loop 2

1. JAHRGANG

Zeitung für Computer-Bauer, -Anwender, -Programmierer und -Starter DM 3,–

Uwe u. Claudia Koch
Frankenstraße 25
5880 LÜDENSCHEID

In eigener Sache . . .

Nun liegt sie endlich vor Ihnen, die neue "loop". Wir haben aus der ersten Ausgabe und den vielen Zuschriften, für die wir uns herzlich bedanken, gelernt. Diese loop enthält fast ausschließlich die Informationen und Hinweise, auf die alle dringend gewartet haben.

Diese zweite loop ist die erste, die wir völlig unter eigener Regie erstellen - wir, das sind Rolf-Dieter Klein und Gerd Graf. Wir sind sicher, auf diese Weise mehr und sachbezogenere Information übermitteln zu können.

Diese zweite loop hat lange auf sich warten lassen - zu lange, haben manche Leser gemeint und schon daran gezweifelt, ob ihr Aboscheck überhaupt angekommen ist. Die Umstellung auf Eigenerstellung war mit daran schuld. Nun haben wir das Problem im Griff - loop 3 wird im Februar 1985 erscheinen.

"Wenn Sie so wollen - der NDR-Computer ist das Ding des Jahrhunderts!" - diese Aussage von einen erfahrenen, alten Elektronikhasen, Herrn Wolfgang Meier vom Fernmeldeamt 1 in Bremen steht für viele Meinungen der NDR-Computer Anwender. Zwanzigtausend werden es wohl dieses Jahr noch werden, und es werden immer mehr.

Wir freuen uns über solches Lob - wir freuen uns aber auch über fundierte Kritik, und wir versprechen, diese Kritik sehr ernst zu nehmen. Schon der Inhalt dieser loop-Ausgabe basiert auf solcher Kritik.

Ein steter Kritikpunkt war das Fehlen der kursbegleitenden Information. Der Franzis-Verlag hat uns hier mit seinem neuen Sonderheft "Mikrocomputer Schritt für Schritt" sehr viel geholfen. Im Januar/Februar 85 wird das zweite Sonderheft, das sich mit den weiteren Folgen der Sendereihe und schon mit dem Ausbau des NDR-Computers auf CP/M befasst, folgen.

1985 wird das Jahr der Dokumentation. Schon jetzt werden die Begleithefte kostenlos den Bausätzen beigelegt, und ab 1985 werden auch alle Programme mit ausführlicher Dokumentation und vielen Beispielen ausgeführt.

Auch wenn nun loop direkt von GES herausgegeben wird, soll loop kein Werbeblatt werden - das versprechen wir unseren Lesern und unseren Kunden. Prüfen Sie diese Ausgabe kritisch und abonieren Sie loop, falls Sie es nicht schon längst getan haben. Das geht nun viel einfacher - bei jeder Bestellung können Sie einfach ein loop-Abo mitbestellen.

Schreiben Sie uns, ob und wie Ihnen diese Ausgabe gefällt, welche Themen wir noch intensiver behandeln sollen, was wir für Sie tun können - wir richten uns danach!

Wir wünschen Ihnen allen ein frohes und erfolgreiches neues Jahr und viel Spaß und viel Erfolg mit dem NDR-Computer.

Rolf-Dieter Klein Gerd Graf

Christiani steigt mit ein!

Zur Sendereihe „Mikrocomputer – selbst gebaut und programmiert" gibt es eine ganze Reihe Schriften, Bücher und Bauanleitungen.
Dennoch äußerten viele Computer-Bauer und -Anwender den Wunsch nach zusätzlichem Lehrmaterial, um Themen der Sendereihe noch gründlicher bearbeiten zu können.
Diesem Wunsch ist das Lehrinstitut Christiani aus Konstanz nachgekommen, das nun im Frühjahr 1985 einen Lehrgang herausbringen wird, der auf die Sendereihe „Mikrocomputer – selbst gebaut und programmiert" abgestimmt ist und intensiver in die Mikroprozessortechnik einsteigt.
Das Lehrinstitut Christiani betätigt sich seit über 50 Jahren auf dem Gebiet der Erwachsenenbildung durch Fernunterricht.

Bereits Mitte der siebziger Jahre brachte Christiani den Lehrgang „Mikroprozessor-Labor", basierend auf dem Mikroprozessor SC/MP, heraus. Über 5000 technisch Interessierte erarbeiteten sich mit diesem Lehrgang erfolgreich die ersten Kenntnisse im Programmieren von Mikroprozessoren in Maschinensprache. Angesichts dieser Tatsache kann man sicherlich damit rechnen, daß in Zukunft dem NDR-Kleincomputer-Bauer und -Anwender fundiertes Lehrmaterial zur Verfügung stehen wird.

Grundthematik des neuen Christiani-Lehrgangs ist natürlich der NDR-Computer. Der Lehrgangstitel lautet: „Mikrocomputer-Praxis mit dem NDR-Computer". Die Hardware, das Betriebsprogramm und die Programmierung des Z 80 in sedezimaler Form werden dem Computer-Anwender nähergebracht. Der Aufbau des Lehrgangs hält sich an ein seit vielen Jahren bewährtes Christiani-Prinzip: die Aufteilung des Lehrstoffs in verschiedene Fachgebiete. Wechselweise wird nur soviel eines Fachgebietes besprochen, wie zum Verständnis des nächsten notwendig ist. Viele Beispielprogramme vertiefen das Fachwissen, anhand von zahlreichen Aufgaben kann der Computer-Anwender das soeben erworbene Wissen sofort selbst überprüfen.
Für manchen, der beim Zusammenbau des Computers Lötschwierigkeiten hat oder sich mit Elektronik-Grundkenntnissen schwertut, hat Christiani bereits zwei Kurse parat: das „Lötpraktikum" und den „Kompakt-Kurs Elektronik".
Es sind noch weitere Christiani-Lehrgänge zum NDR-Computer geplant, wie ein Pascal-Kurs und ein Basic-Kurs. Der an das NDR-Kleincomputer-Basic angepaßte Basic-Kurs wird im April auf dem Markt sein. Durch die Zusammenarbeit mit dem Konstanzer Lehrinstitut darf man sicher sein, daß sich die Sendereihe „Mikrocomter – selbst gebaut und programmiert" in Zukunft noch größerer Einschaltquoten erfreuen wird, da dem Zuschauer über eine hervorragende Hardware hinaus nun auch noch optimales Lehrmaterial geboten wird.

Endlich -

die Bauelementepreise sinken! Statische 8 K Byte RAMs nun für 70,– erhältlich!

Die Bauelementeverknappung hat ein Ende. Fast alle Bauteile sind wieder zu vernünftigen Lieferzeiten und zu einigermassen "normalen" Preisen erhältlich.

Besonders drastisch ist der Preisverfall bei den statischen 8KByte RAMs, die pinkompatibel zu den 8 KByte EPROMs 2764 sind und auf der ROA64 eingesetzt werden. Die ICs waren bis November nur sehr schwer und sehr teuer zu erhalten, sind nun innerhalb einer Woche um 50 % billiger geworden und nun auch in grossen Stückzahlen verfügbar.

Wir geben die Verbilligung natürlich sofort weiter. Wir vermuten, dass dieser Preis bis Frühjahr 1985 stabil bleibt, danach könnten, da dann noch weitere Hersteller diese Bausteine im Programm haben, die Preise noch etwas fallen.

Auch zeichnet sich bei den EPROMs eine Verbilligung ab. Da wir ab 1985 zu den Eproms die komplette Dokumentation dazuliefern werden, können wir dies vermutlich ohne eine Anhebung des Preises tun. Endlich können wir auch leere Eproms zu günstigen Preisen liefern.

Wie der NDR-Klein Computer entstand

von Joachim Arendt Abteilungsleiter Schulfernsehen NDR

Etwa vier Jahre ist es jetzt her, als ich in der Ausbildungsstätte eines großen deutschen Industrieunternehmens der Mikroelektronik – ein Fünfzigjähriger – wieder die Schulbank drückte. Mein Lehrer war ein Mittdreißiger, heute Herausgeber dieser Zeitschrift und Inhaber einer Computerfirma.

Als 50jähriger nochmals auf der Schulbank?

Damit sind zugleich Probleme und Chancen der neuen Technik – der Mikroelektronik – und der neuen Wissenschaft – der Informatik – umrissen: Die Jungen sind den Älteren in der Sache unbestritten voraus und geben den Ton an; die Älteren müssen und dürfen noch einmal lernen, können die Gelegenheit eines neuen Anfangs genießen.
Ich meine das mit dem „Genießen" überhaupt nicht spöttisch und schon gar nicht bitter, obwohl ich mir bewußt bin, daß für viele Ältere die neue Technik auch den Verlust des vertrauten Arbeitsplatzes und auch den Entzug des gewohnten Einkommens bedeutet.
Die heute Älteren haben in den fünfziger und sechziger Jahren das lebenslange Lernen gefordert – eigentlich haben wir die Zukunftsprobleme damals recht genau vorausgesagt. Warum sind wir heute so überrascht, daß unsere Mahnung an die damals Älteren nun für uns gilt?
Der Neubeginn ist für die Jungen genauso hart, wie er uns erscheint. Wir Älteren haben sogar den Jungen einiges voraus – wir haben die Technik des Arbeitens „im Griff", wir können leichter Wesentliches aus dem Wust von Unwesentlichem herauslösen – wichtige Voraussetzungen, wenn man sich neue Wissensgebiete erarbeiten muß. Im Grunde lernt der Ältere leichter als der Jüngere; man muß eben nur „dabei" bleiben, das heißt „trainieren".

Der NDR-Computer entsteht

In einem Gespräch nach Kursabschluß waren sich Gerd Graf und ich schnell einig. Was wir in der Phase der hektischen Entwicklung von Mikroelektronik und Informatik brauchen, sind umfassende modulare Systeme, unbegrenzt vielseitige „Bausteine", aus denen man seinen speziellen Computer zusammenbauen kann, mit denen man Steuerungs- und Regelungsprobleme genauso lösen kann wie Aufgaben der Textbe- und -verarbeitung. Problemlösung mit Hilfe vorgegebener Bausteine – mit Programmbausteinen und mit Rechnerbausteinen – im Zusammenspiel der beiden gleichgewichtigen Bausteinarten.
Die zukünftigen Probleme und ob und wie sie mit Hilfe des Computers zu lösen sind, können wir heute weder ahnen noch gar voraussehen. Man kann heute bloß vermuten, daß die unzähligen kleinen Probleme nur mit kleinen, gut angepaßten Spezialrechnern zu lösen sein werden. Viele dieser Rechner werden die großindustrielle Fertigung nicht lohnen, also braucht man modulare, sehr vielseitige Systeme, die man selbst handhaben, die man selbst weiterentwickeln, mit denen man seine eigene, besondere Problemlösung aufbauen kann – mit Rechner- und Programmbausteinen.

NDR-Computer auch im Handwerksbetrieb

Das gilt insbesondere für den Handwerksbetrieb wie für das kleine, mittelständische Unternehmen, und das gilt für alle Formen der Ausbildung und der Weiterbildung Erwachsener: von den allgemeinbildenden zu den berufsbildenden Schulen, zu den Hochschulen und Forschungsstätten, den Volkshochschulen wie den innerbetrieblichen Weiterbildungsstätten.
„Hab' ich", sagte Gerd Graf – und meinte „unser" System – „genauer gesagt, ich kenne jemanden, der einen solchen Computer- und Sprachbaukasten fertig ‚im Schrank' hat. Er hat ihn mir zur Fertigung übergeben".
Damit ist der „Dritte Mann" im Spiel, der eigentlich der Erste ist und der Jüngste dazu: Rolf-Dieter Klein. Sein Problem war, ein System zur Hand zu haben, mit dem er im großen und ganzen die vorhandenen Personal-Computer nachbilden konnte, das jedoch offen genug war, der technischen Entwicklung zu folgen.
Ihm kam es zum Beispiel darauf an, neue Prozessortypen aufnehmen zu können, und er hatte die Entwicklung in Richtung 16- und 32-bit-Rechner im Auge. Dazu schuf er sich ein neues, übersichtliches Bussystem, besonders gut geeignet für Anfänger, aber auch für die Profis, die von einem auf ein anderes System springen müssen oder für die Praktiker, die sich für die Weiterentwicklung der Technik „fit" halten wollen und dazu übertragbare Strukturen brauchen.

Schnelle Vollgrafik – kein Problem

Und auf eine schnelle Vollgrafik kam es dem Rolf-Dieter Klein an: Damit man sich die Wirkung in Programmstrukturen am Rechner selbst anschaulich klar machen kann – und aus sehr praktischen Gründen. Ohne Simulationstechnik mit Hilfe von Grafik und ohne zeichnerische Entwicklung am Bildschirm ist moderne Technik nicht denkbar.
Damit ist doch eigentlich schon das Konzept umrissen – auf geht's!
Joachim Arendt

Mehr als nur Schaltungen nachbauen

Kompakt-Kurs Elektronik

Technik ist ohne Elektronik nicht mehr denkbar. Deshalb sind Grundlagen in der Elektronik heute unentbehrlich, sei's für das berufliche Weiterkommen oder um Spaß an der Hobby-Elektronik zu finden. Schwierig ist es mit der Elektronik nicht. Sie müssen die Funktion der Bauelemente kennen und Grundschaltungen dimensionieren können.

Genau das vermittelt Ihnen der Kompakt-Kurs Elektronik. Neben der Theorie kommt auch die Praxis nicht zu kurz. Mit dem zum Kurs gehörigen Experimentiermaterial werden zahlreiche Versuche durchgeführt.
Nach Durcharbeitung des Kurses (ca. 3 Monate werden dafür veranschlagt) können Sie den Christiani Test Elektronik machen – wir bestätigen Ihnen Ihre Kenntnisse in Form eines Zertifikats.

Der Lehrgang kostet DM 248,–.

Christiani Fortbildung

Technisches Lehrinstitut
Postfach 35 53182 · 7750 Konstanz
in Österreich: Ferntechnikum 6901 Bregenz

Coupon auf Postkarte aufkleben oder im Umschlag einsenden.
Sie erhalten sofort kostenlos ausführliches Informationsmaterial über den Kompakt-Kurs Elektronik.

Name, Vorname

Straße, Nr.

PLZ, Ort 53182

Der grosse loop Wettbewerb

Wir suchen die interessantesten Anwendungen mit dem NDR-Computer!

Mitmachen! Mitgewinnen!

Verlängert bis 31. 3. 1985

Wer kann teilnehmen?

Einzelpersonen oder Gruppen, die selbständig eine Anwendung mit dem NDR-Computer realisiert haben, können teilnehmen. Eine Altersbeschränkung gibt es nicht.

Was muß gemacht werden?

Wir suchen Anwendungen mit dem NDR-Computer, die mit diesem System realisiert worden sind. Reine Software-Lösungen oder Hardware/Software-Lösungen sind gleichberechtigt. Bei den nur Software-Lösungen ist die Programmiersprache egal; es sollten nur möglichst die spezifischen Eigenschaften des NDR-Computers ausgenützt werden, wie hochauflösende Graphik, Routinen der Grundprogramme usw.
Hardware/Software-Lösungen sollen soweit wie möglich mit dem NDR-Computer aufgebaut sein. Bei aufgebauter Zusatz-Hardware ist darauf zu achten, daß möglichst nur gebräuchliche Bauelemente eingesetzt werden.
Die Lösung muß komplett und ausführlich dokumentiert werden, die Software mit Struktogramm und dokumentierten Listings, die Hardware mit Schaltbildern und, falls nötig, Aufbauhinweisen.

Was muß eingesandt werden?

Zunächst nur die vollständige Dokumentation der Lösung, bei Hardware-Lösungen ein Photo des Aufbaues – Polaroid – genügt.

Wer sucht aus?

Die Jury besteht aus:
Rolf Dieter Klein, Entwickler, Gerd Graf
Hans-Joachim Arendt, Leiter des Schulfernsehens NDR
Der Rechtsweg ist ausgeschlossen, die Gewinner werden sofort direkt benachrichtigt.

Was kann man gewinnen?

1.Preis
ein Warengutschein imWert von DM 1.000.-

2. - 10. Preis
je ein Warengutschein im Wert von DM 100.-

Wann ist Stichtag?

NEU : 31. März 1985

Was geschieht mit Ihren Einsendungen?

Die Unterlagen verbleiben bei uns. Senden Sie uns deshalb bitte keine Originale. Mit der Teilnahme am Wettbewerb erteilen Sie uns das Recht, Teile oder komplette Einsendungen unter Nennung Ihres Namens zu veröffentlichen und zu verwenden. Die Auswertung Ihrer Einsendungen im Fernsehen ist möglich.

NEUHEITEN - NEUHEITEN - NEUHEIT

TAST2 - eine etwas grössere Tastatur für den NDR-Computer mit abgesetztem Zehnerblock und speziell an den NDR-Computer angepassten Zehnerblock ist ab sofort lieferbar.

Die Funktionstasten sind meist mit Kommandos des Texteditors im 68008-Grundprogramm belegt - ein Tastendruck genügt nun, um den Einfügemodus einzusachalten, in den Teil "Suchen" oder "Suchen und Ersetzen" zu gelangen oder den Zeichensatz zu wechseln. Die Tastatur wird von der Firma PREH speziell für den NDR-Computer gefertigt und ist komplett, mit Gehäuse, Rundkabel , 15-poligem Stecker und Anschlussanleitung lieferbar.

FLO2, die Floppy-Controller-Karte, ist bereits lieferbar. Das BIOS, das heisst die Anpassung für das Betriebssystem CP/M wird gerade von Herrn Klein fertiggestellt. CP/M zum NDR-Computer wird, komplett mit Dokumentation, ab Januar verfügbar sein.

RAM64/256 ist ebenfalls ab Januar lieferbar. Da besonders die Preise für die 256 KByte RAMs stark nach unten gehen, wird diese Karte für grössere Speicher immer interessanter.

RAM 64/256 Rolf-Dieter Klein

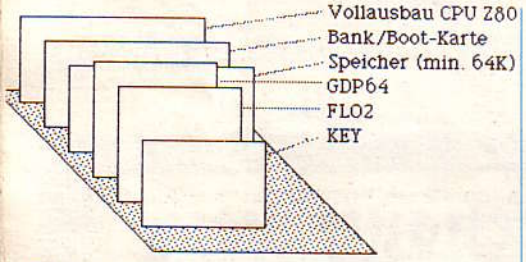
Auf der Baugruppe DYN 64/256 kann man wahlweise 64 K Bytes oder 256 K Bytes (!) RAM unterbringen, indem man entweder die 64K oder die neuen 256 K Bit RAM-Speicherbausteine verwendet. Die Karte ist durch Brücken auf den benötigten Adressbereich einstellbar.
Da die Karte kostengünstige dynamische RAM-Speicher verwendet, gibt es ein paar Dinge zu beachten. Zunächst einmal muß ein Refresh-Signal erzeugt werden. Die geschieht mit Hilfe von TTL-Bausteinen direkt auf der Baugruppe. Daher kann man die Baugruppe sowohl für den Z80 als auch für den 68008 verwenden. Zur Erzeugung des Refresh-Signals befindet sich ein eigener Quarzoszillator auf der Baugruppe, der mit 24MHz arbeitet. Der Refresh arbeitet wie folgt. Wenn der Prozessor mit Hilfe des MREQ-Quer-Signals zugreift, so wird zunächt die Refreshadresse an den Speicher ausgegeben und dann, falls auch die Adresse der Baugruppe mit der vom Prozessor gelieferten übereinstimmt, so wird der eigentliche Zugriff durchgeführt. Diese einfache Methode hat den Vorteil, das man für den Refresh keinen besonderen Refreshcontroller benötigt, hat aber auch den Nachteil, das man Zeit verschenkt, da immer zunächst ein Refresh-Zyklus durchgeführt wird und dann erst der Zugriff erfolgt. Bei 68008 muß man daher ca. 2 Wait-Zyklen einfügen, bei der Z80-Vollausbau-CPU erfolgt das automatisch. Bei neueren 68008-Karten, die den WAIT-Eingang auswerten, wird die CPU auch automatisch angehalten.

Ein weiterer Nachteil ist, daß die CPU dauernd Zugriffe durchführen muß, um den Refresh aufrechtzuerhalten. Die Zugriffe müssen aber nicht auf die RAM-Baugruppe erfolgen, sondern auch z.B. von einer ROA-Karte mit EPROMs. Man darf aber die CPU nicht längere Zeit anhalten, dann gehen Daten verloren.
Mit den neuen 256 K ICs bekommt man aber schnell einen großen Speicher, der auch für umfangreiche Projekte ausreicht.

CP/M kommt bald

Rolf Dieter Klein

CP/M ist ein sogenanntes Betriebssystem. Damit ist es möglich Floppy-Disks an den Computer anzuschließen. Es gibt dabei zwei verschiedene CP/M-Versionen. CP/M 80 ist das Betriebsystem für den Z80 und CP/M 68K ist das Betriebssystem für unsere 68008-CPU. Um CP/M auf dem NDR-Computer fahren zu können braucht man ein paar Voraussetzungen. Zunächst das Z80-CP/M.



CP/M 80 - Konfiguration

Man benötigt die Vollausbau-CPU, auf der sich der Prozessor Z80 befindet, sowie Bustreiber, um den gesamten Speicher zu adressieren. Dann braucht man die BANK/BOOT-Baugruppe. Sie enthält Fassungen für EPROMs oder RAMs. Auf diese Baugruppe wird ein EPROM mit dem Namen FLOMON in Fassung 0 gesteckt. Das FLOMON-EPROM enthält alle Unterprogramme für den Floppy-Betrieb, sowie einen umfangreichen Treiber um die GDP-Baugruppe als Datensichtgerät und Graphikausgabe verwenden zu können. Dabei sind Steuerzeichen für die Cursorpositionierung genauso vorhanden, wie für das Zeichnen von Linie, Kreisen, Rechtecken etc. Für die, die die TERM1 aus der Zeitschrift MC kenn, sein gesagt, daß sich praktisch die gesamte TERM-Software im FLOMON befindet und den NDR-Computer damit zum MC-Computer kompatibel macht, was den Softwareaustausch schon stark erleichtert. Auf der BANK/BOOT-Karte befinden sich dann auch noch zwei kleine 2K RAM-Bausteine, die vom FLOMON als Zwischenspeicher verwendet werden. Darin werden zum Beispiel alle Zeichen einer Textseite gespeichert, um einen Bildschirmscroll durchführen zu können. Ferner enthält die BANK/BOOT-Karte eine Logik, mit der man auch beim Z80 1 Megabyte Speicher adressieren kann. Dann noch das wichtigste, die Boot-Logik. Das CP/M-Betriebsystem verlangt nämlich RAM von Adresse 0 ab. Nun wissen aber alle, die schon mit dem Z80 gearbeitet haben, daß dort nach dem Spannungseinschalten und RESET das EPROM mit dem Programm sitzen muß, denn der Z80 beginnt die Programmausführung immer auf Adresse 0. Die Boot-Logik ermöglicht es, das RAM zunächst auszublenden, wenn ein RESET erfolgt. Später kann es dann aktiviert werden, wenn Teile des FLOMON in den höheren Speicher (F000H) kopiert werden. Dort wird dann die untere Hälfte dazugeschaltet, denn die BOOT-Logik kann den Bereich 0 bis 7FFFH ausblenden. Es wird aber nur ein Teil des FLOMON-Programms kopiert, so daß die umfangreichen Routinen keinen RAM-Platz verbrauchen. Bei Bedarf wird dann der untere EPROM-Bereich wieder eingeblendet, wenn das CP/M z.B. eine Ausgabe auf den Bildschirm durchführt.
Auf der Bank/Boot-Baugruppe ist noch ein Sockel frei, und dahin kann man z.B. das Z80-Grundprogramm stecken, das dazu in einer verschobenen Form geliefert werden kann. Dann kann man z.B. zu Testzwecken auch mit dem Grundprogramm auf Fehlersuche gehen, den FLOMON enthält ausser der Möglichkeit das Grundprogramm aufzurufen, keine solche Möglichkeit.
Nun zum RAM. Man benötigt 64K Byte RAM und kann dazu entweder die ROA-Baugruppe mit den statischen RAMs oder die neue RAM 64/256 Baugruppe verwenden, wie auch jede andere RAM-Baugruppe, die verfügbar ist.
Nun braucht man natürlich ein Floppy-Laufwerk und der NDR-Computer, genauer die FLO2-Baugruppe, die den Floppy-Controler enthält unterstützt verschiedene Laufwerkstypen. Man kann alle 8", 5 1/4", 3 1/2" oder 3" Laufwerke mit einfacher oder doppelter Dichte oder ein- oder doppelseitig, verwenden. Folgende Laufwerke werden aber zunächst bevorzugt behandelt:

8 Zoll	einfache Dichte, IBM-kompatibel (240K Bytes)
5 1/4 Zoll	40 Spuren, einseitig, doppelte Dichte, ECMA70-Format. (140 K Bytes)
5 1/4 Zoll	80 Spuren, doppelseitig, doppelte Dichte. (780 K Bytes)
3 Zoll	40 Spuren, einseitig, doppelte Dichte, ECMA70-Format. (140 K Bytes)

Andere Formate folgen.
Das 8 Zoll Format ist übrigends am Besten geeignet um CP/M-Software aus Amerika zu beziehen, bei den anderen bekommt man größere Schwierigkeiten. Auf der CP/M-Diskette hat man aber ein Hilfsprogramm, um eine Anpassung an Standard-Formate durchzuführen.

Bei CP/M 68K für den 68008-Rechner sieht das ganze ähnlich aus. Zunächst aber braucht man für das CP/M 68K einen größeren Speicher, nämlich mindestens 128 K Bytes. Dann braucht man die Bank/Boot-Karte, da auch beim CP/M für den 68008 auf Adresse 0 RAM liegen muß. Ein verschiebbares Grundprogramm sitzt dann aber nicht auf der Bank/Boot-Karte, sondern auf einer ROA64-Baugruppe, die auch ein 8K RAM für lokalen Arbeitspeicher besitzen muß. Die ROA64 liegt dann auf der Adresse $E0000, also ganz am Schluß des Speicherraums. Auf der Bank/Boot Karte sitzt nur ein EPROM, das einen Sprung auf die Startadresse des Grundprogramms auf $E0000 enthält, sowie einen Mechanismus um das RAM auf Adresse 0 freizugeben. Im Grundprogramm für das CP/M sind alle Routinen für den Betrieb mit der Floppy enthalten, und es ist vollkommen verschiebbar.

So kann man dann auch von CP/M 68K aus alle Unterprogramme des Grundprogramms weiter nutzen, wie man es gewohnt war. Übrigends darf man bei der Bank/Boot-Karte nicht vergessen vier Leitungen, die zu den Adressen A16-A19 führen, aufzutrennen, wenn man sie mit dem 68008 verwendet, da diese Adressen vom 68008 schon belegt sind, und nicht wie beim Z80, durch die Karte erzeugt werden müssen. Für die Floppy-Formate gilt das Gleiche, wie beim Z80. Übrigends kann man beim 68008 auch die ältere Banksel-Baugruppe verwenden, wie sie noch im Buch "Mikrocomputer selbstgebaut..." beschrieben ist.

Das CP/M 68K kann dann, sobald es verfügbar ist, z.B. direkt vom Franzis-Verlag Software-Service oder von den Lieferfirmen bezogen werden, genauso wie die neuen EPROM-Sätze.

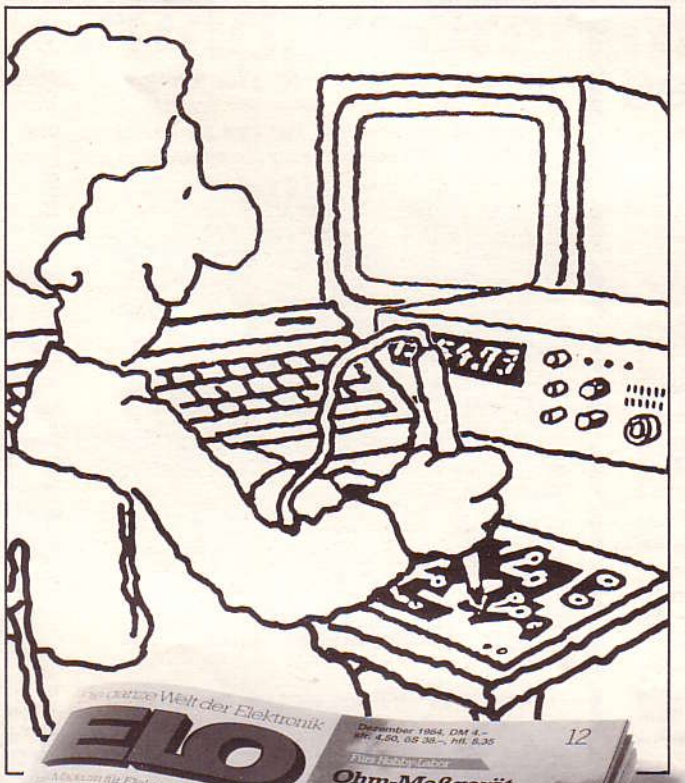
Elektronik verstehen, Computer verstehen – mit der neuen ELO fängt der Spaß an moderner Technik erst so richtig an.

In der ELO findet der Leser leicht zu realisierende **Bauanleitungen,** die ihm Elektronik praktisch nahebringen und die oft **neue Einsatzbereiche für den Heimcomputer** erschließen: Messen, Regeln, Steuern, Erfassen, Auswerten.

Die ELO regt an, mit dem Lötkolben umzugehen und Hardware für eigene Zwecke zu verändern oder selbst zu bauen.

Leicht verständliche **Beiträge zu den Grundlagen der Elektronik** machen technische Zusammenhänge klar und damit nutzbar.

Im Magazinteil der ELO und in spannenden Reports erfährt der Leser,



NEU
Erweiterter Inhalt, neue Optik!

was mit Elektronik und Computern alles möglich ist. In der Forschung, in der Industrie, im Verkehrswesen, im Umweltschutz.

ELO-**Tests** und ELO-**Marktübersichten** sind für den Leser Entscheidungshilfen vor einem Kauf bzw. Orientierungshilfe im vielfältigen Angebot elektronischer Geräte aus dem HiFi- und Videobereich, der Meßtechnik und der Mikrocomputerei.

Die ELO gibt es an jeder größeren Zeitschriften-Verkaufsstelle.
Sie können die ELO im Rahmen unseres Kennenlern-Angebotes zweimal kostenlos prüfen.

ELO
Das Magazin für Elektronik und Computer

Ich möchte von Ihrem **ELO-Kennenlern-Angebot**
Gebrauch machen: Schicken Sie mir die beiden neuesten Ausgaben kostenlos. Informiere ich Sie danach nicht anders, abonniere ich die ELO ab .....
zum Jahrespreis von DM 46,80 (im Ausland DM 55,20) incl. Porto.

Name

Beruf

Straße

PLZ/Ort

Datum Unterschrift

Die Kündigung ist jeweils 8 Wochen vor Ablauf des Abonnements möglich.
Wichtig: Nach Erhalt des 2. kostenlosen Heftes kann ich innerhalb von 10 Tagen durch einfache schriftliche Mitteilung an den Verlag von einem Abonnement Abstand nehmen.

Bitte hier Ihre zweite Unterschrift L01

Im Umschlag an:
Franzis' Franzis Verlag, Postfach 37 01 20, 8000 München 37.
Das genügt. Wir schicken Ihnen Ihre ELO sofort.

TIPS

Schriftgröße

Rolf-Dieter Klein

Wenn man für Unterrichtszwecke große Schrift im Texteditor des Grundprogramms 68K braucht, so gibt es einen einfachen Weg. Man kann die Routine SIZE aufrufen und damit die Textgröße einstellen.

```
start:          ; Neues Bildformat
                ; einstellen
 move #$32,d0   ; dx=3, dy=2
 jsr @size
 rts
```

Textstart=009000 Fenster=009000 Tor=009000 einf aner CTRL-J=Hilfe

Im Register D0 wird dazu der Code für die Größe übergeben. Wenn man den Wert sedezimal angibt, so steht die Breite in X-Richtung auf der Linken Seite nach dem Dollarzeichen und die Breite in Y-Richtung auf der rechten Seite.
Damit die neue Größe aktiv wird, übersetzt man das Programm und startet es mit dem Namen "START". Wenn man dann anschließend zurück in den Editor geht, ergibt sich folgendes Bild:

```
start:          ; Neues Bildf
                ; einstellen
 move #$32,d0   ; dx=3, dy=2
 jsr @size
 rts
```

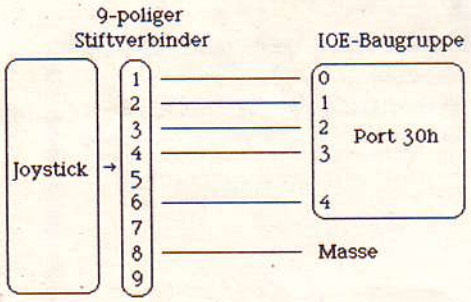
Textstart=009000 Fenster=009000 Tor=009000 aner CTRL-J=Hilfe

Die Schrift erscheint nun vergrößert auf dem Bildschirm. Wenn man aber den Cursor nach unten bewegt, so verschwindet er vom Bildschirm. Die oberste Zeile ist übrigends ggf. angeschnitten, was aber nicht weiter stört.
Wenn man wieder eine normale Ausgabe haben möchte, so kann man SIZE entweder mit $21 oder $11 (80 Zeichen pro Zeile) aufrufen, oder man drückt die RESET-Taste.

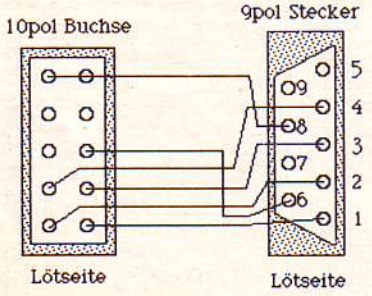
Anschluß eines Joysticks

Rolf Dieter Klein

Auf dem Markt befinden sich eine Vielzahl von sogenannten Joysticks. Sie werden vorwiegend für Spiele gebraucht und enthalten meist vier Kontakte, mit denen die Richtung "oben", "unten", "links" und "rechts" erkannt werden, wenn man den Joystick in die entsprechende Richtung bewegt. Die Joysticks enthalten ferner meist noch einen Feuerknopf, der als fünfter Kontakt herausgeführt ist. Nun hat sich bei den Joysticks, bis auf wenige Ausnahmen eine Art Steckernorm herausgestellt. Man verwendet dabei eine 9-polige Buchse, deren Stiftbelegung festliegt. Nun kann man solch einen Joystick relativ einfach an die IOE-Baugruppe anschließen, wenn man sich das passende Gegenstück im Handel besorgt. Damit nun nicht jeder eine eigene Belegung erfindet, soll in diesem Artikel eine Belegung festgelegt werden, an die man sich halten kann. Dann sind Programme leicht austauschbar, wenn sie mit einem Joystick arbeiten.
Das nachstehende Bild zeigt die Anschlußbelegung. Die Kontakte 5,7 und 9 bleiben offen.



Man kann die Verbindung direkt auf der IOE-Baugruppe auflöten, oder besser eine weitere Steckverbindung verwenden. Dazu lötet man in die Lötreihen der IOE-Baugruppe Stiftleisten ein, und verwendet eine 10- polige Buchse, die man dann direkt auf die Stiftleisten aufsteckt. Das nächste Bild zeigt die Verdrahtung von der 10-poligen Buchse mit dem 9-poligen Stecker.



Man kann übrigends auch ein kleines Flachbandkabel auf die 10-polige Buchse auffpressen, wenn man eine entsprechende Buchse verwendet. Die Leitungen des Flachbandkabel entsprechen dann der im Bild gezeigen Reihenfolge der einzelnen Leitungen.

Wenn man zwei Joysticks verwenden will, so verwendet man für den zweiten Joystick den Port 31H, also den zweiten Eingabeport auf der IOE-Baugruppe.

Die Belegung der Bits ist auch festgelegt und lautet wie folgt:

Bit 0	oben
Bit 1	unten
Bit 2	links
Bit 3	rechts
Bit 4	Feuertaste

Wenn man den Anschluß testen will, so geht man dazu in das Menue "IO-LESEN" und gibt beim Z80 die Adresse 30 (sedezimal) und beim 68008 die Adresse ($FFFFFF30) an.

Dann drückt man die Tasten SHIFT und D, um die Eingabe dauernd durchzuführen. Wenn man nun den Joystick bewegt, so werden die entsprechenden Bits auf 0 gehen. Wenn man z.B. die Feuertaste drückt ergibt sich das Bitmuster 11101111 auf dem Bildschirm, Bit 5 ist also 0.

Das Signal 0 erscheint, weil man einen Kontakt nach 0V dürchführt und 0V der logischen 0 entspricht. Ein offener Eingang führt aber bei einem TTL-Eingang zur logischen 1.

Übrigends kann man mit diesem Anschluß zum Beispiel direkt das Mondlandespiel aus der Fernsehfolge spielen.

FLOH MARKT

NDR-Klein-Computer (CPU 68K)
Pac-Boy; 10 Bilder; 4 Levels
EPROM gegen Scheck DM 60,–
Seele, Fr.-Kenkel-Str. 2
2848 Vechta

Braunschweig! Gibt es hier eine Arbeitsgruppe für NDR-Klein-Computer, oder wer hat Interesse an einer solchen ?
Telefon: 05 31 / 33 28 62

NDR-Klein-Computer
REPARIERE - TESTE - HELFE !
Telefon: 0 41 81 / 3 37 07

Die Bibliotheksfunktion beim 68 008 - Grundprogramm

Rolf-Dieter Klein

Die Bibliotheksfunktion ist eine interessante Möglichkeit, den Befehlssatz 68008-Grundprogramms zu erweitern. Dabei wird ein Anwenderprogramm im Speicher abgelegt und man kann es dann über die Bibliothek starten. Damit das möglich ist, muß man vor das eigentliche Programm eine Anzahl von Bytes mit dem Code: $55AA0180 schreiben. Das Grundprogramm durchsucht nämlich den Speicher im Abstand von 1024 Bytes nach diesem Muster. Wenn das Muster übrigends einmal zufällig im Speicher auf einer solchen 1K-Grenze stehen sollte, so ergibt sich eine fehlerhafte Anzeige in dem Bibliotheksmenue.

Nach der Kennung folgt der Name des Programms, der genau 8 Zeichen lang sein soll. Dann die Startadresse und die Programmlänge. Die Programmlänge wird aber vom Grundprogramm nur auf dem Bildschirm ausgegeben, und nicht weiter ausgewertet. Sie dient nur dem Benutzer als Information. Dann folgt ein Byte in dem festgelegt wird, ob das Programm relokativ ist, oder absolut.

Relkokative Programme sind auf Adresse 0 übersetzt und verwenden keine absoluten Adressierarten . Danach folgen 3 freie Bytes und 2 Langworte, die für Erweiterungen reserviert sind.

Danach kann ein neuer Block mit Kennung folgen, so kann man mehrere Eingänge in ein Programm durchführen, ohne von den 1K-Grenzen abhängig zu sein.

Nun ein Beispiel. Das Bild zeigt einen Eintrag mit absoluter Adressierung.

```
* Bibliotheksfunktion, absolut
org $9c00
 dc.l $55aa0180          ; Kennung
 dc.b 'senkr.  '         ; Name
 dc.l senkrecht          ; Programmstart
 dc.l ende-waagrecht     ; Laenge
 dc.b 0                  ; absolut
 dc.b 0,0,0
 dc.l 0,0
 dc.l $55aa0180          ; weitere Eintrag
 dc.b 'waagr.  '         ; Name
 dc.l waagrecht          ; Programmstart
 dc.l ende-waagrecht     ; Laenge
 dc.b 0                  ; absolut
 dc.b 0,0,0
 dc.l 0,0
 dc.l 0                  ; kein weiterer
waagrecht: move #90,d0   ; nur als Beispiel
           jsr @drehe    ; hier beliebiges
senkrecht: move #100,d0  ; Anwenderprogramm
           jsr @schreite
           rts
ende:
        end
```

Textstart=009000 Fenster=009000 Tor=009004 aber CTRL-J=Hilfe

Wenn man das Programm übersetzt, so wird es auf Adresse $9C00 abgelegt. Wenn man dann die Bibliotheksfunktion aufruft, so ergibt sich folgendes Bild:

```
Bibliothek

Name   : senkr.
Start  : 009C4E
Laenge : 000016

Starten J = JA
cr = weiter, M = Menue
```

```
Bibliothek

Name   : waagr.
Start  : 009C44
Laenge : 000016

Starten J = JA
cr = weiter, M = Menue
```

Absolute Programme sind einfacher herzustellen als relokative Programme, das nächste Bild zeigt den Bibliothekskopf mit einem relativen Programm.

```
* Bibliotheksfunktion, relativ
org $0                   ; Start=adr 0
offset $9c00             ; im Speicher
 dc.l $55aa0180          ; Kennung
 dc.b 'senkr.  '         ; Name
 dc.l senkrecht          ; Programmstart
 dc.l ende-waagrecht     ; Laenge
 dc.b 1                  ; relativ
 dc.b 0,0,0
 dc.l 0,0
 dc.l $55aa0180          ; weitere Eintrag
 dc.b 'waagr.  '         ; Name
 dc.l waagrecht-32       ; Programmstart
 dc.l ende-waagrecht     ; Laenge
 dc.b 1                  ; relativ
 dc.b 0,0,0
 dc.l 0,0
 dc.l 0                  ; kein weiterer
waagrecht: move #90,d0   ; nur als Beispiel
           jsr @drehe    ; hier beliebiges
senkrecht: move #100,d0  ; Anwenderprogramm
           jsr @schreite
           rts
ende:      end
```

Textstart=009000 Fenster=009000 Tor=009004 aber CTRL-J=Hilfe

Dabei gibt es eine Reihe von Unterschieden. Zunächst beginnt das Programm mit einer ORG $0 - Anweisung. Das Programm wird also so übersetzt, als wolle man es auf Adresse 0 ablegen. Mit der nächsten Anweisung OFFSET $9C00 wird aber verhindert, das der Maschinencode auch tatsächlich dort hin gelegt wird. Er gelangt auf Adresse $9C00. Denn der Wert von OFFSET wird auf die aktuelle Programmzähleradresse bei der Übersetzung addiert, und dorthin erfolgt die Ablage. Nun folgt der Bibliothekskopf, der diesmal eine 1 nach der Längenangabe erhält, um dem Grundprogramm mitzuteilen, daß es sich um ein relokatives Programm mit relativen Angaben handelt. Weil alle Angaben relativ zur ersten Kennung sind, muß bei der zweiten Startadresse der Wert 32 subtrahiert werden (bei der dritten dann 2*32 etc.).

Das Programm selbst hat sich in diesem Fall nicht weiter geändert.
Hier noch ein paar Hinweise zum Schreiben von relativen Programmen.
a) Tabellen und Konstante.

1. Absolutes Programm:

```
    LEA TABELLE,A0
    ...
    TABELLE:
    ...
```

2. Relokatives Programm:

```
    LEA TABELLE(PC),A0
    ...
    TABELLE:
```

b) Variable und Felder.

1. Absolutes Programm:

```
    MOVE ALPHA,D0
    ADD #1,D0
    MOVE D0,ALPHA
    ...
    ALPHA: DC.W 0
```

2. Relokatives Programm mit Variablen im gleichen Bereich

```
    BASIS EQU 0       ; wie ORG-Anweisung
    ...
    LEA BASIS,A5      ; freies Adressregister
    MOVE ALPHA(A5),D0 ; laden
    ADD #1,D0         ; z.B. Operation
    MOVE D0,ALPHA(A5) ; speichern
    ...
    ALPHA: DC.W 0     ; Variablengebiet
```

3. Relokatives Programm mit Variablen im getrennten RAM-Bereich:

```
    MOVE ALPHA,D0
    ADD #1,D0
    MOVE D0,ALPHA
    ...
    ORG $A000         ; Variable hier absolut
    OFFSET 0
    ALPHA: DC.W 0
```

Wenn man die Programm in EPROMs ablegt, so kann man die absolute Adressierung bei Variablen verwenden. Die EPROMs mit dem Bibliotheksprogramm kann man dann dennoch irgendwo hin stecken, das Programm wird immer korrekt ausgeführt.
Bei relokativen Programmen darf man keine JMP oder JSR-Befehl verwenden, wenn nicht gerade eine Grundprogrammroutine angesprungen wird. Alle Adressierarten müssen wie schon gesagt relativ sein. Beim 68008 ist das immer möglich, wenn man auch nicht immer gleich den passenden Befehl findet. Übung hilft auch hier weiter.
Denn noch eine Bitte. Man sollte möglichst den TRAP-Befehl verwenden, um Grundprogrammroutinen aufzurufen, sonst läuft das Programm vielleicht nicht mehr, wenn mal eine neue Grundprogrammversion kommt, denn im EPROM liegen dann die Adressen fest.

Das Z80-Grundprogramm

Das Z80-Grundprogramm bietet eine Reihe von Befehlen, deren Beschreibung im Buch "Mikrocomputer selbstgebaut ..." bisher sehr kurz gekommen ist. Daher hier eine ausführlichere Darstellung mit kleinen Beispielen. Alle Programmbeispiele können mit dem Menue "AENDERN" z.B. ab Adresse 8800 eingegeben werden.

SCHREITE (Schildkrötensprache)

Der Befehl SCHREITE läßt die Schildkröte um die im HL-Registerpaar angegebene Anzahl weiterlaufen. Dabei schreitet sie in der aktuellen Blickrichtung weiter. Die Zahl kann auch negativ sein, dann erfolgt die Angabe im Zweierkomplement. Das Grundprogramm kann aber die Umrechnung direkt vornehmen, wenn man z.B. -#100 eingibt.

Beispiel:

```
21 #100.W     ; dezimale Zahl Hundert
CD SCHREITE   ; Befehl Schreite ausführen
C9            ; Programmende
```

DREHE (Schildkrötensprache)

Die Schildkröte wird um einen im HL-Registerpaar angegebenen Winkel gedreht. Der Winkel ist dabei in Grad angegeben. Eine positive Zahl dreht die Schildkröte entgegen dem Uhrzeigersinn, also mathematisch positiv.

Beispiel:

```
21 -#90.W     ; um 90 Grad im Uhrzeigersinn
CD DREHE      ; drehen
C9
```

HEBE (Schildkrötensprache)

Nach Aufruf des Befehls hinterlässt die Schildkröte keine Schreibspur mehr.

Beispiel:

```
21 #10.W      ; 10 Schritte
CD SCHREITE   ; mit Schreibspur
CD HEBE       ; dann ohne
21 #10.W      ; Schreibspur
CD SCHREITE   ; neue Position anfahren
C9
```

SENKE (Schildkrötensprache)

Wenn man die Schreibspur mit Hebe aufgehoben hat, wird sie durch diesen Befehl wieder eingeschaltet.

Beispiel:

```
21 #10.W      ; erst mal schreiten
CD SCHREITE
CD HEBE       ; dann Stift hoch
21 #10.W      ; ohne Schreibspur
CD SCHREITE   ; schreiten
CD SENKE      ; dann wieder einschalten
21 #10.W      ; und wieder Linie zeichnen
CD SCHREITE
C9            ; Programmende
```

SCHLEIFE
ENDSCHLEIFE

Damit kann man Schleifenkonstruktionen realisieren. Dazu wird die Befehlssequenz die man mehrfach ausgeführt haben möchte zwischen SCHLEIFE und ENDSCHLEIFE genommen. Im Registerpaar HL lädt man zuvor die Anzahl der Schleifendurchläufe. Man kann mit diesen beiden Befehlen auch Schleifen mehrfach ineinander schachteln.

BEISPIEL:

```
21 #72.W      ; äussere Schleife
CD SCHLEIFE   ; 72 Mal durchlaufen
21 #36.W      ; innere Schleife
CD SCHLEIFE   ; 36 Mal durchlaufen
21 #10.W      ; Kreis annähern
CD SCHREITE   ;
21 #10.W      ;
CD DREHE
CD ENDSCHLEIFE ; Ende der inneren Schleife
21 #5.W       ; wird 72 Mal ausgeführt
CD DREHE      ;
CD ENDSCHLEIFE ; Ende der äusseren Schleife
C9            ; Programmende
```

SET (Schildkrötensprache)

Damit kann man die Schildkröte auf eine absolute Position setzen. Im Registerpaar HL wird dazu die X-Koordinate übergeben (Bereich 0 bis 511) und im Registerpaar DE die Y-Koordinate (auch 0 bis 511, jedoch ist nur jeder zweite Punkt auf dem Bildschirm unterscheidbar) und im Registerpaar BC steht der Winkel in den die Schildkröte blickt. Dabei zeigt die Schildkröte bei einem Winkel von 90 Grad nach oben.

Beispiel:

```
21 #256.W     ; HL:=256 , X-Koordinate
11 #300.W     ; DE:=300 , Y-Koordinate
01 #180.W     ; BC:=180 , Winkel
CD SET
C9
```

Die Schildkröte zeigt danach nach links.

MOVETO

Damit kann man die Koordinatenregister des Graphikprozessors setzen. HL ist dabei die X-Koordinate (0 bis 511 sichtbar) und DE die Y-Koordinate (0 bis 255 sichtbar). Hier wird im Gegensatz zur Schildkrötengraphik die physikalische Adressierung gewählt, daß heißt ein Bildpunkt enspricht dem Unterschied 1. Bei der Schildkrötengraphik wurde, um symmetrische Bilder zur Erhalten, die Y-Koordinate bei der Umrechnung durch 2 dividiert.

Beispiel:

```
21 #256.W     ; X:=256
11 #128.W     ; Y:=128
CD MOVETO     ; Positionieren
CD WAIT       ; warten bis GDP bereit
3E 80         ; Befehl für Einzelpunkt
D3 70         ; an Graphikprozessor ausgeben
C9            ; Programmende
```

Das kleine Programm setzt einen Punkt auf dem Bildschirm in die Bildmitte.

DRAWTO

Von der letzten Position, die im Koordinatenregister des GDPs (z.B. durch MOVETO) gespeichert ist wird eine Linie zur neuen Position gezeichnet. Die neue Position wird durch HL und DE bestimmt. HL enthält dabei die X-Koordinate (0 bis 511) und DE die Y-Koordinate (0 bis 255).

Beispiel:

```
21 0.W        ; X:=0
11 0.W        ; Y:=0
CD MOVETO     ; erst mal Positionieren
21 #511.W     ; X:=511
11 #255.W     ; Y:=255
CD DRAWTO     ; Linie zeichnen
C9            ; Programmende
```

Das Z80-Gru

Rolf-Diet

SCHR16TEL (Schildkrötengraphik)

Dieser Befehl arbeitet wie der SCHREITE-Befehl, jedoch wird im HL-Registerpaar 1/16 Schrittweite angegeben. Dies ist möglich, da das Grundprogramm intern alle Koordinaten für die Schildkrötengraphik mit 16-facher Genauigkeit speichert. Dieser Befehl ist z.B. für die Realisierung unterschiedliche Kreisdurchmesser nötig. Man achte bei der Anwendung des Befehls darauf, daß sich bei ungünstiger Verwendung arithmetische Fehler häufen können. Man sollte daher versuchen zunächst mit dem SCHREITE-Befehl auszukommen und nur im Notfall den SCHR16TEL Befehl anzuwenden.

Beispiel:

```
21 #36.W      ; Kreis als 36-Eck
CD SCHLEIFE   ;
21 #50.W      ; SCHRITTWEITE laden
CD SCHR16TEL
21 #10.W      ; Gesamtwinkel muß 360 Grad
CD DREHE      ; sein für Kreis.
CD ENDSCHLEIFE
C9            ; Programmende
```

Formel zur Errechnung der Schrittweite für den SCHR16TEL-Befehl, für das obige Programm aus gegebenem Radius:

SCHRITTWEITE := (R * 1000)/ 358

WRITE

Damit kann man eine Text auf dem Bildschirm ausgeben. Der Text wird zusammen mit weiteren Parametern im Speicher wie folgt abgelegt:

```
x.W           ; X-Koordinate des Textanfangs
y.W           ; Y-Koordinate des Textanfangs
höhe.W        ; Textgröße als Code :
                breitey*16+breitex
schräge.w     , Schriftrichtung, normalerweise
                0 (4,8,12 möglich)
"BELIEBIGER TEXT" ; Text in ASCII
0             ; Ende des Textes
```

Im Registerpaar HL wird vor Aufruf des Befehls WRITE die Adresse des Speicherblocks mitgeteilt auf der sich der Text befindet.

Beispiel:

```
21 8900.W     ; Adresse des Textblocks
CD WRITE      ; Befehl ausführen
C9            ; Programmende
...
8900:         ; Startadresse für Textblock
#10.W         ; X-Koordinate
#100.W        ; Y-Koordinate
33.B          ; Schriftgröße y,x = 3
0.B           ; Schriftrichtung Standard
"Hallo"       ; Text
0             ; Endekennung
```

Der Text wird normalerweise Hell auf dunklem Grund ausgegeben. Das Programm WRITE richtet sich aber nach dem Zustand des PEN/ERASER-Bits im

ndprogramm

r Klein

Graphikprozessors. Dieses Bit kann man auch gezielt stellen.

Beispiel:

```
CD WAIT      ; warten bis GDP bereit
3E 01        ; ERASER auswählen
D3 70        ; an GDP-Port
C9           ; Ende ERASERSELEKT

CD WAIT      ; warten bis GDP bereit
3E 00        ; PEN auswählen
D3 70        ; an GDP-Port
C9           ; Ende PENSELEKT
```

Wenn man den ERASER auswählt kann man übrigends gezeichnete Linien auch wieder weglöschen. Dabei darf aber nicht die Schildkröte eingeblendet sein, denn für die Einblendung der Schildkröte wird der GDP wieder auf PEN geschaltet.

READ

Damit kann ein Text in den Speicher eingelesen werden. Dazu wird auch wieder ein Parameterblock verwendet, der ähnlich wie beim WRITE-Befehl aussieht.

```
x.W          ; X-Koordinate des Textanfangs
y.W          ; Y-Koordinate des Textanfangs
höhe.W       ; Textgröße als Code :
               breitey*16+breitex
0.B          ; nur eine Schriftrichtung erlaubt
max.B        ; maximale Anzahl der
               einzulesenden Zeichen
aktuell.B    ; dort steht nachher die
               tatsächliche Anzahl der Zeichen
--------     ; Speicherbereich freihalten,
               dort kommt der Text hin
```

Beim Aufruf des READ-Befehls steht im Registerpaar HL die Adresse des Parameterblocks. Im C-Register kann man zusätzlich eine Umrahmung des Textfenster verlangen, wenn man den Wert 1 im Register C ablegt. Wenn der Wert 0 im Register C steht, wird keine Umrahmung ausgegeben. Die Breite und Höhe des Rahmens wird durch die Werte "max" und "höhe" automatisch berechnet.

Beispiel:

```
21 8900.W    ; Adresse des Parameterblocks
0E 01        ; mit Rahmen
CD READ      ; einlesen
C9           ; Programmende
...
8900:        ; Dort Rest eingeben
*10.W        ; X-Koordinate
*20.W        ; Y-Koordinate
22.B         ; Schriftgröße y,x = 2
0.B          ; immer 0
6.B          ; max-Zahl
0.B          ; wird erst später besetzt
0 0 0 0 0 0 0 ; Platz für Text
```

Nach dem Start erscheint ein umrahmtes Textfenster auf dem Bildschirm. Nun kann man einen beliebigen Text eingeben, der maximal 6 Zeichen lang sein kann. Mit der Taste CR wird die Eingabe beendet, oder, wenn man mehr als 6 Zeichen eingibt. Im Parameterblock wird danach die Anzahl der Zeichen gültig sein und der Text im restlichen Block abgelegt sein. Der Text wird ausserdem durch eine 0 beendet, die hinter dem Text steht, so daß man immer ein Byte mehr im Speicher reservieren muß, als die "max"-Zahl angibt. Das zum Abbruch führende Zeichen ist zusätzlich im Register A enthalten, z.B. der Code für CR, wenn man ein Carriage Return eingegeben hat. Jedes Steuerzeichen führt zum Abbruch, ausser das Delete- und Backspace-Zeichen, mit dem man im Texte im Fenster korrigieren kann.

CI

Eingabe eines Zeichens von der Tastatur im ASCII-Code. Das Zeichen wird im Register A im ASCII-Code abgelegt.

Beispiel:

```
TERMINAL:=$      ; Name des Programms
    CD CI        ; Zeichen einlesen
    CD WAIT      ; warten bis GDP fertig
    D3 70        ; Ausgabe als Kommando
    C3 TERMINAL  ; wieder zurück
```

Wenn man einen Text eingibt, so erscheint er auf dem Bildschirm, jedoch ist bei diesem einfachen Programm noch kein Zeilenvorschub möglich.

CSTS

Damit kann man prüfen ob eine Taste auf der Tastatur gedrückt wurde. Wenn ja, so erscheint im Register A der Wert FF, sonst der Wert 0.

Beispiel:

```
DAUER:=$     ; Programmname
    21 *1.W  ; Einen Schritt gehen
    CD SCHREITE
    21 *1.W  ; um Kreis zu zeichnen
    CD DREHE
    CD CSTS  ; solange, bis
               Taste gedrückt wird
    B7       ; (OR A, Flags bestimmen)
    CA DAUER ; =0, dann nochmals
    C9       ; Sonst beenden
```

RI

Wie CI, jedoch wird ein Zeichen von der CAS-Baugruppe gelesen. Das Zeichen erscheint auch im Register A.

Beispiel:

```
WARTE:=$
    CD RI    ; Zeichen lesen
    FE FF    ; warten bis FF auftaucht
    C2 WARTE ; nein, dann zurück
    C9       ; Ende gefunden
```

PO

Zeichen an die CAS-Baugruppe ausgeben. Dabei steht das Zeichen im Register C. Wenn man selbst einmal Daten auf dem Kassettenrekorder mit einem eigenen Programm abspeichern will, so muß man darauf achten, daß das erste ausgegebene Zeichen ungültig ist. Daher sollte man, auch um einen definierten Start zu haben zunächst ein paar Synchronisationsbytes ausgeben.

Beispiel:

```
SYNCAUS:=$
    21 *10.W     ; erst man 10 STück FF
    CD SCHLEIFE
    0E FF        ; Zeichen laden
    CD PO        ; aufzeichnen
    CD ENDSCHLEIFE ; und dann
    0E 00        ; z.B. ein Erkennungsbyte
    CD PO        ; danach kann man Daten
                   aufzeichnen
    C9           ; hier aber Ende
```

CLR

Alle vier Bildseiten der GDP-Baugruppe werden gelöscht. Die Löschung erfolgt in ca. 80 Millisekunden.

Beispiel:

```
CD CLR       ; löschen
C9           ; Ende
```

CLPG

Wenn man eine unsichtbare Seite löschen will, so muß man diesen Befehl anwenden. Er ist langsamer als der CLR-Befehl, kann jedoch ohne Bildstörungen zu verursachen eine beliebige Seite löschen. Die Löschseite wird im Seitenregister (Port 60H) angegeben.

Beispiel:

```
CD WAIT      ; erst muß GDP bereit sein
3E 40        ; Seite 1 schreiben, 0 lesen
D3 60        ; und
CD CLPG      ; diese Seite löschen
C9
```

Der Code für das Seitenregister kann man wie folgt berechnen:

schreibseite*64+leseseite*16

Der Wertebereich ist: 0,1,2,3

WAIT

Diesen Befehl hatten wir schon oft verwendet. Nach Aufruf wartet das Programm solange, bis der Graphikprozessor seinen letzten Befehl abgearbeitet hat. Wenn man einmal ein WAIT vergißt, so kann es zu Störungen kommen. Die Schildkrötenbefehle, wie auch MOVETO, DRAWTO, WRITE und READ warten automatisch, so daß man WAIT eigentlich nur dann braucht, wenn man direkt auf die Register des GDPs zugreifen will.

Beispiel:

```
CD WAIT
3E 41        ; Buchstabe "A" ausgeben
D3 70        ; mit aktuellen Parametern
C9
```

RAM

Ein Symbol, daß nicht als Befehl verwendet werden kann. Damit wird die Adresse 8800 angesprochen. Man kann den NAMEN RAM z.B. eingeben, wenn man im Menue AENDERN die erste Speicherzelle verwenden will.

Hier noch ein Hinweis. Der Bereich 8000 bis 81CA ist vom Grundprogramm reserviert und sollte nicht verwendet werden. Der Bereich 81CA bis 8FFF ist eigentlich frei, jedoch werden beginnden bei 81CA die Symbole abgelegt, die man durch ":=" definiert hat, und daher sollte man erst bei 8800 beginnen Programme abzulegen. Die Adresse 8800 hat sich als guter Kompromiß herausgestellt. Wenn man nicht sehr viele Symbole verwendet, so kann man auch bei 8400 oder gar bei 8200 beginnen.

* Rolf-Dieter Klein (RDK)

Bausätze und Geräte zum Buch und zur Schulfernsehreihe

III Mikroelektronik

Preisliste Baugruppen

Typ	Funktion	P	Einzelpreise inkl. MwSt. LP	BS	FB
POW5V	Spannungsversorgung 5 V/3 A	M,1,68	15,–	39,95	58,50
SBC2	Z80A-CPU, 4 K RAM, EPROM-Sockel	M,1	15,–	79,95	129,–
BUS2I	Grundpl. 6 Plätze, 4 teilbest.	1	15,–	39,95	58,50
BUS2II	Grundpl. 6 Plätze, vollbestückt	–	15,–	69,90	89,–
BU2III	Grundpl. 12 Plätze, vollbestückt	68	30,–	137,80	169,–
BU2IV	Grundpl. 18 Plätze, vollbestückt	–	45,–	206,70	249,–
MINIBUS	Grundpl. für POW5V, SBC2, IOE	M	–	19,50	24,50
IOE	16-Bit-Ein-, 16-Bit-Ausgabe	1	15,–	39,95	69,–
IOE-EX	IOE f. Exp. Musik, Ampel, Robot.	M	15,–	89,90	145,–
GDP64K	Vollgrafik m. 64-KByte-Bildsp.	1,68	15,–	359,–	458,–
KEY	Tastaturanschluß	1,68	15,–	49,95	89,–
TAST	DIN-Tastatur o. Funktionstasten	1,68	–	–	198,–
TAST/G	Gehäuse zur kleinen Tastatur	68	–	–	49,90
DINTAST	Große Tast. m. Funktionstasten	–	–	–	410,40
GEH/DT	Gehäuse zur großen Tastatur	–	–	–	112,86
TAST/K	Rundkabel für Anschluß an KEY	68	–	–	12,50
CAS	Kassettenrecorder-Anschluß	1,68	15,–	74,90	129,–
CPU68K	16-Bit-Prozessor 68008	2,68	15,–	199,–	265,–
ROA64	Speicherkarte für 8 K × 8 RAM/EPROM	2,68	15,–	39,95	84,–
DRAM64	128 K dyn. RAM-Speicher, 64-K-bestückt	–	15,–	398,–	479,–
DRAM128	128 K dyn. RAM-Speicher, vollbestückt	68	15,–	599,–	699,–
PROMER	EPROM-Programmierzusatz	2	15,–	79,95	129,–
POW21/26	Spannungswandler für PROMER	–	15,–	58,60	76,–

Paketpreise

PAK-M	Minipaket für Experimente	219,90	333,–
PAK-1	Z80-Paket	849,–	1145,–
PAK-2	68008-Aufbaupaket	595,–	749,–
PAK-3	PAK-1 und PAK-2 zusammen	1435,–	1885,–
PAK-68	68008-Paket	1950,–	2499,–

P = Paketzugehörigkeit, LP = Leiterplatte, BS = Bausatz, FB = Fertigbaugruppe

Software und Begleitmaterial

Typ	Funktion	P	Einzelpreise inkl. MwSt. LP	BS	FB
MON1	Monitorprogramm für SBC2	1	–	–	60,–
MUO	Musik-Testprogr. SBC2 o. RAM	–	–	–	30,–
MUM	Musik-Testprogr. SBC2 m. RAM	–	–	–	30,–
AMPEL	Programm für Experiment Ampel	–	–	–	30,–
ROBOT	Programm für Experiment Roboter	–	–	–	30,–
BASIC	BASIC für SBC2, mit Handbuch	–	–	–	75,–
GOSI	Grafiksprache f. SBC2, m. Handb.	–	–	–	75,–
MON68K	68008-Monitor/-Editor/-Assembler	2,68	–	–	155,–
PASCAL	68008-PASCAL/S	68	–	–	155,–

Begleitmaterial

BUCH	R.-D. Klein: Mikrocomputer selbstgebaut u. programm.	38,–
VCAS	2 Videokassetten mit der Sendereihe (VHS, Beta, V2000)	248,–
MC-SCH	mc-Begleitheft Schaltpläne und Unterlagen	8,–
MC-Z80G	mc-Begleitheft Z80-Grundprogramme	12,–
MC-Z80A	mc-Begleitheft Z80-Aufbauprogramme	12,–
MC-68G	mc-Begleitheft 68008-Grundprogramme	12,–
MC-68A	mc-Begleitheft 68008-Aufbauprogramme	12,–
MC-PAS	mc-Begleitheft PASCAL/S-Quellprogrammliste	12,–

Zubehör

ZVM123	Bildschirm-Monitor 12", grün	293,–
ZEUG	Werkzeugsatz mit Lötkolben, Zangen, Draht usw.	69,95
BU18	Buchsenleiste 18polig für BUS2	3,80
ROBOT	Fischertechnik-Roboterbausatz	194,50
EXMUS	Bauteilsatz Experiment Musik, inkl. Eprom MUO o. MUM	41,20
EXAMP	Bauteilsatz Experiment Ampel, inkl. Eprom	34,50
EXROB	Bauteilsatz Experiment Roboter, inkl. Eprom	64,80
NETZ	Großes Schaltnetzteil 5 V/6 A, 12 V/3 A, –5 u. –12 V/0, 5 A	198,–

Alle Preise einschließlich Mehrwertsteuer

ELEKTRONIKLADEN 4930 DETMOLD 18 ☎ 05232/8171

Verkaufsstelle München: Schulstr. 28, 8000 München 19, Tel. 0 89/1 67 94 99

NEUE BÜCHER ZUR SENDEREIHE
MIKROELEKTRONIK
UND ZUM NDR-COMPUTER

Die sichere Basis zum Verstehen, Aufbauen, Programmieren und Anwenden des NDR-Computers

Mikroelektronik und Mikrocomputer
von Heiner Jaap

Band 1, Kapitel 1-9
Aufbau und Arbeitsweise des NDR-Computers
nur 19.80 DM, 170 S., brosch., ab Ende Dezember '84 im Buchhandel

Band 2, Kapitel 10-16
Programmieren des NDR-Computers
nur 19.80 DM, 160 S., brosch., ab Januar '85 im Buchhandel

Band 3, Kapitel 17-26
Der NDR-Computer als 16-Bit System
nur 19.80 DM, 190 S., brosch., ab Februar '85 im Buchhandel

Jedes Kapitel ist für die Themen der Folgen der Fernsehreihe 'maßgeschneidert'. Damit kann der Neuling die einzelnen Sendungen nachbereiten und sein Wissen vertiefen. Dem technisch Versierten wird eine Fülle weiterführender Informationen geboten. Marginalien am Seitenrand erleichtern die Übersicht.

Speziell für Anwender des Z-80 bzw. 68008-Systems werden im Februar bzw. März '84 zwei HANDBÜCHER erscheinen. Von Peter Pook-Haffmans/Kurt Burghardt, ~400 S.

perComp Verlag GmbH Holzmühlenstr. 88 2000 Hamburg 70 Tel.: 040/693 20 33

fischertechnik computing
Aus der erfolgreichen Schulfernseh-Serie „Einführung in die Mikro-Elektronik".

fischertechnik computing. Roboter, Automaten und Graphikgeräte zum Selberbauen.

Aus dem fischertechnik computing Baukasten können 10 verschiedene Modelle gebaut werden. Von der einfachen Ampelsteuerung bis zum 2-achsigen Roboter.

Ausführliche Informationen von:
GES, Graf Elektronik Systeme GmbH, Magnusstraße 13
D-8960 Kempten

fischertechnik
Technik. Mit Zukunft.

Tips und Tricks

Speichererweiterung für die SBC 2 – Baugruppe

Dr. Hans Hehl

Bei der SBC-2-Baugruppe stehen als Datenspeicher zwei statische Speicherbausteine HM 6116 als RAM (Random Access Memory) mit einem Speicherplatz von 4 KByte zur Verfügung. Diese belegen die Adressen 8000H – 87FFH (IC 8) und 8800H – 8FFFH (IC 9, siehe Sonderheft Nr. 88, MIKROCOMPUTER, Franzis-Verlag). Nach dem Start des BASIC-Interpreters ergibt der Befehl PRINT FRE(0) die Größe des zur Verfügung stehenden Speicherplatzes von 1784 Bytes. Bei längeren BASIC-Programmen reicht bald dieser Platz nicht mehr aus und man wünscht sich eine einfache Form der Speichererweiterung.

Eine Möglichkeit ergibt sich durch die Verwendung einer weiteren SBC-2-Platine, auf der nur die statischen Speicherbausteine mit Sockel sowie ein Pufferelko eingesetzt werden. Die Datenleitungen D0-D7, die Adressenleitungen A0-A7 und die Leseleitung –RD (Read Acces) sowie die Schreibleitung – WR (Write Acces) beider Platinen sind über den BUS (Trägerplatine) sowieso miteinander verbunden. Zusätzlich werden noch die drei Adressenleitungen A8-A10 und die Auswahlleitungen (–CS = Chip Select) für die bis zu vier zusätzlichen Speicherbausteine benötigt. Die Verbindung kann entweder mittels eines siebenadrigen Flachkabels oder über weitere BUS-Leitungen vorgenommen werden. Auf der Erweiterungsplatine muß dann nur noch die –WR-Leitung zu IC6 und IC7 geführt werden. Wenn auf der Erweiterungsplatine vier Speicherbausteine HM 6116 eingesetzt sind, ergibt sich ein Speicher bis zur Adresse AFFH. Allerdings ist er nicht bis zum Ende nutzbar, da dort beim BASIC der STACK liegt. Es ergibt sich folgende Adressenzuordnung auf der Erweiterungsplatine, wobei die IC-Bezeichnungen IC 6 bis IC 9 beibehalten wurden.

IC 6	9000H	–	97FFH
IC 7	9800H	–	9FFFH
IC 8	A000H	–	A7FFH
IC 9	A800H	–	AFFFH

Vor dem Löten sollte man unbedingt die CMOS-Bausteine aus den Sockeln nehmen und auf einen leitenden, schwarzen Schaumstoff stecken. Im Einzelnen müssen nun folgende Verbindungen zwischen der SBC-2-Baugruppe und der Erweiterungsplatine geschaffen werden:

SBC 2 ⟷ Erweiterung					
IC		Pin	IC		Pin
IC5	74LS138	13	IC6	HM6116	18
IC5	74LS138	12	IC7	HM6116	18
IC5	74LS138	11	IC8	HM6116	18
IC5	74LS138	10	IC9	HM6116	18
IC9	HM6116	23	IC9	HM6116	23
IC9	HM6116	22	IC9	HM6116	22
IC9	HM6116	19	IC9	HM6116	19
Erweiterung ⟷ Erweiterung					
IC7	HM6116	21	IC8	HM6116	21

Besser als ein Flachbandkabel zwischen beiden Platinen ist jedoch die Verdrahtung ausschließlich über die BUS-Leitungen. Auf jeder SBC-2-Platine kann nämlich noch ein gewinkelter Einzelstift am rechten Rand der Steckerleiste (Blick auf die Bestückungsseite der Platine) eingebaut werden. Es muß neben der Steckerleiste mit gleichem Abstand wie die anderen Stifte ein Loch mit 1 mm Durchmesser gebohrt und der Einzelstift mit einem Schnellkleber auf der Bestückungsseite in der richtigen Lage fixiert werden, da bei neueren Platinen die Kupferfläche zum Einlöten des zusätzlichen Stiftes nicht mehr vorhanden ist. Da die BUS-Leitungen für 12 Volt in der Grundausbaustufe nicht benötigt werden, stehen somit am linken Steckerteil drei und am rechten Steckerteil vier Stifte bzw. BUS-Leitungen zur Verfügung. Allerdings müssen auf der BUS-Platine die Buchsenleisten vollständig bestückt sein, also pro Reihe drei Buchsen enthalten. Provisorisch kann auch für den zusätzlichen Stift jeder SBC-2-Platine eine Einzelbuchse eingelötet werden. Nun ergibt sich eine kurze Leitungsführung auf jeder Platine und man kann die Erweiterungsplatine leicht entnehmen.

Wie nun die Verbindungen zu legen sind, zeigen die Bilder 1 und 2. Die zusätzlichen Stifte wurden mit den Ziffern 0 – 3 und 4 – 6 gekennzeichnet. Auf jeder Platine müssen die mit den gleichen Ziffern gekennzeichneten Lötaugen miteinander mittels dünnen, isolierten Schaltdraht verbunden werden.

Vor dem Einschalten muß unbedingt nochmal die korrekte Lage der IC's und die Verdrahtung der Speichererweiterung auf beiden SBC-2-Platinen gewissenhaft kontrolliert werden. Nach dem Einschalten meldet sich der Interpreter dann wie gewohnt. Der Befehl PRINT FRE(0) ergibt nun einen zur Verfügung stehenden Speicherbereich von 9976 Bytes, da 8192 (4*2048) Bytes Speicherplatz hinzugekommen sind.

Die Speichererweiterung ist auch beim Grundprogramm verwendbar.

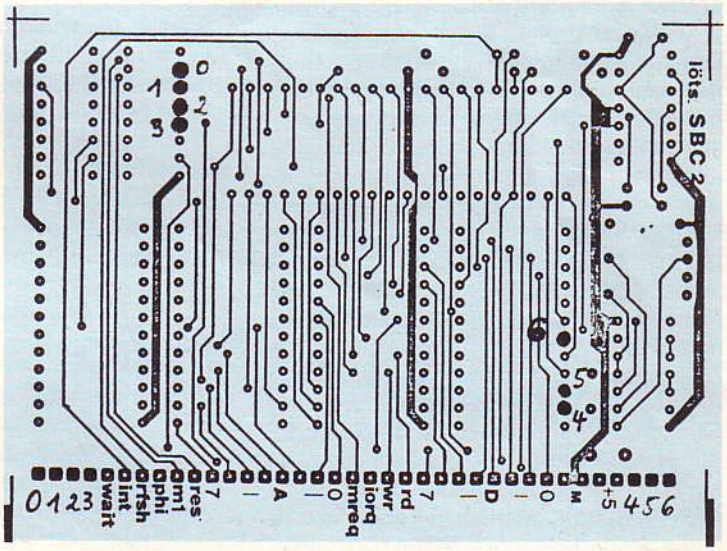


Bild 1: Lötseite der SBC-2-Baugruppe

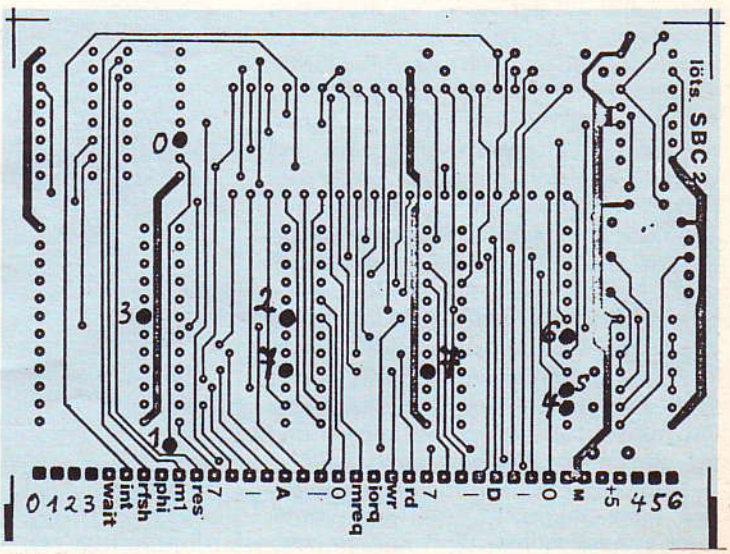


Bild 2: Lötseite der Erweiterungsplatine

Grundlagen

Löten in der Elektronik

Wolfgang Meier

Welcher Lötzinntyp ist denn nun am besten geeignet?

Im Elektronik-Handel gibt es ja viele Lötzinntypen, wer weiß schon welchen er auswählen soll. Nun, es ist schon viel über Löten geschrieben worden, aber welcher Lötzinntyp der bessere ist und welchen Typ man auf keinen Fall nehmen sollte, darüber schrieb bisher noch niemand. Deshalb soll hier das Thema Lötzinn angesprochen werden. Denn für den Hobby-Elektroniker und den NDR-Klein-Computerbauer ist es wichtig zu wissen, daß die sogenannten „Radiolote" halogenhaltiges Flußmittel enthalten. Diese sind für uns in der Elektronik nicht geeignet. Es gibt einen viel besseren Lötzinn-Typ, wie wir noch sehen werden.

Wir Techniker wissen alle, daß heute im Zeitalter der Raumfahrt und der Computertechnik, Löten nicht mehr Löten und Lötzinn nicht mehr Lötzinn ist. Aber warum? Das ist bedingt durch neue Technologien, wie Printplatten mit eben sehr dünnen Leiterbahnen aus Kupfer und den hohen Anforderungen, wie sie in den einzelnen Elektronikbereichen, z.B. in Datenanlagen, Digitaltechnik, Nachrichtenwesen sowie im Computerbau gefordert werden. Vorallem aber in der Raumfahrttechnik, hier muß man ganz besonders „kalte" Lötstellen sowie ein späteres korrodieren der Lötstellen und der Kupferleiterbahnen unbedingt vermeiden.

Also nochmal: Hände weg in der Elektronik von Lötdrähten mit halogenhaltigen Flußmitteln. Deren Halogene sind hier die aggressiven Stoffe, die auf den Printplatten die erwähnten Schäden bewirken.

Man verwendet deshalb in der Elektronik einen besonders entwickelten Lötzinntyp, bei dem das Zinn (Sn) mit Kupfer (Cu) angereichert ist und mit einem Flußmittel (hier F-SW 32) als Seele, welches völlig halogenfrei ist und auf der Basis von reinem Kolophonium (WW), daß mit Aktivierungsstoffen angereichert wurde.

Aus genannten Gründen hat man schon vor Jahren eine sehr aufwendige und lange Untersuchungsreihe, unter der Federführung des Fernmeldetechnischen Zentralamtes (FTZ) der Deutschen Bundespost in Zusammenarbeit mit führenden Lötdraht-Herstellern (Bleiwerke) durchgeführt. Hieraus ergab sich für die genannten Elektronikbereiche Röhrenlötzinn aus Zinnlegierungen mit Kupfer (Cu), auch welche mit Silber (Ag) und Phosphorzusatz (P) nach Gruppe B und mit einer Flußmittelseele nach F-SW 3 (= Gruppe der nicht korrodierend wirkenden Rückstände), welches aus reinem Kolophonium (WW), mit Aktivierungsstoffen angereichet, besteht.

Dieses Flußmittel F-SW 3 ist nach C-Gruppen fein unterteilt, je nach %-Gehalt der Aktivierungsstoffe.

Die Unterteilung der F-SW 3 Gruppe sieht so aus:

Typ F-SW 31-K (k = Kurzzeichen) = nur reines Kolophonium WW
Typ F-SW 32-C Hergestellt aus reinem
Typ F-SW 32-C2 Kolophonium WW mit
Typ F-SW 32-C3 Zusätzen von organischen, halogenfreien Aktivierungsstoffen
Typ F-SW 32-C4 ist ein Sonder-Kolophonium-Flußmittel

Die Herstellung der Röhrenlötdrähte ist in den DIN-Vorschriften DIN 1707, 8511 und 8516 verbindlich festgelegt worden.

Die angeführte Untersuchungsreihe hatte nun ergeben, daß der Lötzinntyp „L-Sn 60 Pb Cu2 mit Flußmittel F-SW 32 nach Typ C3, also mit 2 % Kupfergehalt sowie mit mehr%tigem Aktivierungsstoffgehalt, sich besonders und hervorragend für die erwähnten Elektronikbereiche eignet. Es wird von Anlagenherstellern in bestimmten Fertigungsbereichen vorgeschrieben und in der Raumfahrttechnik ist es sogar zwingend vorgeschrieben; das haben auf unsere Anfrage Bleiwerke, bestätigt. Übrigens: Bei der NASA wird ein Lötzinn nach gleichen Kriterien vorgeschrieben. (Der Hamburger NDR-Klein-Computerclub der Hamburger Handelskammer hat sich solches amerikanisches Lötzinn beschafft, die wußten also schon warum!). Genau dieser Lötzinn-Typ ist also gerade für den NDR-Klein-Computerbau das richtige Lötzinn.

Dieses Röhrenlötzinn hat folgende Bezeichnung:

L-Sn 60 Pb Cu 2 – Typ C 3 – 2,5; DIN 8516.

Die Bedeutung:

1. L-Sn60PbCu2
   = 60 % Sn, 38 % Pb, 2 % Cu,
   L = Weichlot
2. F-SW 32
   = F steht für Flußmittel
   = S steht für Schwermetall
   = W steht für Weichlöten (bis 450 °C)
   3 ist die Gruppe der nicht korrodierend wirkenden Rückstände
   2 gilt für Anwendung in der Elektronik, Leiterplatten (1 Anwendung für normale Elektrotechnik und Elektronik).
3. Unterteilung des Flußmittels F-SW32 nach C-Gruppen:
   Typen C / C 2 / C 3
   Hergestellt aus reinem natürlichem Kolophonium WW mit Zusätzen von organischen, halogenfreien Aktivierungsstoffen (%-Zusätze nach C-Gruppe).
4. 2,5 ist der %-Anteil des Flußmittels im Lötdraht.
5. DIN 8516 bedeutet: nach DIN-Vorschrift 8516 hergestellt.

Nach DIN 8516 – 4,7 Kennzeichnung: Jede Lötdrahtspule muß die oben angeführte Kennzeichnung neben dem Zeichen des Herstellers, also die Kurzbezeichnung enthalten, wie z.B.:

L-Sn 60 Pb Cu2 – F-SW 32 DIN 8516 – 2,5

sowie Angaben über die Maße, z.B. 1 mm Ø und den Flußmittelgehalt hier: 2,5 %.

Dieser angegebene Lötzinntyp ist für Lötungen auf den Leiterplatten und für den Computer-Selbstbau also genau das richtige Lötzinn. Das Beste ist gerade gut genug und so teuer ist es nun auch wieder nicht.

Wer mehr über Löten und Lötzinn wissen möchte, sollte sich mal vom Bleiwerk Goslar das Faltblatt über ELSOD-Feinlote schicken lassen, bzw. das Buch „Weichlöten in der Elektronik", vom Verlag Schiele und Schön GmbH in 1 Berlin 61, Markgrafstraße 11. Hierin sind Korrosionen und Zersetzungen an Lötstellen und Leiterbahnen durch Fotos belegt.

Also nochmal: In der Computertechnik keine „sogenannten" Radiolote verwenden, denn diese haben ein halogenhaltiges Flußmittel, wie z.B. F-SW 26.

Es sei hier einmal erlaubt zu fragen: Warum führen eigentlich die meisten Elektronik-Händler sowie die Computer-Bausatz-Lieferer dieses Lötzinn nicht?

Wenn gerade letztere ihren Kunden zu den Bausätzen dieses Lötzinn anbieten würden, hätten diese keine Beschaffungsprobleme mehr, das geeignete Lötzinn zu erhalten. Besonders unsere Schüler und Einsteiger in die Elektronik wissen sehr oft nicht, welches Zinn muß ich nehmen.

WM

Gründung des Mikromodul e.V.

Die Erfahrungen der Handelskammer Hamburg bei der Betreuung von aktiven Teilnehmern der NDR-Sendereihe "Mikroelektronik" durch Maßnahmen wie Telefonberatungsdienst oder Organisation von Arbeits- und Erfahrungsgruppen führte im August 1984 zur Gründung des **Mikromodul e.V.** mit dem Ziel der Unterstützung von Aus- und Weiterbildung im Bereich innovativer Technologien. Diese Aktivitäten sind nicht regional beschränkt, sondern überregional orientiert. Selbstverständlich sollen dadurch keinesfalls bereits aktive Gruppen behindert oder gestoppt werden. Vielmehr wird diesen eine zusätzliche Unterstützung angeboten.

Zu den Aktivitäten des Mikromodul e.V. gehören beispielsweise:

- Gründung überregionaler **Benutzergruppen** und regionaler **Clubs**
- Organisation des **Erfahrungsaustauschs** von Arbeitsgruppen für die NDR-Sendereihe "Mikroelektronik" und Koordination des **Telefonberatungsdienstes** der regionalen Industrie- und Handelskammern
- Herausgabe eines **Mitteilungsblattes** (Beantwortung von Fragen, Besprechung von Publikationen, Programmen, Hinweise auf Fehler in Hardware und Software, Test- und Erfahrungsberichte usw.)
- Förderung der Entwicklung von didaktisch gutem **Begleitmaterial** für Sendereihen wie "Mikroelektronik" des NDR
- Aufbau und administrative und-fachliche Betreuung des elektronischen **Mikromodul-Auskunftssystems**

Der Aufbau des Mitteilungsblattes der Benutzergruppe NDR-Computer (erstes Heft im Januar 1985) und des elektronischen Mikromodul-Auskunftsystemes sind die ersten Aktivitäten des Mikromodul e.V., die bereits initialisiert wurden.

Während der Gründungsversammlung des Mikromodul e.V. in der Handelskammer Hamburg am 1. August 1984 wurde als Vorstand Herr Dr. Otmar Zembold, Handelskammer Hamburg, Herr Joachim Arendt, NDR Hamburg, Herr Dr. Günter Flöss, Wolfsburg und Herr Günter Mußtopf, PERCOMP-Verlag Hamburg, einstimmig gewählt.

Der Mitgliedsbeitrag beträgt für persönliche Mitglieder DM 30,- pro Jahr. Die Beiträge werden für die Verwaltung und den Druck und Versand des Mitteilungsblatten verwendet. Fragen bezüglich einer Mitgliedschaft oder Aktivitäten des Mikromodul e.V. richten Sie bitte an:

Dr. Otmar Zembold
Handelskammer Hamburg
Börse
2000 Hamburg 11

Mikromodul Auskunftssystem

Die Diskussionen im Mikromodul e.V. (siehe Beitrag Gründung des Mikromodul e.V.) über die Organisation des Erfahrungsaustauschs zwischen Benutzern des NDR-Computers führten zur Idee, das **Mikromodul-Auskunftssystem** aufzubauen. Sein oberstes Ziel ist eine **überregionale, flächendeckende Förderung der Aus- und Weiterbildung** im Bereich der Mikroelektronik und der Mikroprozessortechnik und deren Anwendungen in Beruf und Hobby. Dieser Service soll **unabhängig vom Wohnort** und der jeweiligen Tages- oder Nachtzeit wirksam werden. Nicht zuletzt stellt das Mikromodul-Auskunftssystem ein sehr gutes Beispiel für die Anwendung des NDR-Computers und des DATEX-P Netzwerkes auf der einen und für eine sinnvolle Nutzung innovativer Informations- und Kommunikationstechniken auf der anderen Seite dar.

Selbstverständlich bietet das Mikromodul-Auskunftssystem ein breites Spektrum von Themen an, die von Fachredakteuren betreut werden:

- Informationen über Sendereihen in Funk und Fernsehen und Sendetermine; Veranstaltungskalender (Tagungen, Kurse, Seminare usw.)
- Hardware (neue Produkte, Schaltungsvorschläge usw.)
- Fehler in Hardware, Software und Veröffentlichungen
- Besprechung von Büchern und Hinweise auf Publikationen in Fachzeitschriften
- Tagungsberichte, Test- und Erfahrungsberichte
- Programmbibliothek (Quell- und Objektprogramme)
- Schwarzes Brett (Fragen, Suche ..., Biete ..., Kontaktecke usw.)

Selbstverständlich muß auch der Benutzer einer solchen Dienstleistung etwas investieren, bevor er den Service des Mikromodul-Auskunftssystems in Anspruch nehmen kann. Im Fall des NDR-Computers benötigt er eine zusätzliche Platine mit einer seriellen Schnittstelle (V.24-Interface), die auch die erforderliche Kommunikationssoftware enthält. Diese Platine wird einschließlich der Software (EPROMs) bereits im Frühjahr 1985 lieferbar sein. Zusätzlich ist ein Akustikkoppler (Übertragungsgeschwindigkeiten von 300 Baud oder 1200/75 Baud) für die Verbindung zum Mikromodul-Auskunftssystem über eine Telefonleitung erforderlich.

Zur Zeit wird eine ausführliche Information über das Mikromodul-Auskunftssystem vorbereitet, die ab Januar 1985 verfügbar sein wird. Fragen richten Sie bitte an:

Frau A. Koch oder Herrn Günter Mußtopf
Handelskammer Hamburg PERCOMP-Verlag GmbH
Börse Holzmühlenstr. 88
2000 Hamburg 11 2000 Hamburg 70

Der Line by Line Assembler für den Z 80 ist da!

Gerd Graf

Beim Arbeiten mit dem Grundprogramm stösst man bei etwas längeren Programmen bald an die Grenze der Programmierung in Maschinensprache. Auch kann man, speziell beim Z80, sich nicht alle Codes in Maschinensprache leicht merken .

Abhilfe bietet hier die Programmierung in Assembler. Der Assemblercode besteht aus Befehlsabkürzungen, die man sich leicht merken kann, z.B. "LD" für "Lade", "call" für den Aufruf eines Unterprogrammes und vieles mehr.

Auch muss man sich beim Programmieren in Assembler nicht mehr um physikalische Adressen kümmern, man kann "symbolisch" programmieren. Programmteile werden mit Adressen versehen, auf die man Bezug nehmen kann. Beispiel: Zur Textausgabe legt man meist einen Speicherbereich an, in dem der auszugebende Text steht. In Maschinensprache muss man nun genau die Adresse, ab der der Speicherbereich beginnt, angeben (z.B. 8900H). Wird nun das Programm länger als geplant, ist diese Adresse vielleicht schon vom Programm belegt, man muss sie ändern und damit auch alle Befehle, die sich auf diese Adresse beziehen. Meist vergisst man dann mindestens einen Befehl und die Fehler sind mit einprogrammiert.

Auch die Festlegung von Sprungadressen ist nicht sehr leicht. Springt man nach vorne, weiss man ja in der Maschinensprache noch nicht, auf welche Adresse der Befehl, den man anspringen will, zu liegen kommt. Ändert sich diese Adresse z.B nach dem Einfügen von Befehlen, wirds noch schwieriger, weil man dann wieder alle Sprungbefehle ändern muss.

Bei der Programmierung in Assembler werden Adressen nun als Namen angegeben. Man schreibt einfach vor einen Befehl oder ein Unterprogramm einen Namen wie folgt:

```
Schleife:   ld a,5
            .....        ; weitere Befehle
```

Ein "richtiger" Assembler, wie man ihn mit Diskettenlaufwerken verwendet, ist ein sogenannter "Zwei-Pass-Assembler", der zwei Durchläufe (Passes) durch das gespeicherte Programm braucht.

Beim ersten Durchlauf werden alle Befehle übersetzt und unter anderem alle symbolischen Namen in eine Tabelle, die sogenannte Querverweisliste, eingetragen. Die echten Adressen kann der Assembler ja noch nicht wissen - die Adressteile der Befehle werden zunächst durch leere Bytes freigehalten.

Beim zweiten Durchlauf schaut sich der Assembler nun alle solche Sprung-und Callbefehle an und sucht in seiner Querverweisliste, die er beim ersten Mal erstellt hat, die echte physikalische Adresse und trägt sie ein. Findet er keine, gibts eine Fehlermeldung.

Für einen solchen Assembler benötigt man einen grossen Massenspeicher, wie eine Diskette oder eine Festplatte. Unter dem Betriebssystem CP/M gibt es bewährte Assemblerprogramme, die schon im Preis des Betriebssystems enthalten sind.

Zum Üben und für einfachere Zwecke genügt ein "Ein-Pass-Assembler", der nur einen Durchlauf macht. Diesen Assebler kann man auch in Eproms liefern, da er ja nichts zwischenspeichern muss.

Rolf-Dieter Klein hat einen solchen Assembler geschrieben, er passt in ein 8 KByte Eprom und ist, komplett mit Dokumentation, ab Januar lieferbar.

Der Einweg-Assembler kann also nur auf Namen, die vorher definiert wurden, Bezug nehmen - das sind Rückwärtsreferenzen und Namen, die im Z80-Grundprogramm als Namen enthalten sind, wie "SCHREITE", "DREHE" usw.

```
8800             ; kleines Beispiel
8800 210400      ld hl,4
8803 CD0F20      call schleife
8806 216400       ld hl,100
8809 CD0320       call schreite
880C 215A00       ld hl,90
880F CD0620       call drehe
8812 CD1220      call endschleife
8815 C9          ret
8816             end _


8816             ; Beispiel 2
8816             start:
8816 0605         ld b,5
8818             schleife:
8818 C5           push bc
8819 216400       ld hl,100
881C CD0320       call schreite
881F 214800       ld hl,72
8822 CD0620       call drehe
8825 C1           pop bc
8826 10F0         djnz schleife
8828 C9           ret
8829             end_
```

BILD1 zeigt zwei Beispiele, direkt vom Bildschirm des NDR-Computers mit eingesetztem Assembler aufgenommen. In diesem ersten Beispiel steht rechts der eingegebene Assemblercode, ganz links die Spalte

mit der Adresse und dazwischen der vom Assembler erzeugte Maschinencode. In diesem ersten Beispiel sind nur symbolische Namen des Grundprogrammes enthalten. In zweiten Beispiel, sind auch selbst erstellte Namen, wie SCHLEIFE und START enthalten.

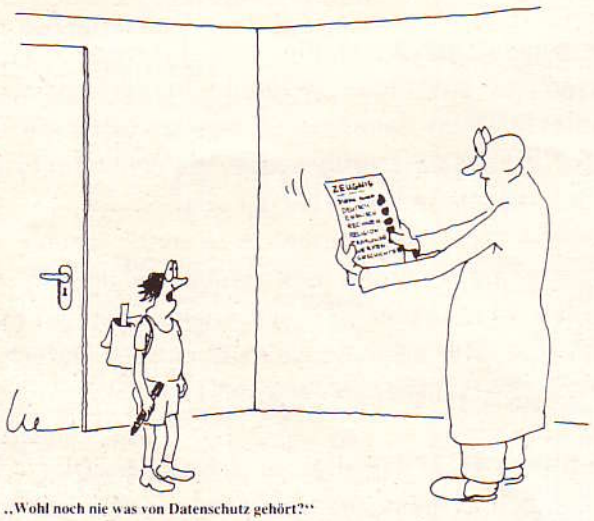
Somit kann der auf Adresse 8826 übersetzte Befehl "DJNZ SCHLEIFE" vom Assembler korrekt ausgeführt werden, da Schleife ja auf Adresse 8818 definiert wurde.

Der Assembler bezieht sich auf Routinen des Grundprogrammes und kann nur parallel mit diesem in Betrieb genommen werden. Dazu ist also die Konfiguration "Vollausbau - CPU Z80" sowie eine ROA64-Karte nötig. Auf der ROA64 sitzt ab Adresse 0 das Grundprogramm, auf Adresse 6000H der Assembler.

Zum Assembler wird noch ein komfortabler Disassembler erstellt, das ist ein Programm, das aus Maschinencode wieder, soweit wie möglich, Assemblercode erzeugt. Natürlich kann ein Disassembler symbolische Adressen nicht wieder rekonstruieren, sie sind ja im Code nicht enthalten.

Mit dem Eprom Assembler und Disassembler ist der NDR-Computer speziell im Z80-Bereich wesentlich stärker geworden, ein Kritikpunkt , den wir nun aufgehoben haben. Das Programm "Assembler" ist besonders zur Ausbildung und zum Lernen der Assemblerprogrammierung hervorragend geeignet.

aus computer colleg 1/85



„Wohl noch nie was von Datenschutz gehört?"

SBC 2 oder Vollausbau - CPUZ 80 ?

Gerd Graf

Der NDR-Computer in der Z80-Version muss , um funktionieren zu können, über eine Zentraleinheit (den Z80), Programmspeicher in Eproms und Arbeitsspeicher in RAMs verfügen.

Diese drei Elemente sind auf der Single Board Computer Baugruppe, der SBC2 vereinigt. Dies ist der grösste Vorteil der SBC2. Sie hat aber auch einige Nachteile.

Der erste Nachteil ist der, dass die Speicherbereiche nicht erweiterbar sind. Um z.B. eine Programmiersprache zu wechseln, muss man die EPROMs ziehen und andere Eproms stecken, ein Verfahren, das eigentlich einem so modernen Computer wie dem NDR-Computer unwürdig ist. Leider ist auch der RAM-Speicherberich von 4 KByte nur über eine weitere SBC2 zu erweitern - dieser Nachteil tritt besonders bei der Programmiersprache BASIC stark hervor.

Der Vorteil der SBC2 liegt im günstigen Preis und in der Möglichkeit, komplette Anwendungen mit einer Baugruppe zu realisieren.

Die Alternative zur SBC2 bietet die Vollausbau-CPUZ80 zusammen mit einer ROA64-Baugruppe. Diese Konfiguration scheiterte bisher am hohen Preis der 8KByte RAM-Bausteine, von denen ja mindestens eins auf der ROA64 gesteckt werden muss. Mit den heutigen Preisen sieht es schon viel besser aus. Man kann nun folgende Programme auf eine ROA64 stecken:

Adresse	Einbauplatz	Programm	Bestellnr.(GES)
0000	J1	Grundprogramm	EGRUND2
2000	J2	GOSI	EGOSI2
4000	J3	BASIC	EBASIC2
6000	J4	ASSEMBLER	EZASS2
8000	J5	RAM-Baustein	R8
A000	J6	.	
C000	J7	.	
E000	J8	weitere RAM-Bausteine	

Vom Grundprogramm kann man nun durch Programmstart auf 2000 GOSI, auf 4000 BASIC usw. aufrufen, ohne Eproms umstecken zu müssen. Natürlich können auch BASIC-Programme auf Unterprogramme des Grundprogrammes zugreifen.

Der weitere Vorteil, schon mit der Vollausbau-CPU zu beginnen, liegt darin, dass das System dann nahtlos zum CP/M80-System zu erweitern ist. Hier wird nur noch die BANK-BOOT-Karte eingesetzt, und eine RAM-Karte mit mindestens 64 KByte, z.B. die RAM64/256 von GES oder die DYN-Karte vom Elektronikladen.

Die Anwender, die schon über eine SBC2 verfügen, müssen nicht verzweifeln: Sie bestellen sich einfach die Platine CPUZ80 und die entsprechenden Sockel und die fehlenden Bausteine. Dann kann die CPU umgesteckt werden, die RAM-Bausteine können auf der BANK-BOOT für CP/M eingesetzt werden. Die SBC2-Baugruppe kann immer wieder Verwendung für eine abgeschlossene Aufgabe finden.

Blick in die Versandabteilung bei GES

INDUSTRIE- UND HANDELSKAMMER ZU KIEL
2300 Kiel 1 · Lorentzendamm 24
Postanschrift: 2300 Kiel, Postfach 26 40
Tel. 04 31/59 04-1 oder 59 04 - 243

Bankkonten: Deutsche Bank Aktiengesellschaft Filiale Kiel (BLZ 210 700 20) Kto.-Nr. 34 40 [illegible]
Landesbank Schleswig-Holstein Girozentrale Kiel (BLZ 210 550 00) Kto.-Nr. 520 [illegible]
Vereins- und Westbank Aktiengesellschaft Filiale Kiel (BLZ 210 300 [illegible]) Kto.-Nr. 30-350 [illegible]
Postscheckkonto: Hamburg (BLZ 200 100 20) Kto.-Nr. 17 20[illegible]

Ohne Anschreiben mit der Bitte um Verständnis für diese Arbeitsvereinfachung

Herrn/Frau/Firma: Reinhard Koslitz, Kaufm. Leiter in der Fa. Graf Elektronik Systeme GmbH, Magnusstr. 13, 8960 Kempten

Auf das Schreiben/den Anruf vom _____ Dortiges Zeichen _____

mit der Bitte um
- X Kenntnisnahme
- X Verbleib
- X weitere Erledigung
- ☐ telefonischen Anruf
- ☐ Rücksprache
- ☐ Mit- oder Gegenzeichnung
- ☐ Rückgabe
- ☐ Weitergabe (Irrläufer)
- ☐ Ablage

Termin/Frist: In der Anlage übersenden wir Ihnen eine Pressemitteilung zum "Computerklub in Kiel" zur Kenntnisnahme.

Anmerkungen:

Mit freundlichen Grüßen

6. November 1984 — gez. Schröder
Datum — Namenszeichen

Anlaß war die Sendereihe "Mikroelektronik-Bauen eines Computers" im 3. Programm des NDR-Fernsehen. Nun wurde in Kiel von der Industrie- und Handelskammer ein Computerklub ins Leben gerufen, der sich weiter mit diesem Thema beschäftigt. 18 angehende Computerbesitzer und -betreiber fanden sich zu ihrem ersten Treffen bei der Firma Siemens in Kiel zusammen, um mit fachlicher Unterstützung und Nutzung der vorhandenen Technik einen 8-Bit-Computer (Z 80) zu bauen, der zu einem komfortablen 16-Bit-Mikrocomputer (68008) ausbaufähig ist.

Dazu Völker Schröder, der Betreuer des Klubs, von der Industrie- und Handelskammer zu Kiel: "Für die IHK war die Gründung und Unterstützung selbstverständlich. Wir sehen darin eine Möglichkeit, die Akzeptanz neuer Technologien und deren Beherrschung zu verbessern. Für die Zukunft ist geplant, diesen Klub mit anderen Computerklubs zusammenzubringen, einen Erfahrungsaustausch zu organisieren und eventuell auch Weiterbildungsmaßnahmen anzubieten".

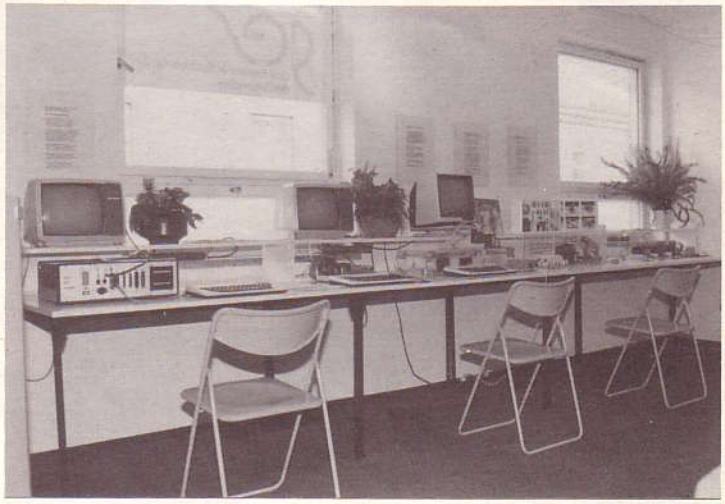
Interessant ist die Tatsache, daß keiner der Teilnehmer eine entsprechende elektrotechnische Vorbildung besitzt, alle also ein für sie völlig neues Gebiet betreten haben.

Sicher ist, daß mit diesem "Hobby" auch ein Schritt zur beruflichen Höherqualifikation der Klubmitglieder geboten wird.

Pressedienst der IHK zu Kiel · 2300 Kiel · Lorentzendamm 24 · Postfach 26 40 · Tel. (04 31) 59 04-1 · Telex 02 99 864 · Fax: (04 31) 5 90 42 34
– verantwortlich: Jürgen Grabowsky –

Mehr zum Club als zur reinen Verkaufsstelle haben sich die Filialen von GES in Hamburg (Ehrenbergstr. 56, 2000 Hamburg 40) und München (Georgenstr. 61 8000 München 40) entwickelt.

An mehreren Arbeitsplätzen sind NDR-Computer in verschiedenen Konfigurationen aufgebaut. Hier kann man die Systeme in Ruhe selbst testen. Ein eigener technischer Kundendienst hilft schnell bei Problemen, die selbst nicht lösbar sind.



Teilansicht unserer Filiale München

Viele Kontakte der Kunden untereinander werden in der Filiale geschlossen. Für NDR-Anwender, die nicht im Einzugsbereich der Grosstädte wohnen, veröffentlichen wir ab sofort (Heft LOOP3) Kleinanzeigen, die der Kontaktaufnahme dienen, kostenlos! Einfach die Adresse und die Schwerpunkte (z.B. Software Z80, 68000-System, Programmieren, Steuerung) an GES, loop-Redaktion, senden!

POST

An dieser Stelle und unter diesem Zeichen erscheint auch Ihr Brief an "loop". Bitte schreiben Sie uns recht bald: Was Sie von "loop" halten, wie Ihnen diese Ausgabe gefällt, was für Wünsche Sie an "loop" haben, was wir besser machen sollten und welche Themen "loop" Ihrer Meinung nach noch behandeln sollte. Wir freuen uns auf Ihre zahlreiche Post.

COUPON
(ausschneiden und absenden)

Ja, ich abonniere **"loop"**, die Zeitung für Computerbauer - 4 Ausgaben im Jahr zu DM 10.- zzgl. DM 4.80 Versandspesen (Scheck liegt bei).

Name.....

Adresse.....

.....

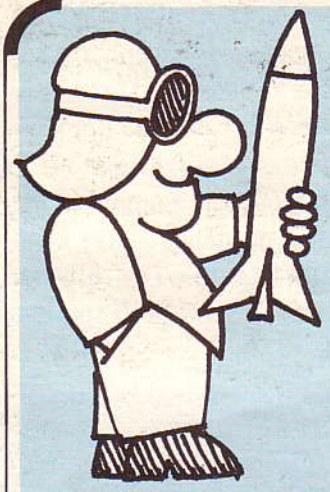
Graf Elektronik Systeme GmbH
Postfach 1610 8960 Kempten

Sie können ein "loop"-Abo bei jeder Bestellung einfach mitbestellen!

IMPRESSUM

loop Zeitung für Computerbauer

Herausgeber : Gerd Graf
Redaktion : Rolf-Dieter Klein, Gerd Graf
Gestaltung:Karl-H.Rieder
Druck: Rieder, Kempten
Herstellung und Anzeigenverwaltung: GES GmbH
Magnusstr.13 8960 Kempten
Anzeigenpreisliste 1/84



Wenn Sie wirklich begreifen wollen, wie ein Computer funktioniert – bauen Sie ihn doch einfach selbst!

GES liefert die beiden interessantesten Selbstbau-Computer:

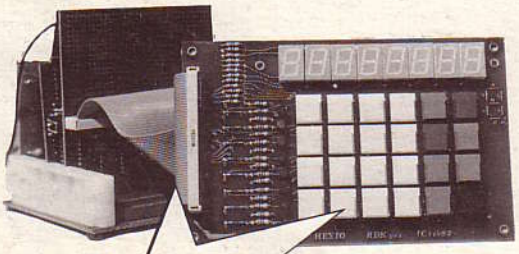
Den NDR-Computer nach Rolf-D. Klein

– aus den III-Programmen des NDR, SFB, BR

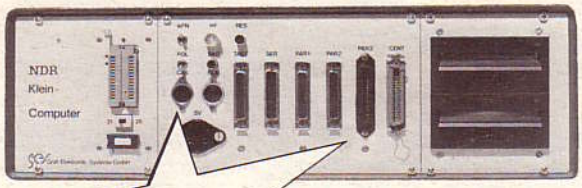
Neu! Jetzt auch mit Floppy-Anschluß und CP/M!

Den mc-CP/M-Computer

– ein Europakartensystem für Fortgeschrittene



Einsteigerpaket: Bausatz DM 295.–



Zum großen System ausbaubar

Der NDR-Computer aus dem Fernsehen – ein Selbstbau-Computer mit unbegrenzten Möglichkeiten!

Steigen Sie klein ein mit dem NDR-Computer – schon für etwa DM 300.– können Sie sich einen funktionsfähigen Computer selbst bauen, der später zum Profi-System in verschiedenen Variationen (z. B. 16 Bit oder CP/M) ausgebaut werden kann.

Lernen Sie mit dem NDR-Computer: Durch den Selbstbau lernen Sie wirklich, wie ein Computer funktioniert. Sie lernen auch bei der Programmierung: Beginnend bei der Maschinenprogrammierung im Einsteigerpaket bis zu allen wichtigen Programmiersprachen beim späteren Ausbau.

Natürlich gibt es auch alle Baugruppen fertig aufgebaut und geprüft.

Sie entscheiden sich für einen Computer, der nie veralten wird! Der NDR-Computer besteht aus kleinen Einheiten, die leicht erweitert oder ausgetauscht werden können. Damit sind Sie immer mit vorne dran!

Der NDR hat sich für unseren Computer entschieden – tun Sie es auch!

Sie investieren in Ihre Zukunft – fordern Sie heute noch unsere ausführliche, kostenlose Info + Probeexemplar unserer Kundenzeitung an (bitte 1.40 Briefmarken für Rückporto beifügen).

Der mc-CP/M-Computer – das Computersystem für Fortgeschrittene, Industrie und Profis!

Der erste Selbstbaucomputer, der das bekannte Betriebssystem CP/M unterstützt, wurde ein Riesenerfolg. Vier Europakarten (ECB-Bus) bilden den weit über 10 000mal nachgebauten mc-CP/M-Computer, der laufend verbessert wird.

Zum CP/M-Computer gehört TERM1, eine preisgünstige, hochauflösende Graphik-Baugruppe, die auch von anderen Systemen über eine serielle (V.24-)Schnittstelle angesteuert werden kann. TERM1 gibt es auch anschlußfertig im Gehäuse mit Netzteil als Graphik-Subsystem GSS.

Preise und alle Einzelheiten entnehmen Sie unserer Info. In unseren Filialen in Hamburg und München können Sie alle unsere Produkte sehen, prüfen und kaufen.

NEU: 8K x 8 RAM R8 DM **70,–**

Graf Elektronik Systeme GmbH, Postfach 1610, 8960 Kempten, Tel. (08 31) 62 11

Filiale Hamburg, Ehrenbergstr. 56, 2000 Hamburg 50 (Altona), Telefon (0 40) 38 81 51
Filiale München, Georgenstr. 61, 8000 München 40 (Schwabing), Telefon (0 89) 2 71 58 58

2800 Bremen, GMCP GmbH, Werftstr. 160, 2800 Bremen 21, Tel. (04 21) 61 30 15
A-1030 Wien, ISYS-Informationssysteme, Landstraßer Hauptstraße 2a, Tel. 75 33 25 CH-4106 Therwil, SYSTECH, Postfach, Tel. (0 61) 73 49 73

ges