiben. Hier möchte ich Ihnen nur kurz den Weg beschreiben, wie man das macht. Wenn Ihnen ein Assembler (z.B. GENA31) zur Verfügung steht, müssen Sie nur die ersten 5 Zeilen des Uhrentreibers &01,00,00,00 ersetzen, beginnend bei der Adresse &C000. Diese Bytes sind der Kennsatz oder auch HAEDER genannt. Das erste Byte &01 ermöglicht dem Kernel, das ROM als Hintergrund-ROM zu identifizieren. Sieben dieser Extension-ROMs kann der CPC verwalten. Die drei folgenden Bytes sind unbedeutend. Sie können z.B. die Versionsnummer oder einen Programmindex beinhalten, dürfen aber auch mit 0 belegt sein, ganz wie Sie wollen. Diese Änderung im Hintergrund-ROM ist eigentlich die einzige, die Sie durchführen müssen, um

eine vorhandene RAM-residente RSX-Erweiterung an ein Hintergrund-ROM anzupassen. Sollte das RSX-Programm Speicherzellen verwenden, deren Inhalt sich während der Programmausführung ändert, muß dieser Bereich in den Memorypool verlegt werden, denn das Kernel kann nur mit besonderen Vorkehrungen aus externen ROMs lesen. Um nun dem Betriebssystem eine andere Einschaltmeldung verpassen zu können, bedienen wir uns eines einfachen Tricks. Nach dem Einschalten des Computers springt das Betriebssystem in alle angeschlossenen externen ROMs und versucht sie zu initialisieren. Dazu bedient es sich des RSX-Befehls, der in der Befehlsliste an erster Stelle eingetragen ist. Dieser Befehl beinhaltet als Kennzeichen für eine ordnungsgemäße Initialisierung einen Code, der das CARRY-BIT setzt (SCF). Vor diesem Befehl greifen wir mit unserer neuen Titelseite ein. Da natürlich unser neues Programm Register des Prozessors verändert, müssen wir diese beim Einsprung in diesen Programmteil erst einmal retten. Danach springen wir mit einem Call Adresse in das Uhrenprogramm und restaurieren danach die Register wieder. Besitzen Sie einen Assembler, so können Sie den Text direkt eingeben.

```

690 INIT2:PUSH AF      ;REGISTER
691   PUSH BC          ;
692   PUSH DE          ;RETTEN
693   PUSH HL          ;
694   CALL PRDATI      ;UHR AUFRUFEN
695   POP HL           ;
696   POP DE           ;REGISTER
697   POP BC           ;
698   POP AF           ;RESTAURIEREN
699   SCF              ;CARRY SETZEN
700   RET              ;RUECKSPRUNG

```

### Brandheiße Knüllerpreise

Schneider		Epson	
CPC 6128 mit Grünmonitor	889,-	Drucker anschlussfertig an CPC 6128/	
CPC 6128 mit Farbmonitor	1499,-	PC / Atari ST und Comm.AMIGA:	
Joyce PCW 8256	1549,-	LX 86	749,-
Joyce Plus	2099,-	FX 800	1049,-
PC mit SW-mon. + 1 Laufwerk	1849,-	FX 1000	1319,-
+ 2 Laufwerken	2249,-	LQ 800	1499,-
PC mit SW-mon. + 1 Laufwerk		LQ 1000	1949,-
+ 10 MB-Festplatte	3149,-	LQ 2500	2629,-
dto. + 20 MB-Festplatte	3599,-	EX 800	1419,-
PC mit Farbmon. + 1 Laufwerk	2249,-	EX 1000	1879,-
+ 2 Laufwerken	2679,-	IX 800	1679,-
PC mit Farbmon. + 1 Laufwerk		SO 2500	3649,-
+ 10 MB-Festplatte	3599,-	Color für EX 800/1000	219,-
dto. + 20 MB-Festplatte	4049,-		
Akustikkoppler Dataphon S21d	229,-		
Akustikk. Dataphon S 21 / 23	339,-		
		Star	
		NL-10 anschlussfertig	829,-
		NG-10 anschlussfertig	899,-
		Commodore	
		AMIGA 2000 mit Farbmonitor 1081	3599,-
		Commodore PC-10 II	2499,-
		Commodore C 128 D	1169,-
		Commodore SX-64	1479,-
Disketten			
3" Zoll für CPC 6128 1 St.	12,-		
10 St.	95,-		
5 1/4" 3M 744 D-O SSD 10 St.	33,-		
100 St.	289,-		

Versandkostenpauschale (Warenwert bis DM 1000,-) darüber: Vorauskassa (DM 8,-/20,-), Nachnahme (DM 11,20/23,20), Ausland (DM 18,-/30,-). Lieferung nur gegen NN oder Vorauskassa. Ausland nur Vorauskassa. Preisliste (Computertyp angegeben) gegen Zusendung eines Preisumschlages.

**CSV RIEGERT**

Schloßhofstr. 5, 7324 Rechberghausen, Tel. (0 71 61) 5 28 89

**G Computerstore**

Hochstraße 11  
8500 Nürnberg 80  
Tel.: 09 11 / 28 90 28

Alle Original-SCHNEIDER-Produkte zu günstigen Preisen lieferbar.

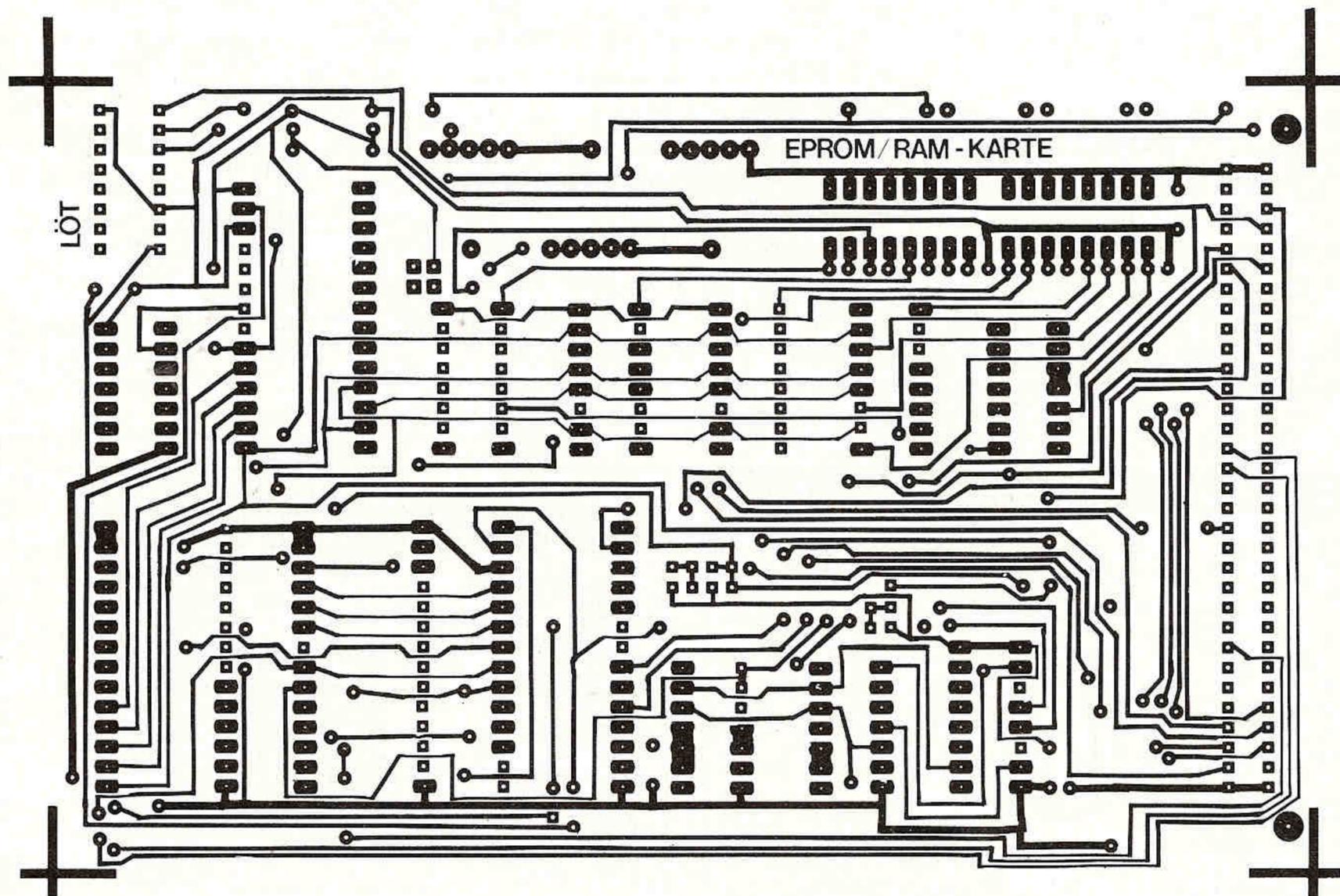
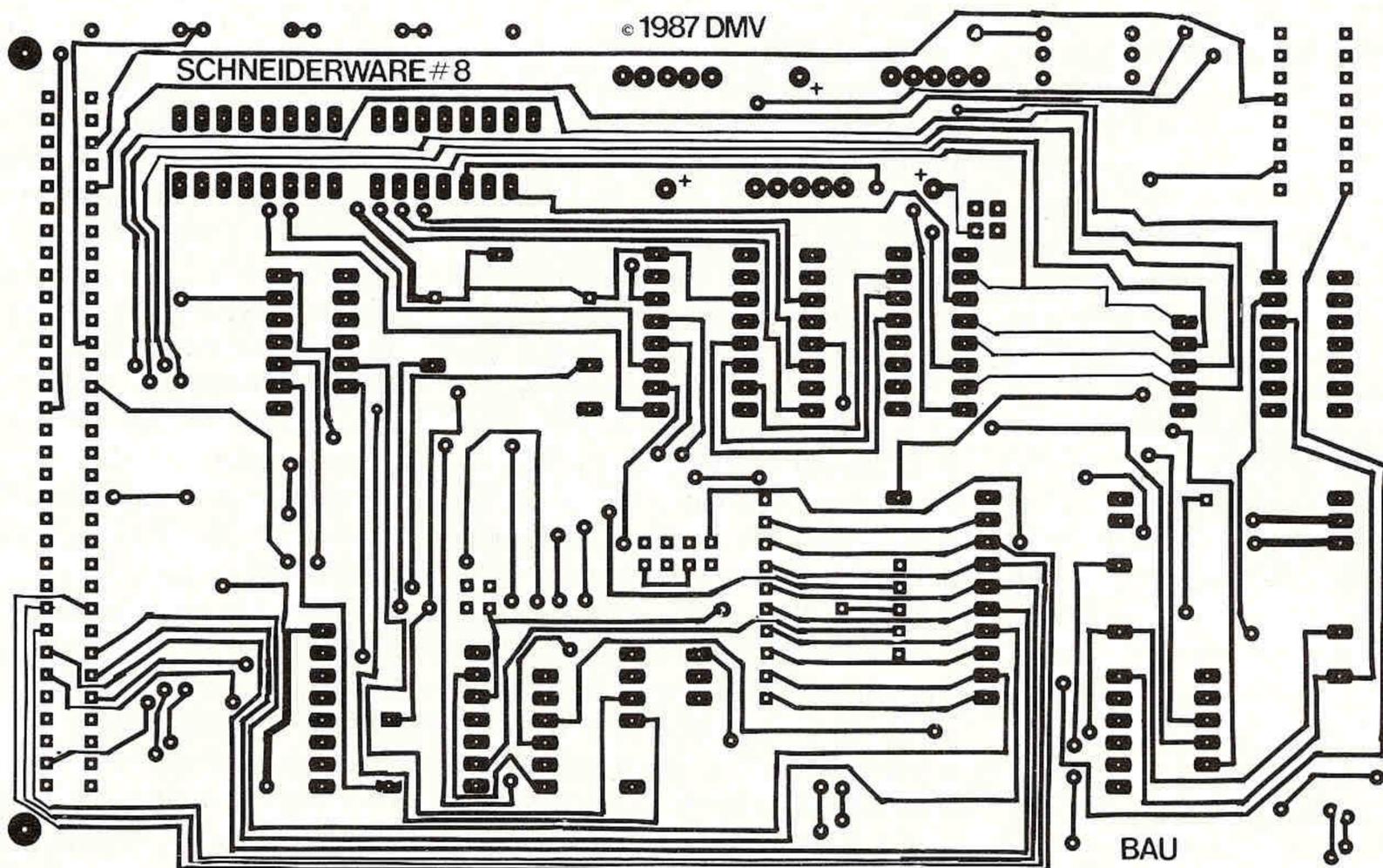
PC-ZUBEHÖR:	
PC-Erweiterung auf 640k	59,-
DYSAN-Filecard 10MByte	898,-
inkl. Software XTREE	
VORTEX-Filecard 20MByte	1398,-
Typendrucker SD-15i	748,-
STAR-WRITER PC	298,-
XTREE-das Progr. für die Verwaltung Ihrer Harddisc	135,-
JOYCE-ZUBEHÖR:	
JOYCE-Nachrüstsatz	698,-
JOYCE-RAM-Erweiterung	80,-
Typendrucker SD 15	698,-
JOYCE-Zweitlaufwerk FD-2	628,-
MOUSE-PACK	248,-
LOCOMAIL	128,-
SUPERCOPY JOYCE	85,-
TASWORD 8000deutsch	148,-



CPC/ JOYCE-ZUBEHÖR:	
3"Markendisketten 10 St.	75,-
NEVADA-COBOL	109,-
NEVADA-FORTRAN	109,-
SUPERCOPY CPC	79,-
TURBO-PASCAL	199,-
DR DRAW	199,-
CPC-ZUBEHÖR:	
CPC 464/Grün + DDI-1	798,-
Laufwerk DDI-1	398,-
Laufwerk FD-1 mit Kabel	398,-
CUMANA-Zweitlaufwerk	369,-
dk-tronics 64K-Erweiter.	129,-
Aufrüstung VORTEX 256K	98,-
Aufrüstung VORTEX um 256K	75,-
SUPERCOPY CPC	79,-
TURBO-PASCAL/Grafik (D)	259,-
DRUCKER:	
NEC-P6 (24 Nadeln)	1298,-
STAR-NL 10	748,-

Wir führen zu den Original-SCHNEIDER-Produkten Artikel verschiedener Firmen wie DATA BECKER, VORTEX, STAR-DIVISION, CUMANA, GERDES, PROFISOFT, ARIOLA, RUSH-WARE, MARKT & TECHNIK, SYBEX, STAR, SCHNEIDER-DATA usw.!

Platinenlayout Bauseite



Platinenlayout Lötseite

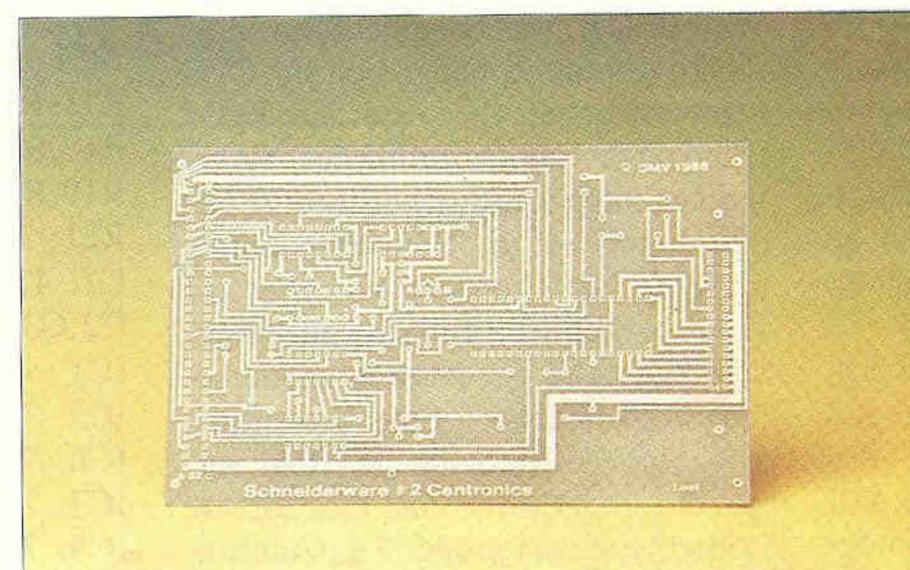
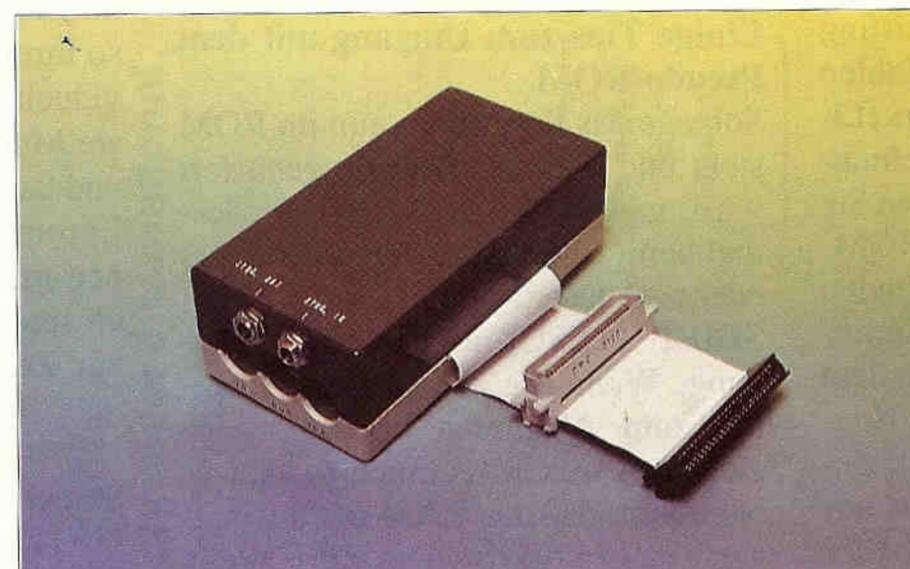
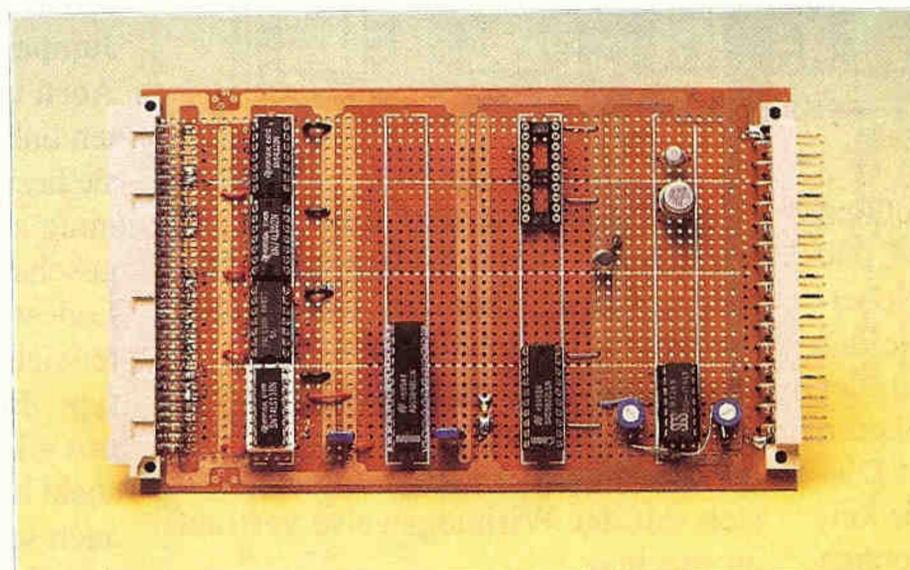
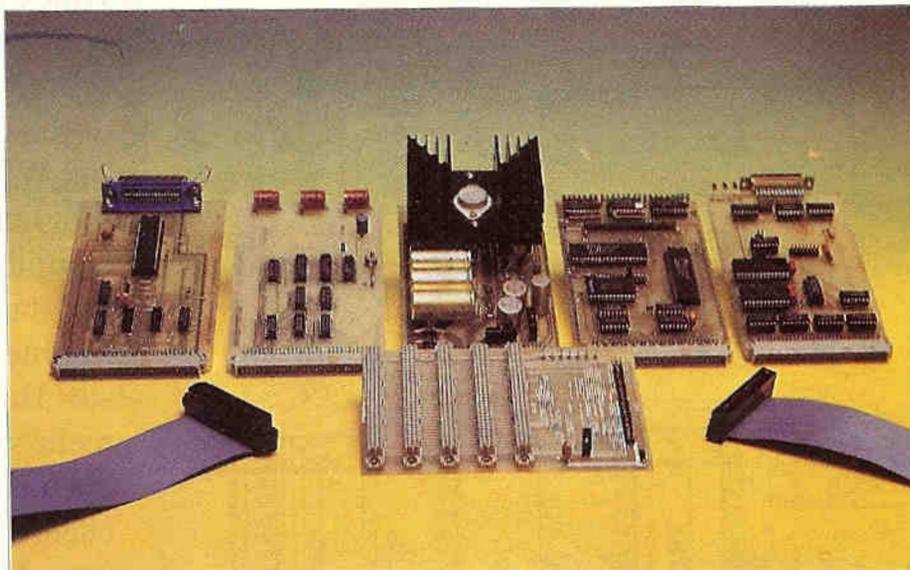
## Für Ihren CPC

Die CPC-Schneiderware ist ein universelles Peripheriesystem für die Schneider CPC's auf der Basis des bekannten ECB-Bussystems. Um die Schneiderware an Ihren CPC anzuschließen, benötigen Sie:

1. Das Verbindungskabel vom Expansionsport des Rechners zur Basisplatine (Rechnertyp beachten, da Anschlüsse bei 464/664 verschieden von 6128)
2. Die Basisplatine, welche die Pinbelegung der CPC-Ports auf die des ECB-Systems umsetzt. Diese Karte enthält fünf Steckplätze zur Aufnahme und gleichzeitigen Ansteuerung der Schneiderware- Erweiterungskarten.

Wollen Sie nur eine Karte betreiben, so können Sie diese über ein selbstgefertigtes Kabel an den CPC anschließen. Die Anschlußbelegung dieses Kabels sehen Sie in Heft 7/86, S.61.

Das verwendete Platinenmaterial ist glasfaserverstärktes Epoxydharz; die beidseitig beschichteten Platinen sind chemisch durchkontaktiert. Für die Fertigbausteine kommen Bauteile erster Wahl zum Einsatz.



## Gesammelte Werke

Die SCHNEIDERWARE begann in Heft 6/86. Über den Platinenservice stehen Ihnen alle Karten zur Verfügung.

### Die Preise:

Basisplatine, unbestückt	24,90 DM
dto., bestückt	62,90 DM
Kabel 464/664	35,90 DM
Kabel 6128	45,90 DM
Centronics, unbestückt	17,90 DM
dto., bestückt	79,90 DM
V/24, unbestückt	29,80 DM
dto., bestückt	139,90 DM
Netzteil, unbestückt	17,90 DM
dto, bestückt	119,90 DM
Trafo	79,90 DM
Karte und Trafo	184,90 DM
Hardware-Uhr, unbest.	29,80 DM
Hardware-Uhr, bestck.	99,90 DM
PIO-Karte, Platine unbest.	29,80 DM
PIO-Karte, Karte bestck.	198,90 DM
MIDI-Interf., Plat. unbest.	39,90 DM
MIDI-Interface kompl. best.	198,00 DM

## EPROM-RAM-Karte

Diese Karte ist eine Erweiterung, die es Ihnen ermöglicht, eigene oder fremde Programme beim Einschalten des Rechners oder nach Aufruf direkt aus EPROM oder akkugepuffertem RAM einzuladen. Diese Kombination hat den Vorteil, daß selbstgeschriebene Programme zunächst im RAM getestet werden können, bevor sie in das EPROM gebrannt werden. Heft 4/87 enthält Bauanleitung und Treibersoftware dieser Karte.

### Die Preise:

Platine, unbestückt	29,80 DM
Karte, funktionsfertig	229,90 DM

## A/D - D/A - Wandler

Mit dieser Karte können Sie analoge Werte (Temperatur, Spannung, Helligkeit usw.) über acht A/D- Kanäle in digitalisierter Form in den Rechner eingeben und digitale Werte, die der Rechner bereitstellt, in analoge Signale wandeln. Die Karte ist mit einem A/D- und zwei D/A-Bausteinen bestückt; einer der D/A-Wandler ist nicht beschaltet und steht zur freien Verwendung zur Verfügung. Ein Rasterfeld ermöglicht die einfache Erstellung von Testschaltungen. Die Bauanleitung zu dieser Karte ist in Heft 3/87 veröffentlicht.

### Die Preise:

Platine, unbestückt	29,80 DM
Karte, funktionsfertig	169,90 DM

## Zahlungsbedingungen:

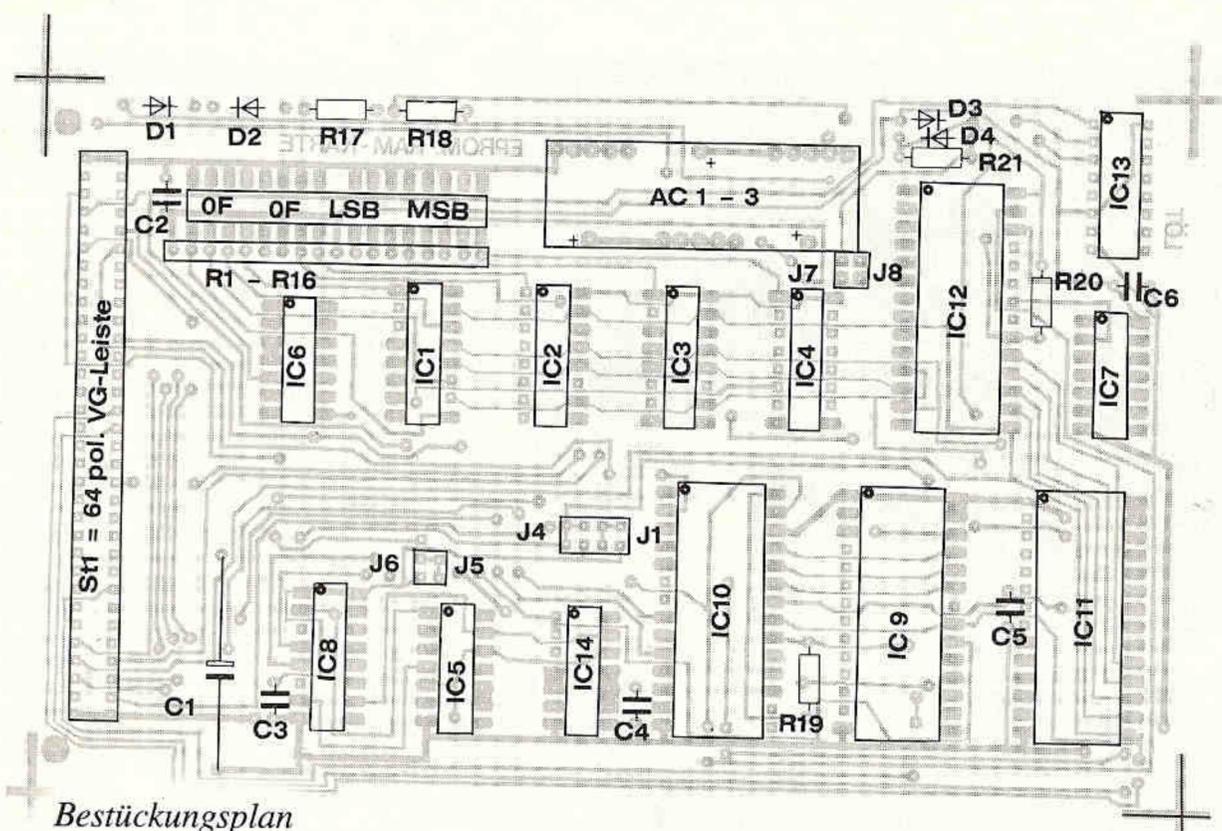
Gesamtpreis zuzüglich 5,— DM Porto/Verpackung (im Ausland 8,— DM Porto/Verpackung).

Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzügl. der Nachnahmegebühr (in das Ausland nicht möglich).

## Bitte Postkarte im Heft benutzen!

## Platine, unbestückt

SCHNEIDERWARE ist in drei Versionen für Sie verfügbar. Sie können nach Bauplan selbst bauen, die fertig bestückten und geprüften Karten über den Platinenservice erhalten oder die unbestückte Platine erwerben. Diese werden in Industriequalität gefertigt, sind verzinkt und gebohrt; doppelseitig beschichtete Platinen sind chemisch durchkontaktiert und geprüft. Hierbei haben Sie den Vorteil, die Platine nicht selbst herstellen zu müssen, jedoch die Bestückungskosten zu sparen und die Bauteile selbst einzukaufen.



Bestückungsplan

Nun assemblieren Sie den Quelltext mit `ORG #C000` und `ENT #C000` mit der `OPTION 16` in einen Speicherbereich hinter die Symboltabelle des Devpac (das macht der Assembler selbständig). Nach dem Assemblieren speichern Sie den Quellcode auf Diskette. Diejenigen unter Ihnen, die keinen Assembler ihr Eigen nennen können, können sich die Werte aus dem abgedruckten Hexdump (Listing 1) entnehmen. Sie müssen die Zahlen aber mit Hilfe des ROM-Monitors (Listing 2) eintippen. In der Menüfunktion 2 des ROMMONitors können Sie nach Angabe der gewünschten ROM-Selectadresse und der entsprechenden Anfangsadresse durch bloßes Eintippen der Hexzahlen Ihr Programm dem Pseudo-ROM übergeben. Das Hexdump enthält die lauffähige und angepasste Version des Uhrentreibers aus Heft 10/86 mit der geänderten Einschaltmeldung. Liegt Ihr Programm als Binärfile auf Diskette vor, so laden Sie das File an die Adresse &4000. Mit dem Verwaltungsprogramm für das Pseudo-ROM (VWS-ROM.OBJ) ist es nun eine Leichtigkeit, das Maschinencodeprogramm an die entsprechende Stelle ins Pseudo-ROM zu schreiben. Der Befehl `:WRSROM,&0F,&4000,&C000,&3fff` überträgt einen Bereich von 16 KB (&3fff bytes) an die Adresse des ROM's, das die Schreibadresse &0f besitzt. Die Startadresse der Datenquelle ist &4000: Demnach lautet die SYNTAX für diesen Befehl `:WRSROM,ROMSEL.,QUELLE,ZIEL,ANZAHL`

Dieser Befehl ist sehr universell zu verwenden. Durch Ändern der ROM-Selectadresse und der Datenquelle kann man den Inhalt eines ROMs in einen bestimmten Speicherbereich überspielen. Am besten, Sie experimentieren mit diesen Befehlen, um sich mit der Wirkungsweise vertraut zu machen.

### Einige Tips zum Umgang mit dem Pseudo-ROM

Solange das Programm nun im ROM steht und über die Batterie gebuffert wird, meldet sich dieses ROM jedesmal beim Einschalten ihres Rechners oder nach jedem Warmstart (`CTRL + SHIFT + ESC`) mit der neuen Meldung. Wenn Sie jedoch Ihren Rechner zum Abstürzen bringen, ganz gleich aus welchem Grund, so kann es passieren, daß das RAM mit unsinnigen Zeichen beschrieben wird. Wenn auch nur das erste Byte einen anderen Wert besitzt, was bei einem Eprom ja nicht vorkommt, hängt sich Ihr CPC beim Abfragen der externen ROM während der Initialisierung auf. Sie erwecken Ihren Rechner wieder zum Leben, wenn Sie die Batteriespannung unterbrechen (Jumper), was aber zur Folge hat, daß Ihr mühsam eingetipptes Programm auf Nimmerwiedersehen verschwindet. Eine wesentlich effektivere Methode ist, dem entsprechenden Eprom eine Leseadresse (mittels Jumper) größer 7 aufzubrummen. Adressen von 08 bis 0f sind möglich. Nach dem Einschalten können Sie mittels ROM-Monitor das veränderte RAM reparieren. Wenn Sie

Glück haben, ist nur das erste Byte zu Null geworden. Ein irrtümliches Beschreiben des RAM zu verhindern ist durch das Schließen des Jumpers, der mit WP (Write Protect) bezeichnet ist, möglich. Diese Brücke setzt den Schreibfreigabeeingang der RAMs auf konstant 5 Volt. Die Herstellung der Platine dürfte für Sie als ausgefuchste Hardwareprofis kein Hindernis mehr sein. Aber lassen Sie mich doch noch einige Tips zum Aufbau nachreichen. Die ICs gehören in gute Fassungen. Im unteren Teil der Schaltung sehen Sie zwei Dioden neben dem Jumper 7, der die Spannung für den Accu unterbricht. Diese Dioden sollten unbedingt Germaniumdioden sein; sie besitzen eine kleinere Schwellspannung als Siliziumtypen. Der parallelgeschaltete Widerstand begrenzt den Ladestrom des Accus; der Wert richtet sich nach dem eingesetzten Accutyp. Bei einem Accu 3,6 V=100 mA=h hat sich ein 120 Ohm Widerstand bestens bewährt. Sie können sich auch selbst beliebige Einschaltmeldungen (Liebeserklärung an Ihre Freundin) ins Pseudo-ROM schreiben, um so Ihrem CPC eine eigene, ganz individuelle Note zu verleihen. Das nächste Mal werden wir die entwickelten und lauffähigen Programme fest in ein Eprom brennen. Wir stellen Ihnen einen universellen Eprommer vor (inclusive Software natürlich), der alle zur Zeit gängigen Typen brennt.

(P.Richter)

Stückliste der Schaltung:

- D1,D2: 1N4148
- D3,D4: AA 109 oder ähnliche GERMANIUMdiode
- AC1-3: Akkus 1,2 V ca. 100 mAh
- R1-R16: 4,7k
- R17,R18: 680
- R19,R20: 15k
- R21: 360 (siehe Text)
- C1: 470 uF
- C2,C3: 100 nF
- IC1-4: 74LS85
- IC5: 74LS00
- IC6: 74LS32
- IC7: 74LS02
- IC8: 74LS175
- IC9,IC11: EPROM 2764
- IC10,IC12: RAM 6264
- IC13: 74LS138
- IC14: 74LS00

# Knack die Nuß

## Laplace-Käfer

**Problemstellung:** Ein Käfer krabbelt auf den Kanten eines Würfelgitters. Für jede Kante benötigt er eine Zeiteinheit. Kommt der Käfer an eine Ecke, so wirft er einen Würfel, um seine weitere Richtung zu bestimmen: bei 1 geht er nach rechts, bei 2 nach oben, 3 vorwärts, 4 nach links, 5 nach unten und bei 6 nach hinten.

Kommt der Käfer genau zum Gegenpunkt des Ausgangspunktes am Würfel, so hat er sein Ziel erreicht. Jetzt stellt sich die Frage, wieviele Zeiteinheiten der Käfer im Durchschnitt braucht.

**Lösung:** Möchte man das Problem auf mathematischem Weg mit Hilfe der Stochastik lösen, so wäre das mit einem sehr hohen Rechenaufwand verbunden. Viel eleganter kann das Problem durch den Computer gelöst werden. Der Computer simuliert dabei den Käfer.

Bevor die Simulation beginnen kann, wird die Auflösung des Würfels bestimmt, d. h. bei einem normalen Würfel beträgt sie 1: keine Unterteilung der Seitenkanten, bei 2 sind die Seitenkanten halbiert, d. h. der Käfer hat jetzt einen größeren Aktionsradius usw.

Bewegt sich der Käfer, so kann er maximal nur unter sechs verschiedenen Richtungen wählen. Die Auswahl der Richtung erfolgt per Zufallszahlengenerator. Der Computer überprüft jetzt, ob die eingeschlagene Richtung überhaupt möglich ist (z. B. Eckpunkte). Wenn wird der Zug ausgeführt und der Zeiteinheitenzähler um eins erhöht.

Die Simulation ist beendet, sobald der Käfer sein Ziel erreicht. Um genauere Werte zu erhalten, sind oft viele Simulationen notwendig. Der sich aus den verschiedenen Simulationen ergebende Mittelwert nähert sich dann dem rechnerischen Ergebnis an. Bei einer Auflösung von 1 liegt der Durchschnitt bei 10.

für 464-664-6128

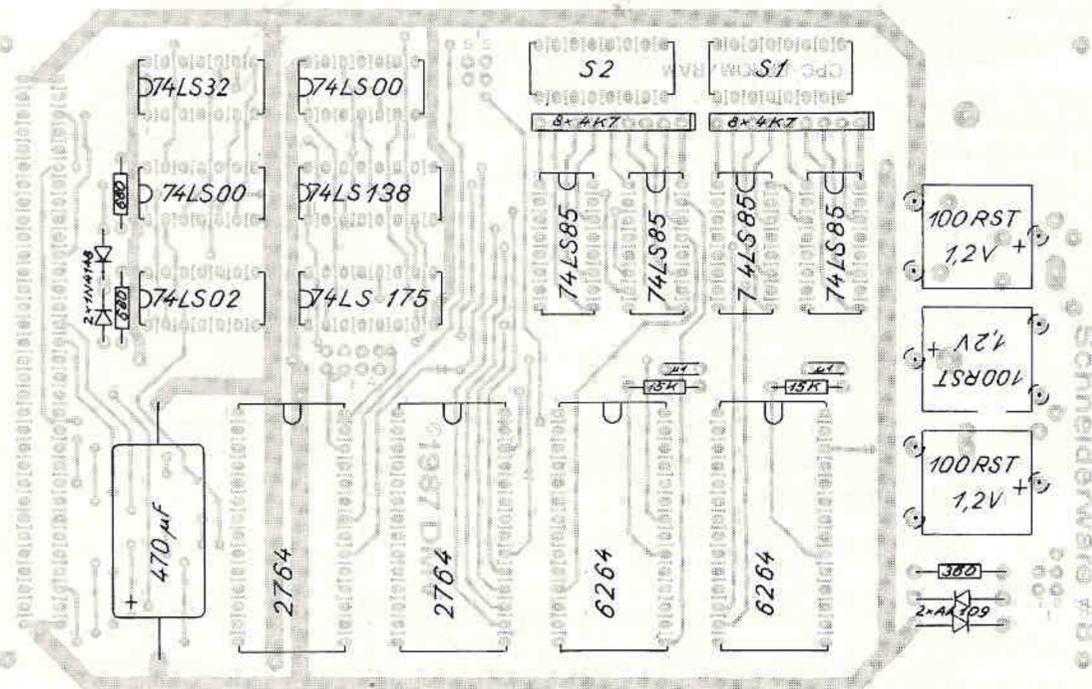


```

1 ' Rudi Weismeyer, Tel. 09642/ [1745]
2630
10 '----- Eingabe der Aufloesung des [5395]
Wuerfels -----
20 MODE 2:DEFINT a-z [2058]
30 WINDOW#0,1,80,7,25:WINDOW#1,12,80,4,6 [2373]
40 LOCATE 12,1:INPUT"Aufloesung des Wuerfe [3274]
ls ";a
50 IF a<1 THEN GOTO 40 ELSE u=190/a:v=120/ [2218]
a:w=55/a
60 LOCATE#1,2,1:PRINT#1,"Die Berechnung er [6247]
folgt mit ";a;"Unterteilungen"
70 '----- Schleife bis Kaefer am Endp [3474]
unkt -----
80 CLS#0:PLOT 120,20 [725]
90 WHILE NOT(x(1)=a AND x(2)=a AND x(3)=a) [2017]
100 i=INT(RND*6+1) [798]
110 IF i<4 AND x(i)<a THEN x(i)=x(i)+1:z=z [4002]
+1:GOTO 130 ELSE IF i<4 THEN 100
120 IF x(i-3)>0 THEN x(i-3)=x(i-3)-1:z=z+1 [3254]
ELSE 100
130 IF i=1 THEN DRAWR u,0,1:GOTO 190 [2170]
140 IF i=2 THEN DRAWR 0,u,1:GOTO 190 [2787]
150 IF i=3 THEN DRAWR v,w,1:GOTO 190 [2704]
160 IF i=4 THEN DRAWR-u,0,1:GOTO 190 [1807]
170 IF i=5 THEN DRAWR 0,-u,1:GOTO 190 [3556]
180 IF i=6 THEN DRAWR-v,-w,1 [802]
190 WEND [390]
200 '----- Berechnung des Durchschnitt [4475]
ts -----
210 zaehler=zaehler+1:summe=summe+z [4373]
220 LOCATE#1,16,3:PRINT#1,SPC(11) [1477]
230 LOCATE#1,2,3:PRINT#1,"Durchschnitt :"; [4727]
summe/zaehler
240 LOCATE#1,35,3:PRINT#1,"Zaehler :";zaeh [3318]
ler
250 z=0:FOR i=1 TO 6:x(i)=0:NEXT [1856]
260 IF INKEY$<>" THEN END ELSE GOTO 80 [2040]
    
```

## Layout der Epromkarte aus Heft 4/87

Bestückungsplan der Epromkarte



Hier finden Sie die Produktionslayouts der RAM/EPROM-Karte aus Heft 4/87.

**Hinweis:**

Die Verwendung verschiedener Akkus ist auf der Platine vorbereitet; der Bestückungsplan zeigt die auch in der Uhrenkarte verwendeten Typen Varta 100 RST (3 St.)

Weiterhin können folgende Akkus verwendet werden: (je 1 St.)

- MEMPAC F 60mAh
- MEMPAC S 100mAh
- SAFETRONIC 100mAh
- 3/170 DK 170 mAh
- 3/250 DK 250 mAh

# PC Schneider International

Postfach 250, 3440 Eschwege



## Platinenservice

### Für Ihren CPC

Die CPC-Schneiderware ist ein universelles Peripheriesystem für die Schneider CPC's auf der Basis des bekannten ECB-Bussystems. Um die Schneiderware an Ihren CPC anzuschließen, benötigen Sie:

1. Das Verbindungskabel vom Expansionsport des Rechners zur Basisplatine (Rechnertyp beachten, da Anschlüsse bei 464/664 verschieden von 6128)
2. Die Basisplatine, welche die Pinbelegung der CPC-Ports auf die des ECB-Systems umsetzt. Diese Karte enthält fünf Steckplätze zur Aufnahme und gleichzeitigen Ansteuerung der Schneiderware-Erweiterungskarten.

Wollen Sie nur eine Karte betreiben, so können Sie diese über ein selbstgefertigtes Kabel an den CPC anschließen. Die Anschlußbelegung dieses Kabels sehen Sie in Heft 7/86, S.61.

Das verwendete Platinenmaterial ist glasfaserverstärktes Epoxydharz; die beidseitig beschichteten Platinen sind chemisch durchkontaktiert. Für die Fertigungsbauweise kommen Bauteile erster Wahl zum Einsatz.

### Zahlungsbedingungen:

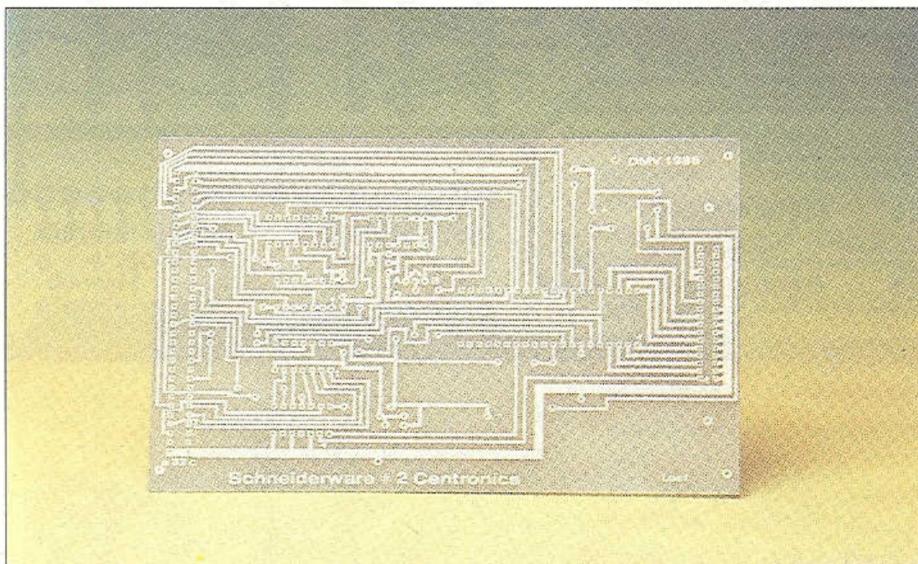
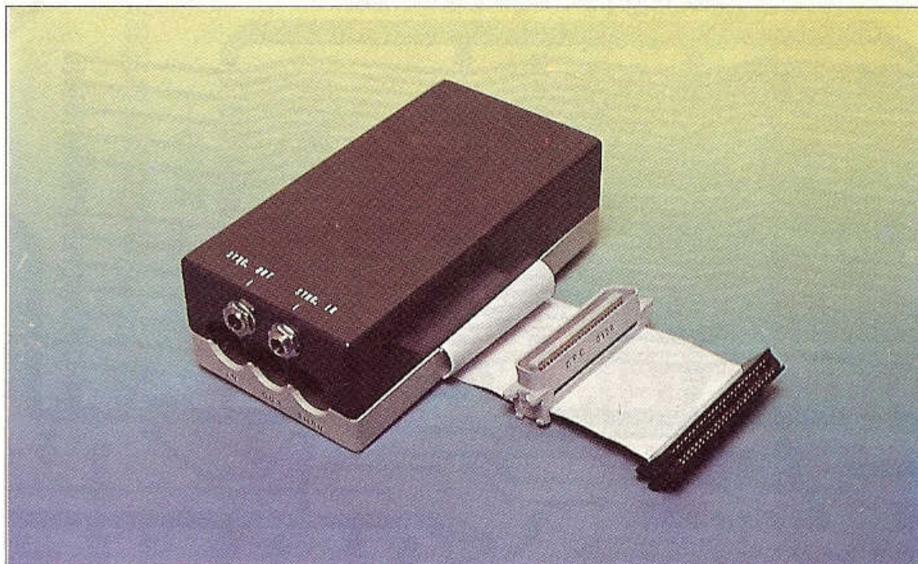
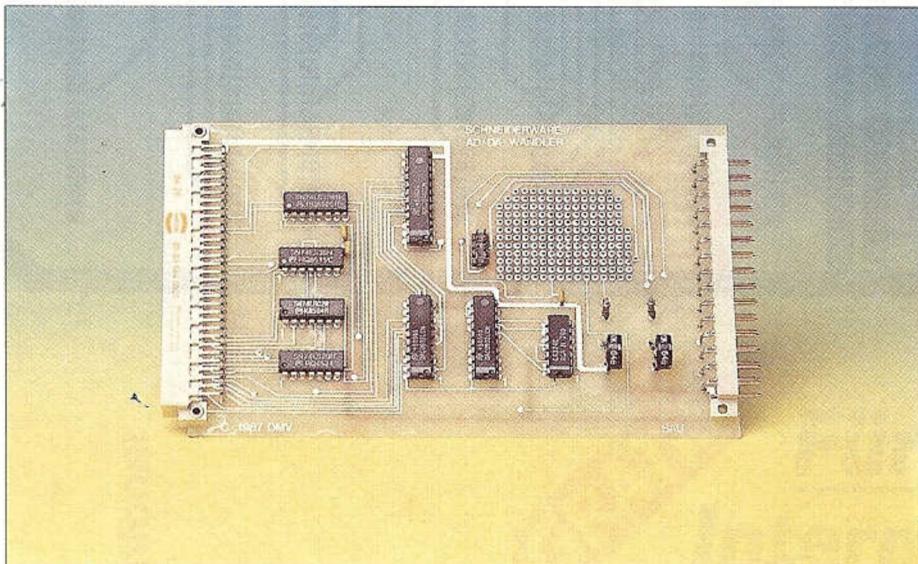
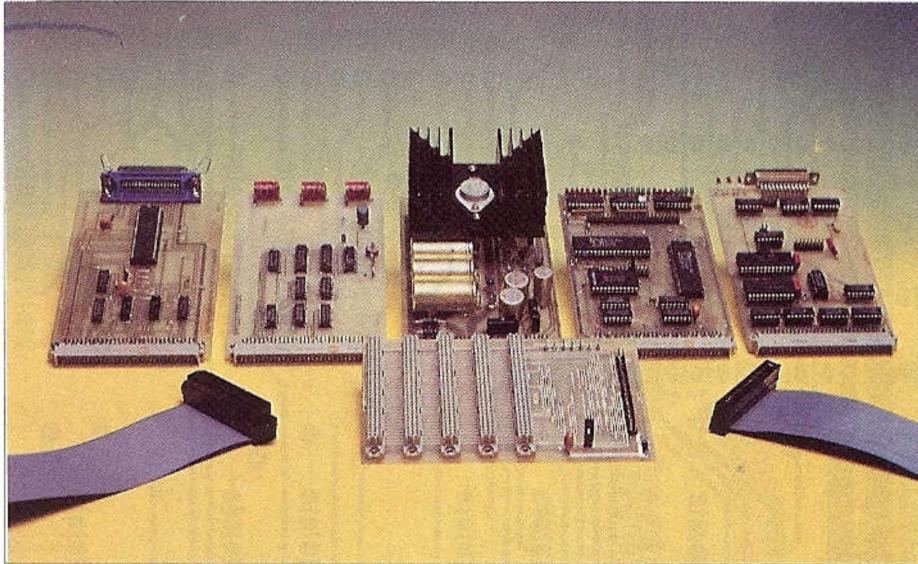
Gesamtpreis zuzüglich 5,- DM Porto/Verpackung (im Ausland 8,- DM Porto/Verpackung).

Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzügl. der Nachnahmegebühr (in das Ausland nicht möglich).

### Bitte Postkarte im Heft benutzen!

### Platine, unbestückt

SCHNEIDERWARE ist in drei Versionen für Sie verfügbar. Sie können nach Bauplan selbst bauen, die fertig bestückten und geprüften Karten über den Platinenservice erhalten oder die unbestückte Platine erwerben. Diese werden in Industriequalität gefertigt, sind verzinkt und gebohrt; doppelseitig beschichtete Platinen sind chemisch durchkontaktiert und geprüft. Hierbei haben Sie den Vorteil, die Platine nicht selbst herstellen zu müssen, jedoch die Bestückungskosten zu sparen und die Bauteile selbst einzukaufen.



### Gesammelte Werke

Die SCHNEIDERWARE begann in Heft 6/86. Über den Platinenservice stehen Ihnen alle Karten zur Verfügung.

#### Die Preise:

Basisplatine, unbestückt	24,90 DM
dto., bestückt	62,90 DM
Kabel 464/664	35,90 DM
Kabel 6128	45,90 DM
Centronics, unbestückt	17,90 DM
dto., bestückt	79,90 DM
V/24, unbestückt	29,80 DM
dto., bestückt	139,90 DM
Netzteil, unbestückt	17,90 DM
dto., bestückt	119,90 DM
Trafo	79,90 DM
Karte und Trafo	184,90 DM
Hardware-Uhr, unbest.	29,80 DM
Hardware-Uhr, bestck.	99,90 DM
PIO-Karte, Platine unbest.	29,80 DM
PIO-Karte, Karte bestck.	198,90 DM
MIDI-Interf., Plat. unbest.	39,90 DM
MIDI-Interface kompl. best.	198,00 DM

### EPROM-RAM-Karte

Diese Karte ist eine Erweiterung, die es Ihnen ermöglicht, eigene oder fremde Programme beim Einschalten des Rechners oder nach Aufruf direkt aus EPROM oder akkugepuffertem RAM einzuladen. Diese Kombination hat den Vorteil, daß selbstgeschriebene Programme zunächst im RAM getestet werden können, bevor sie in das EPROM gebrannt werden. Heft 4/87 enthält Bauanleitung und Treibersoftware dieser Karte.

#### Die Preise:

Platine, unbestückt	29,80 DM
Karte, funktionsfertig	229,90 DM

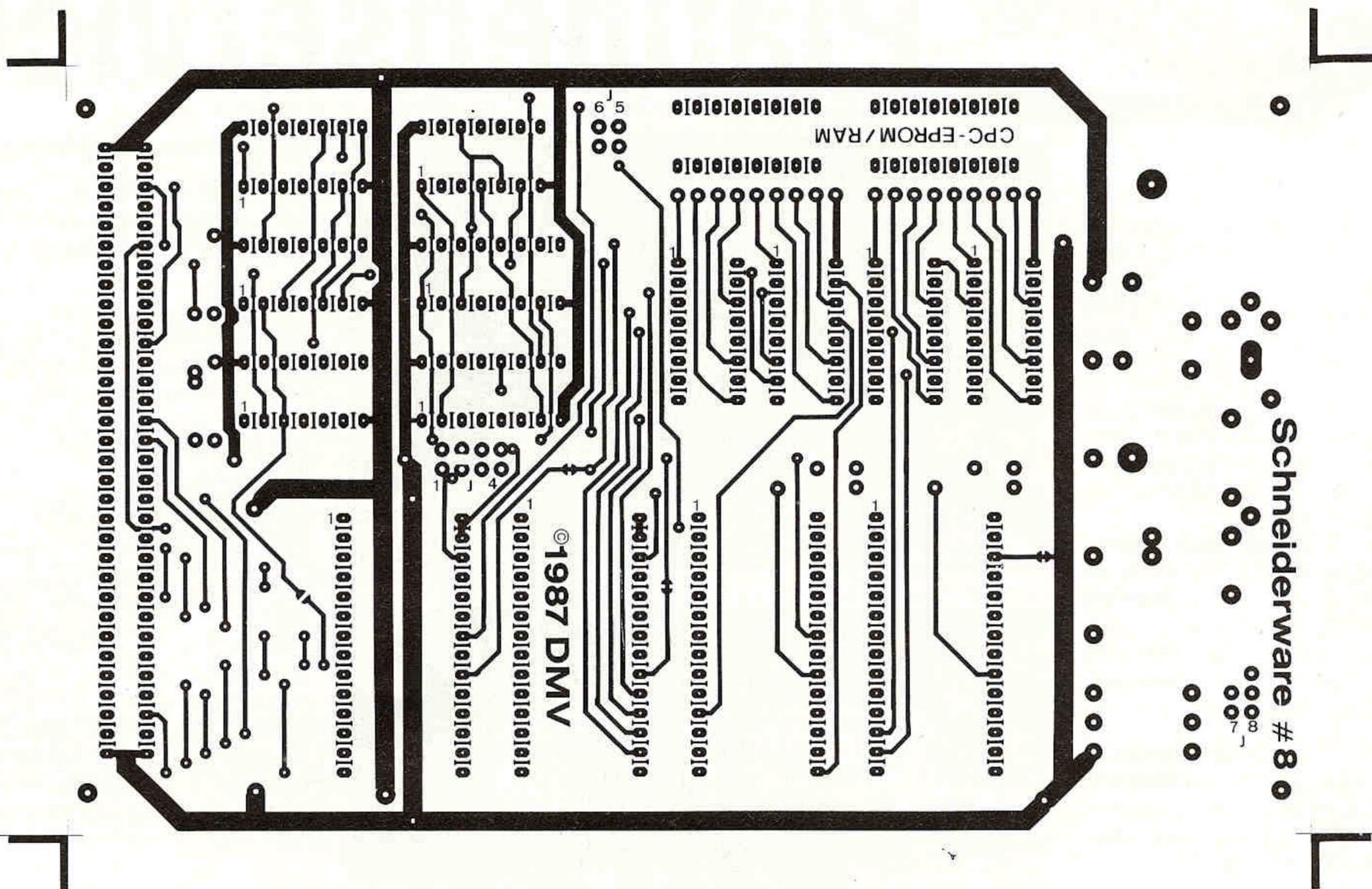
### A/D - D/A - Wandler

Mit dieser Karte können Sie analoge Werte (Temperatur, Spannung, Helligkeit usw.) über acht A/D-Kanäle in digitalisierter Form in den Rechner eingeben und digitale Werte, die der Rechner bereitstellt, in analoge Signale wandeln. Die Karte ist mit einem A/D- und zwei D/A-Bausteinen bestückt; einer der D/A-Wandler ist nicht beschaltet und steht zur freien Verwendung zur Verfügung. Ein Rasterfeld ermöglicht die einfache Erstellung von Testschaltungen. Die Bauanleitung zu dieser Karte ist in Heft 3/87 veröffentlicht.

#### Die Preise:

Platine, unbestückt	29,80 DM
Karte, funktionsfertig	169,90 DM

Layout der Epromkarte, Bauteilseite



Layout der Epromkarte, Lötseite

