

# nascocom journal

**Zeitschrift für Anwender des NASCOM 1 oder NASCOM 2**

3. Jahrgang

Oktober/November 1982

Ausgabe 10/11

**Herausgeber:**

**MK-SYSTEMTECHNIK** Michael Klein · Pater-Mayer-Straße 6 · 6728 Gernersheim/Rhein  
Telefon (0 72 74) 20 93 · Telex 453500 mks d

**MK-SYSTEMTECHNIK** Thomas Gräfenecker · Kriegsstraße 164 · 7500 Karlsruhe · Telefon (07 21) 2 92 43  
**MK-SYSTEMTECHNIK** Michael von Keltz · Pfaffenberg 4 · 5650 Solingen 1 · Telefon (0 21 22) 4 72 67

Der Heftpreis beträgt DM 5,—. Ein Abonnement erhalten Sie für DM 48,— im Jahr. Dafür bekommen Sie 12 Hefte pro Jahr, bzw. 10 Hefte (zwei dicke Doppelausgaben). Die Autoren sind für den Inhalt ihrer Beiträge selbst verantwortlich.

**INHALT**

**ARTIKEL, PROGRAMMBESCHREIBUNGEN, LESERBRIEFE:**

2 NASCOM-Journal Intern	
2 Impressum / Mitarbeiter	
3 TV-Kamera Interface	Günter Böhm
8 Seite(n) für Einsteiger	Otto Föbel
10 Datenausgabe im E- und F-Format	J. Kwasnitza
11 Folienservice	RED
12 Leserbrief	G. T. Klement
Zu DATALINE und SCROLL	G. T. Klement
13 2742/64-EPR0M Programmer - Teil 1	Bernd Schuhmacher
22 NASCOM-1 mit 4 MHz	Michael Bach
23 Abenteuerspiel "Lost Dutchman's Gold"	Michael Bach
23 NASCOM-Taufelchen	Michael Bach
24 STRINGSAVE	G. T. Klement
24 VARPROTECT	G. T. Klement
24 SWINGHOUSE	G. T. Klement
25 Mini Buffer Board	H. G. Ingelaat / U. Forke
26 2 Monitorprogramme auf EPROM Board	Clemens Ballarin
26 Clock-Umschaltung 2/4 MHz	Clemens Ballarin
27 Drehzahlmessung für Cassettenrekorder	H. Gnirck
27 Handshake für UART	Peter Urban

**LISTINGS:**

14 Lost Dutchman's Gold	Michael Bach
19 E- und F-Format	J. Kwasnitza
19 TV-Kamera	Günter Böhm
20 DATALINE Demo	G. T. Klement
20 Entfernung- und Winkelprogramm (Nachtrag)	W. von Jan
21 STRINGSAVE	G. T. Klement
21 SWINGHOUSE	G. T. Klement
22 VARPROTECT	G. T. Klement

UND ANSONSTEN: Kleinanzeigen + NASCONPL

# Impressum

## Cassetten-Rundlauf

Aus organisatorischen Gründen wurde der Cassettenversand für die Ausgaben 9 und 10/82 zusammengelegt - so kommt wenigstens einiges zusammen. Autoren, die ihren Beitrag auf Cassette eingeschickt haben, bekommen diese Cassette in Kürze mit allen Programmen bespielt

REDAKTION: Günter Böhm, Günter Kreidl  
Wolfgang Mayer-Gürr, Josef Zeller  
RESSORTS:  
MASCHINENPROGRAMME:  
Günter Böhm, [REDACTED]  
75 Karlsruhe, [REDACTED] Tel. [REDACTED]  
Günter Kreidl, [REDACTED] Straelen  
Tel. [REDACTED]  
BASIC und FLOPPY:  
Wolfgang Mayer-Gürr, [REDACTED]  
[REDACTED] Recklinghausen  
Tel. [REDACTED]  
HARDWARE:  
Josef Zeller, [REDACTED], [REDACTED] Neu-Ulm  
VERLAG: NASCOM JOURNAL, c/o MK-Systemtechnik  
Pater-Mayer-Str.6, 6728 Germersheim  
Tel. 07274/2756 2033 Telex 453500 mkxd  
VERTRIEB: Direktvertrieb durch den Verlag  
Erscheinungsweise: monatlich  
Bezugspreis: Im In- und Ausland 48.- für  
ein Jahresabonnement. Abonnements können  
aus technischen Gründen immer nur für die  
Dauer eines Kalenderjahres, d.h. vom 1.1.  
bis 31.12. laufen. Bei Bestellung nach dem

zurück. Für die übrigen Leser sind die Programme in einem Rundlauf von Herrn Ballarin erhältlich.

Im Programm STRINGSAVE hat sich ein Käfer eingenistet, wie uns Herr Klement in letzter Minute mitteilt. Statt DOKE 4100,3234 muß es heißen DOKE 4100,3226.

1.1. werden die fehlenden Hefte mit der ersten Lieferung bis zum Bestellzeitpunkt automatisch mitgeliefert. Bei nicht fristgerechter Kündigung verlängert sich das Abonnement automatisch um ein Jahr. Die Kündigung für das Folgejahr muß bis spätestens sechs Wochen vor Jahresende erfolgen.  
Bezugsmöglichkeiten: Durch Bestellung bei MK Systemtechnik.  
Bankverbindungen: Alle Zahlungen für das NASCOM JOURNAL unter Angabe der Rechnungsnummer an MK - Systemtechnik, Germersheim.  
Zahlung: Nach Eingang Ihrer Bestellung erhalten Sie von uns die ausstehenden Hefte bis zur aktuellen Ausgabe sowie eine Rechnung. Bitte, zahlen Sie dann den Rechnungsbetrag.  
Bitte keine Vorauszahlungen!  
Bitte, Anfragen wegen Abonnements oder Lieferung nicht an die Redaktion sondern nur an den Verlag. Die Autoren tragen die Verantwortung für ihre Beiträge selbst. Für Fehler in Text, Bildern und sonstigen Angaben kann keine Haftung übernommen werden.

## MITARBEITER

Josef Zeller, Günter Böhm, Günter Kreidl siehe Impressum \* H.G. Ingelaat u. U. Forke, A. [REDACTED], [REDACTED] Emmerich \* G.W. Delius, [REDACTED], [REDACTED] Bielefeld [REDACTED] \* Constantin Olbrich, [REDACTED], [REDACTED] Berlin \* Gerhard Wilharm, [REDACTED], [REDACTED] Bergheim [REDACTED] \* Peter Urban, [REDACTED], [REDACTED] Niefern [REDACTED] \* Gregor Birnfeld, [REDACTED], [REDACTED] Königsfeld, [REDACTED] \* Otto Föbel, [REDACTED], [REDACTED] Erlangen, [REDACTED] \* J. Kwasnitza, [REDACTED], [REDACTED] München [REDACTED] \* Gerhard T(eddy?) Klement, [REDACTED], [REDACTED] Wien, Österreich, [REDACTED] \* Bernd Schuhmacher [REDACTED], [REDACTED] Karlsruhe \* Michael Bach, [REDACTED], [REDACTED] Stegen, [REDACTED] \* Wolfgang von Jan, [REDACTED], [REDACTED] Langenhagen, [REDACTED] \* Clemens Ballarin, [REDACTED], [REDACTED] Überlingen, [REDACTED] \* H. Gnirck, [REDACTED], [REDACTED] Bietigheim-Bissing, [REDACTED]

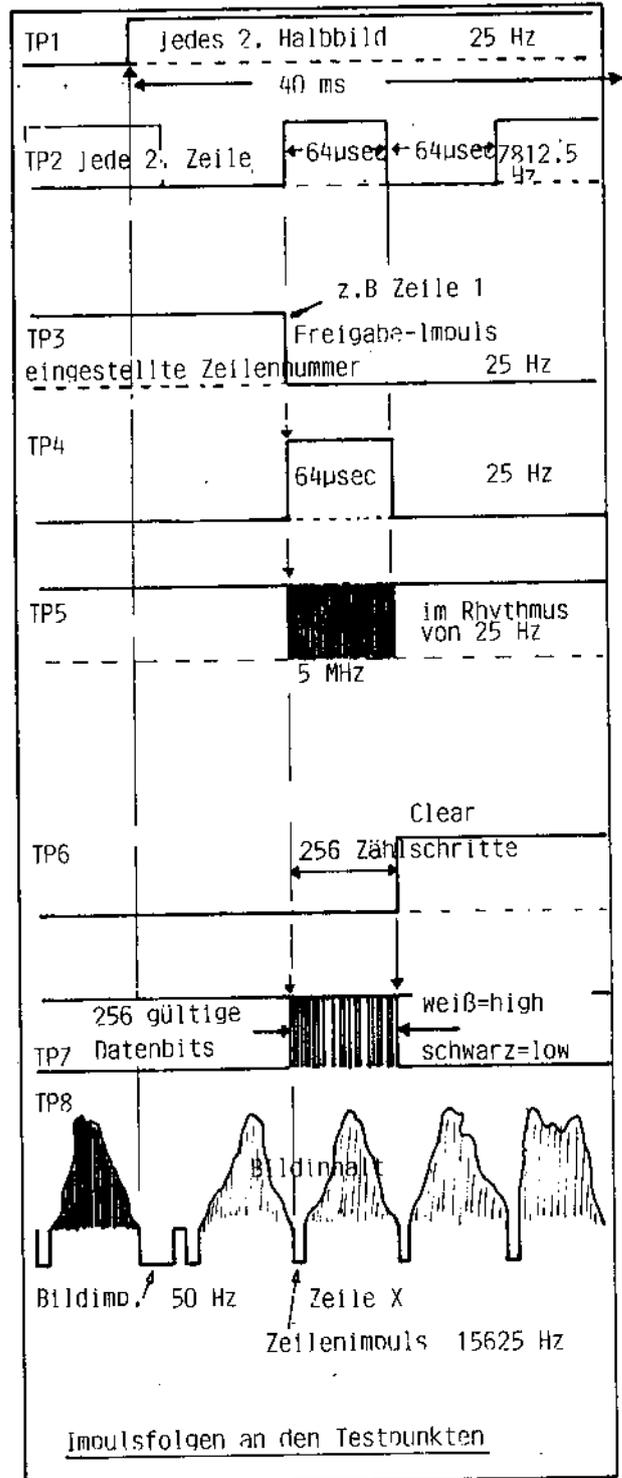
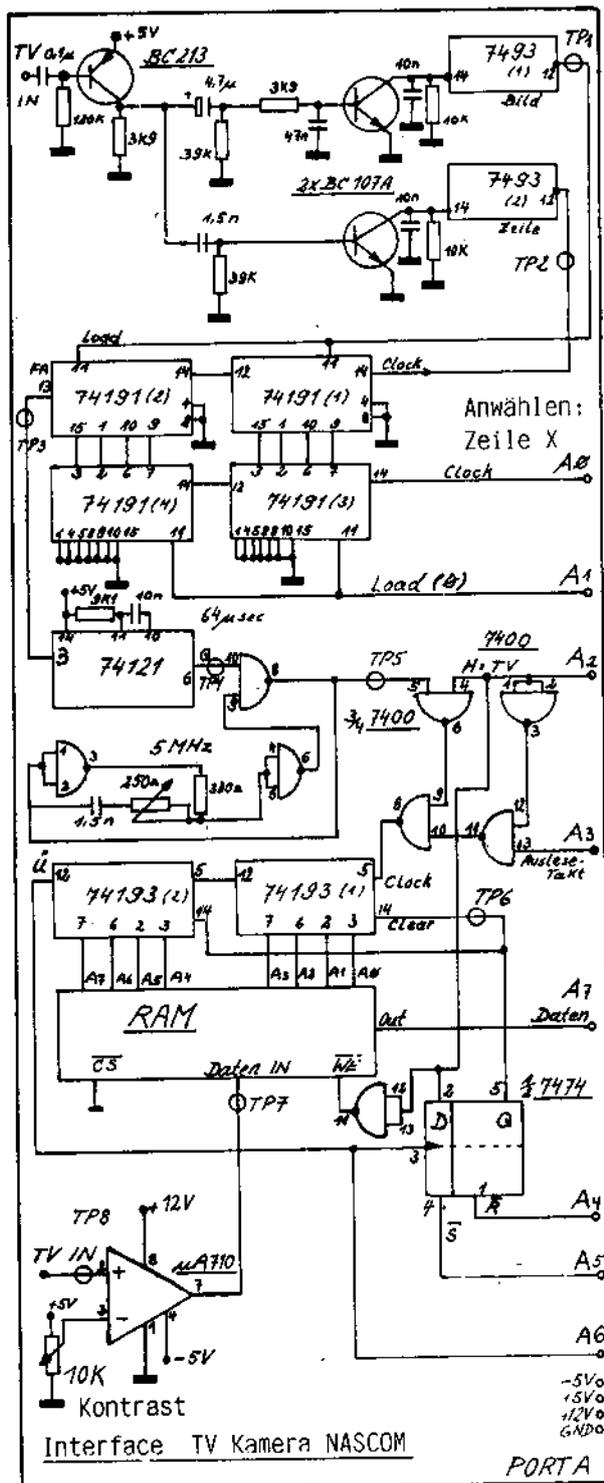
Das Copyright (c) verbleibt grundsätzlich bei den Autoren.

# TV-Kamera Interface

## Teil 2 von Günter Böhm

Die abgedruckte Schaltung zum Interface der Video-Kamera wurde zunächst versuchsweise auf einer Lochrasterplatine frei verdrahtet. Bevor ich darangehe, ein Platinen-Layout zu erstellen, das Ihnen selbstverständlich wie-

der zur Verfügung steht, müssen noch einige Fragen geklärt werden, wobei mir einige Leser sicher behilflich sein können. An Bauteilen habe ich verwendet, was meine Bastelkiste hergab, deshalb auch die Verwendung von Bausteinen, die ihre Funktion nicht unbedingt optimal erfüllen. So sollten anstelle der 191 besser die 193 Zähler verwendet werden, sie sind leichter rückzustellen.



Die Teilung der Bild- und Zeilensynchronfrequenz sollte ursprünglich mit einfachen Fluo-Flops erreicht werden. Damit hatte ich kein Glück. Nur mit den 7493 bekam ich die sauberen Signale, die ich brauchte. (Die 10n am Eingang konnte ich allerdings weglassen).



Das Ergebnis der Schaltung in der Praxis sehen Sie im ersten Bild. Das Prinzip funktionierte, zeigte aber seltsame Störungen. Nach dem Ausdruck eines Kreises, entdeckte ich eine Regelmäßigkeit der Fehlerhaften Zellen: Jede sechzehnte Zelle war zu früh abgebildet. (BILD 2).



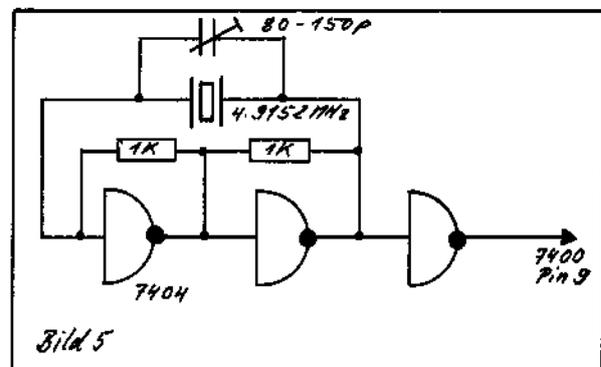
Erst nach vielen Überlegungen versuchte ich, den Zähler zum Anwahl der Zeilennummer ebenfalls rückwärts laufen zu lassen, da vermutlich beim Rückwärtszählen der Übertrag erst von U auf HF erfolgte, und so die beiden Zähler die eingestellte Zahl verschieden interpretierten. (Habe ich recht mit meiner Vermutung)? Jedenfalls funktionierte die Änderung auf Anhieb. Die Ausgabe (Bild 3) war nun allerdings spiegelverkehrt, da links und rechts zwar beibehalten wurden, die Ausgabe aber mit Zeile 156 begann, was mich aber nicht weiter störte.



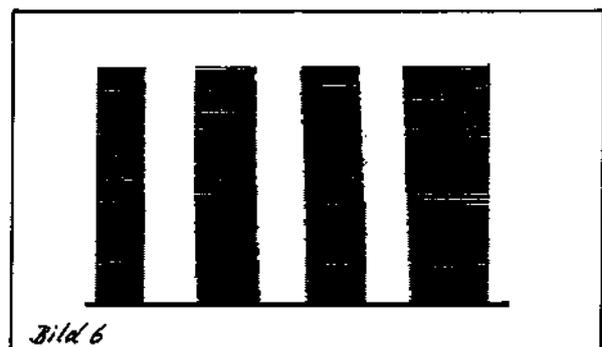
Zufrieden mit den Ergebnissen, begann ich nun, Portraits von einigen Bekannten aufzunehmen. Die beschwerten sich allerdings bald darüber, daß sie alle mit geschwollener Backe abgebildet wurden. Die Festaufnahme mit einem senkrechten Streifenmuster zeigte dann auch hier eine Regelmäßigkeit, die mich sehr stark an einen Sinus erinnerte. (Bild 4).



Bei diesem Stichwort fiel mir "Netzbrumm" ein, und ich versuchte, durch Kondensatoren an allen möglichen Stellen auf der Platine eine Verbesserung zu erreichen, die aber letztlich nicht nennenswert war. Also wurde kurzerhand ein Quarz-Oszillator aufgebaut (Bild 5). Ich bekam nur einen Quarz mit 4.9152 MHz, aber der war zufälligerweise

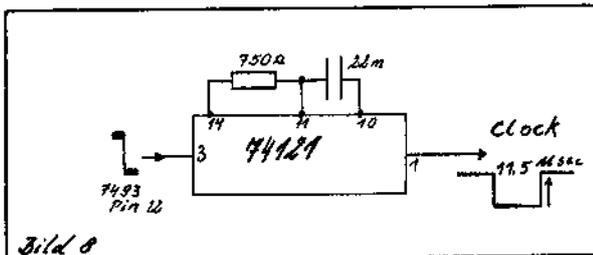


genau richtig. Durch Parallelschalten eines Trimmers von ca. 80-150 pF kann man ihn im Notrill variieren. Der Einsatz des Quarzes hat sich gelohnt; das Ergebnis sehen Sie in Bild 6 und 7.

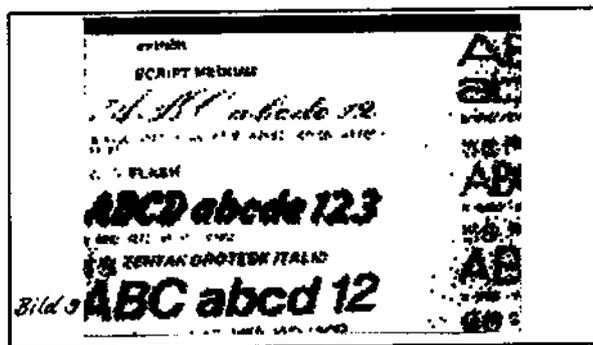




In Bild 7 erkennen Sie rechts einen feinen Rand, der das eigentliche Bild von einem schwarzen Streifen abtrennt. Dieser Streifen nimmt etwa ein Fünftel des Speicherplatzes ein. Er wird beim Abtasten des Bildes durch die Schwarzscherle erzeugt, die jedem Zeilenimpuls folgt. So habe ich den Zeilenimpuls einfach durch ein Monoflop um 11,5 usec verzögert, und dadurch die Auslösung des Bildes wieder etwas verfeinert. (Bild 8)



Um sich ein Bild von der Auflösung zu machen, habe ich Ihnen auch ein Schriftbild ausgedruckt. Durch Verwendung eines reineren Tuschestiftes konnte sie noch etwas gestel-



gert werden. Ganz zufriedengestellt war ich noch nicht, und habe etwas weiterexperimentiert. Hier müßten mir einige Leser weiterhelfen, denn die Ergebnisse sind mir unverständlich.

1. Die Schaltung funktioniert auch, wenn der Bildsynchronimpuls nicht geteilt wird! Könnte es sein, daß meine Kamera überhaupt keine verschiedenen Teilbilder liefert?

2. Wenn ich die Zeilenfrequenz nicht halbiere (also jede Zeile abtaste), wird nach jeder richtigen Zeile (die ersten Zeilen sind immer schwarz) zunächst ein Teil des Bildes abgebildet. Erst nach einigen Zeilen erfolgt die richtige Darstellung des Bildes (Bild 10).



Bild 10

Wenn ich schon beim Fragen bin:

3. Warum funktioniert meine P10 (Erweiterung) nur dann als Ausgang, wenn ich den anderen Port zuvor auch auf Ausgang programmiere?

Wenn ich auf obige Fragen die richtigen Antworten bekomme, werde ich in mein Platinen-Layout die Möglichkeit einbauen, ein Bild mit halber oder ganzer Zeilenanzahl abzutasten, um so zwischen zwei Auflösungen zu wählen. Die einzelnen Zellen in mehr als 256 Punkte zu zerlegen, würde den Schaltungsaufwand wohl zu sehr vergrößern. Zudem nimmt der Speicher im Augenblick nur etwa 4 1/2 K in Anspruch, bei einer Steigerung auf 256 Zellen bereits 8K. Es tritt auch das Problem auf, ein Bild auf dem Monitor betrachten zu können. Eine hochauflösende Grafik müßte entsprechend viel Bildwiederholpeicher zur Verfügung stellen.

Zum Problem der Bildbetrachtung mit unserer 48X96 Grafik noch einiges im 80-Bus Journal. Diesmal nur ein kleines BASIC Programm, um die Funktion der Schaltung zu testen. Es tastet das Fernsehbild Zeile für Zeile ab und stellt einen kleinen Ausschnitt davon mit der N2 Grafik dar. (Bild 11 zeigt den Ausschnitt meines Konterfeis). Bitte nicht

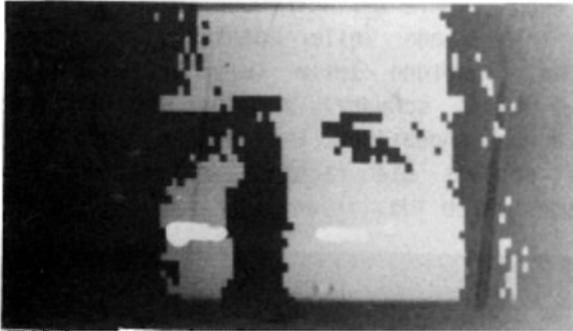


Bild 11

vor der Kamera bewegen! Das Programm benötigt dreieinhalb Minuten, bis das Bild fertig ist. Ein schnelles Maschinenprogramm folgt im nächsten Heft. Am BASIC Programm können Sie den Ablauf der Platinensteuerung gut verfolgen. Zum besseren Verständnis soll die Tabelle der Steuerworte dienen.

Wer schon jetzt an den Nachbau der Schaltung gehen möchte, sollte wissen, daß die TTL RAMs sehr schwer zu bekommen sind. (Meines stammt aus einer geheimen Quelle). Man müßte versuchen, ob man die Typen /4LS200A, LS202A oder entsprechend LS3... direkt vom Hersteller beziehen kann; bei den großen Versandunternehmen hatte ich kein Glück. Ich möchte aber zunächst noch die Eignung der 2102 ausprobieren. Wenn die nicht zu langsam sind, könnte man sie als preiswerte Alternative überall bekommen.

Zur Vollständigkeit hier noch die Anschlußbilder des TTL RAMs, das von mir benutzt wurde, und des  $\mu A710$ , den ich leider nur im Metallgehäuse bekam.

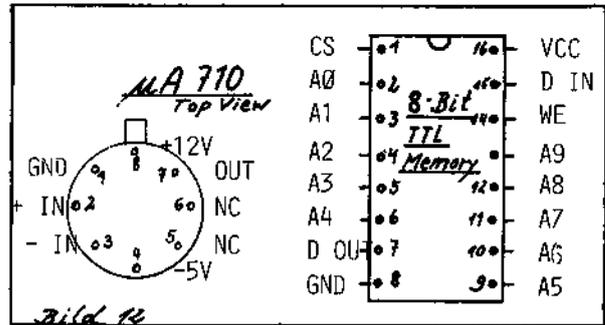


Bild 12

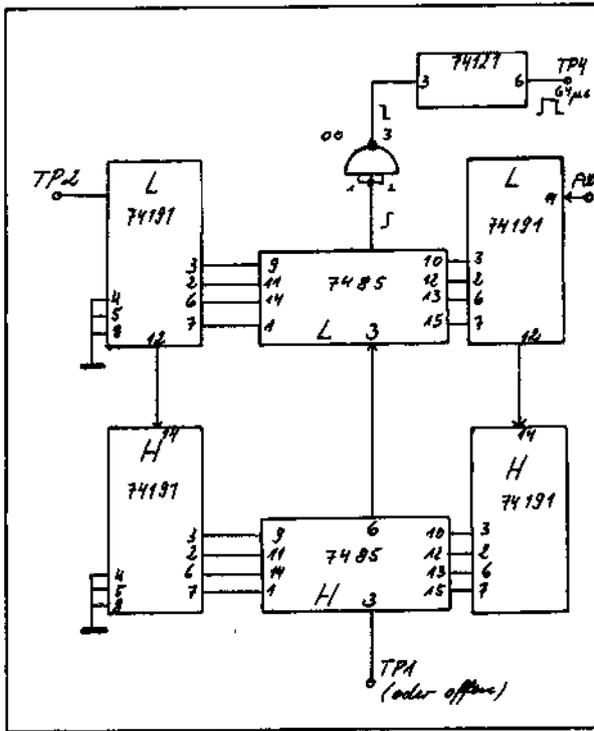
Im 80-Bus Journal folgt die Software zum Abspeichern eines Bildes und zur schnellen Darstellung auf dem Bildschirm. Nach Möglichkeit kann ich dann auch schon eine Atzfolie liefern, allerdings nur, wenn die noch offenen Fragen (siehe oben) rechtzeitig geklärt werden.

Die Beispiele sind übrigens alle mit meinem Selbstbau-Plotter ausgegeben, der augenblicklich noch zum Verkauf steht.

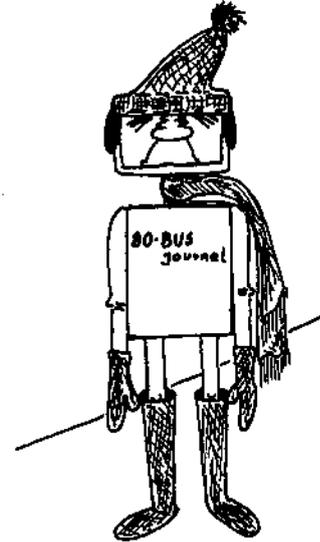
Nach Fertigstellung des obigen Artikels habe ich noch eine kleine Verbesserung vorgenommen, die Ihnen nicht vorenthalten werden soll. Um die Ausgabe wieder seltenrichtig zu machen, wurden die Zeilenzähler wieder beide auf Vorwärtslauf gestellt. Die Ausgänge wur-

STEUERWÖRTE	I N P U T		O U T		P U T		N .		HEX	DEZ
	Daten IN	RAM voll?	RAM Zähler Clear	RAM Zähl. Enabl.	Clock Auslesen	TV/NASC. Umsch	RESET Zeile Nr.	Clock Zeile Nr.		
	A7	A6	A5	A4	A3	A2	A1	A0		
	128	64	32	16	8	4	2	1		
Reset Zeile/RAM ..Ausgangsposition 1	x	x	1	0	0	0	0	0	20	32
Zeilenummer vorwählen (Clock)			1	1	0	0	1	0	32	50
			1	1	0	0	1	1	33	51
			1	1	0	0	1	0	32	50
Zelle einlesen			1	1	0	1	1	0	36	54
Einlesen Ende		1							40	64
NASCOM / RAM Zähler freigeben			1	0	0	0	1	0	22	34
Auslesetakt			1	0	1	0	1	0	28	42
" "			1	0	0	0	1	0	22	34
Daten IN (schwarz=0 / weiß=1)	x								80	28
Clear RAM Zähler			0	1	0	0	1	0	12	18
Ausgangsposition 2 (Zeilen Nr nicht rücksetz.)			1	0	0	0	1	0	22	34

den mit zwei 4-Bit-Komparatoren verbunden, wenn nun der Zeilenzähler die Zahl erreicht hat, die mit dem anderen Zähler durch den Rechner eingestellt wurde, wird ein Impuls ausgegeben, der invertiert das Monoflop zum Einlesen der Daten startet. Hier nun die Schaltung.



# NASCOMPL



*Wallo liebe Leser!*

Nun sind wohl die Zeiten vorbei, in denen Sie für Ihren NASCOM den Lufter einschalten mußten. Im Gegenteil: die tiefen Temperaturen, die von nun an noch weiter fallen werden, machen Ihrem Rechner das Leben schwer. Von einigen Lesern haben wir in letzter Zeit gehört, daß das 8K-BASIC bei geöffnetem Fenster nicht mehr läuft. Kein Wunder, wenn man den Computer ungeschützt den niedrigen Temperaturen aussetzt. Wozu haben wir das NASCOM Journal? Selbstverständlich kommt es Ihnen auch jetzt wieder zu Hilfe. Wir haben nicht nur einen Folienservice, wir haben nun auch einen Strickmuster-Dienst eingerichtet. Schicken Sie uns einfach die Gehäusemaße Ihres NASCOM und wählen Sie dann aus unserem Programm:

1. Gehäusepullover
2. Platinenmütze
3. Lastaturwärmer
4. Steckersocken

Unser Programm wird laufend erweitert. Es ist zur Zeit gerade ein Twin-Set für den NASCOMs in Arbeit. Bitte rechtzeitig bestellen, da die Nachfrage groß ist. (In Sonderfällen strickt Herr Kreidl auch selbst. Ausgestrickt und getestet nur 12,50 DM pro Teil).

In diesem Sinne Ihr NASCOMPL

\*\*\*\*\*

## KLEINANZEIGE in eigener Sache

Anfang Dezember wurde der Vertrieb von NASCOM-Produkten von uns übernommen. Hierzu gehört auch das totgesagte NASCOM-JORNAL, das sich entgegen allen anderslautenden Gerüchten noch höchster Lebendigkeit erfreut. Unsere Übernahme des Journals bedeutet keine Veränderung in Inhalt oder Organisation. Mehr darüber in der in Kürze erscheinenden Dezemberausgabe.

Wir bitten die durch die Umstellung des NASCOM-Vertriebs bedingten Verzögerungen zu entschuldigen.

Unsere Adresse:

L A M P S O N  
Analog & Digitaltechnik  
NASCOM-Vertrieb  
Odenwaldstrasse 21-23  
6087 Büttelborn  
Tel.: 06152/56730

\*\*\*\*\*

# Seite(n) für Einsteiger

von Otto Föbel

Wissenswertes über Interrupts !!

In den Geburtsjahren der Computer galt das Hauptinteresse der Verarbeitung von Zahlenmaterial. Die Möglichkeiten dieser neue Entwicklung führten zu den bekannten Großrechenanlagen in der Größenordnung ganzer Gebäudekomplexe. Erst in den 60er Jahren eröffnete sich durch neue Technologien zur Miniaturisierung die Möglichkeit, frei programmierbare Geräte auch für Steuerungsaufgaben einzusetzen. Mit diesem Ziel vor Augen, entstanden die Mikroprozessoren, deren Struktur den neuen Anforderungen angepaßt wurde.

Man opferte die Fähigkeit zur direkten Verarbeitung von Gleitkommazahlen, die Möglichkeit komplexer arithmetischer Operationen usw. und beschränkte sich auf 8 Bit breite Datenworte. Mit der Abmagerung bekannter Prozessorelemente war es aber nicht getan. Eine der wichtigsten neuen Anforderungen war es, den Kontakt mit der Außenwelt herzustellen. Die Lösung dieser neuen Aufgabe kennzeichnet die Mikroprozessoren und damit sind wir auch (endlich) beim Thema.

Wie allen bekannt, geht es in so einem Mikroprozessor, solange er funktioniert, recht geordnet und nach strengen Regeln zu. Der Oberaufpasser ist der Taktgenerator, der sämtliche Abläufe im Griff hat, und wenn die Sache einmal läuft, wird mit jedem neuen Takt eine neue Operation eingeleitet, aber wie gesagt, immer schön synchron zum Takt und streng nach den Regeln und Programmanweisungen.

Dem gegenüber steht eine Außenwelt, die sich um diesen Taktgenerator überhaupt nicht kümmert, weil sie entweder nichts von ihm weiß oder ihm einfach nicht folgen kann, da er zu schnell ist. Trotzdem muß bei Steuerungsaufgaben diese Umwelt mit in die Abläufe des Computers einbezogen werden. Dafür eröffnen sich zwei Möglichkeiten, die zuerst allgemein dargestellt werden sollen.

1.) Der Mikroprozessor wird so programmiert, daß er regelmäßig durch spezielle Befehle bestimmte Zustände der Außenwelt abfragt. Diese Möglichkeit, "polling" genannt, beruht alleine auf der Software und ist relativ unproblematisch. Bestimmte Zustände, z.B. ob ein Schalter geöffnet oder geschlossen ist, werden in ent-

sprechende "TTL-Pegel" (d.h. 0V oder 5V) umgewandelt und über ein E/A-Port eingelesen. Dem entsprechend kann dann im Programm verzweigt und reagiert werden.

In der Realisierung stellt sich aber diese Möglichkeit als sehr unbefriedigend heraus, da der Prozessor im Grunde nichts anderes macht, als zyklisch eine Verzögerungsschleife zu durchlaufen, an deren Ende das E/A-Port abgefragt wird. Viel mehr kann er nebenbei nicht machen. Ein weiterer Nachteil ist die eingeschränkte Reaktionsfähigkeit auf externe Ereignisse. Als Paradebeispiel die Situation des Stromausfalls. Würde dies sofort dem Prozessor mitgeteilt, könnte er in der kurzen verbleibenden Zeit bis auch die Pufferkapazitäten im Netzteil entladen sind, die wichtigen Informationen, wie Stand des Programmzählers und den Zustand der Register, in einen batteriegepufferten Speicher retten. Das würde genügen, um das Programm später genau dort fortsetzen zu können, wo der Strom ausfiel.

2.) Die Möglichkeit zu diesem schnellen Eingreifen bietet der Interrupt. Die Außenwelt wartet nicht mehr, bis sie bedient wird, sondern meldet dies sofort, wenn etwas los ist. Die interne Steuerung des Prozessors sorgt dafür, daß der laufende Befehl noch abgearbeitet wird, unterbricht dann den Programmablauf, so wie er vom Programmierer geplant war, und springt zu einem ganz anderen Programmteil, der Interrupt-Service-Routine, die dann auf die Unterbrechung reagiert. Danach wird dorthin im Programmablauf zurückgekehrt, wo die Unterbrechung eintraf.

Soweit ganz allgemein das Problem der Interruptbehandlung. Was aber, wenn mehrere Geräte einen Interrupt senden wollen, was ist, wenn zwei Geräte gleichzeitig unterbrechen, woher weiß der Prozessor, von wem die Unterbrechung kommt, und welche soll er zuerst bedienen, und woher weiß er vor allem, wo das entsprechende Unterbrechungsprogramm im Speicher liegt?

Also genug Probleme, die es noch zu lösen gibt, und gerade in der Lösung dieser Probleme liegt das Erfolgsrezept eines guten Mikroprozessors. Im weiteren soll untersucht werden, wie beim Z80 dieses Problem in Angriff genommen wurde. Wichtig für die Interruptstruktur des Z80 waren die Vorarbeiten von Intel. Nachdem der Z80 Software-kompatibel zum 8080 von Intel sein sollte, d.h. Programme dieses bis dahin schon weitverbreiteten Typen sollten unverändert auf dem Z80 laufen können, mußten auch die Inter-

ruptmöglichkeiten des 8080 abgedeckt werden. Daneben ließ man sich natürlich auch noch etwas neues einfallen.

Das Ergebnis sind drei verschiedene Interrupt-Modes, mit denen der Z80 laufen kann und die vorher entsprechend mit dem Befehl "IM mode" (mode=0,1,2) eingestellt werden müssen. Zuvor aber noch einen Blick auf die Hardware. Nachdem an jedem zusätzlichen Pin gespart werden muß, kann also nicht für jeden Interrupt ein eigener Anschluß an der CPU vorgesehen werden. Beim Z80 wurden für diesen Zweck lediglich zwei Pins spendiert. 1.) Der Eingang "NMI" = nicht maskierbarer Interrupt. Eine Unterbrechung an diesem Pin kann nicht maskiert, d.h. softwaremäßig nicht unterdrückt werden und besitzt somit die höchste Dringlichkeitsstufe. Die Interrupt-Bedienroutine muß bei Adresse 66H liegen, d.h. die Verzweigung dorthin ist hardwaremäßig vorgegeben und deshalb auch besonders schnell. Die Interrupt-Bedienroutine muß beendet werden mit dem Befehl "RETN".

2.) Der Eingang "INT" dient als Sammelleitung sämtlicher anderer Interruptanforderungen. Dieser Eingang kann softwaremäßig durch die Befehle Interrupt-Enable und Interrupt-Disable gesperrt oder freigegeben werden. Als weitere Besonderheit ist das Verzweigen auf verschiedene Interrupt-Bedienroutinen möglich und das Priorisieren der verschiedenen Interrupts. Der letzte Begriff muß, da er eine ganz wichtige Rolle spielt, noch näher erklärt werden. Hierbei geht es um die Fragestellung, welcher Interrupt vorrangig zu behandeln ist. Diese Frage ist vor der Realisierung zu klären und kann dann zu folgendem Ablauf führen: Der Prozessor, der gerade eine Unterbrechung bearbeitet, erhält die Meldung eines noch wichtigeren Interrupts. Daraufhin wird er das erste Interrupt-Bedienprogramm verlassen und den Interrupt mit der höheren Priorität zuerst behandeln. Danach kehrt er dann zum ersten Interrupt-Bedienprogramm zurück, beendet dieses, und danach kehrt er ins Hauptprogramm zurück. Bereits geübte Programmierer erkennen eine Parallele zu den geschachtelten Unterprogrammaufrufen. Ein weiterer Vergleich mit Unterprogrammen ist möglich. Durch eine Unterbrechung wird, wie beim CALL-Befehl, die Programmausführung an einem anderen Punkt fortgesetzt und danach wieder zur ursprünglichen Stelle zurückgekehrt. Und genau wie beim CALL-Befehl wird dafür der Programm-Zähler automatisch auf den Stack gerettet. Im Unterschied

zum Unterprogramm muß beim Interrupt die entsprechende Routine mit dem Befehl "RETI" (return from interrupt) abgeschlossen werden. Aber jetzt zurück zu den drei verschiedenen Interrupt-Modes, die den Eingang "INT" betreffen. Mode 0: Dieser Mode entspricht genau der Interruptbehandlungsweise des 8080. Dabei muß neben dem Interruptsignal das unterbrechende Gerät auch den Datenbus bedienen. In der Regel wird dieses Gerät die Bitfolge eines Restart-Befehls auf den Datenbus legen. Der Prozessor holt also den nächsten Befehl nicht vom Speicher, sondern der nächste Befehl wird vom peripheren Gerät bestimmt. Speziell dafür gibt es die Ein-Byte-Befehle "RST", die gleichzeitig die Startadressen der Bedienroutinen enthalten. Da dieser Befehl nur ein Byte lang ist, sind damit nur 8 Interrupt-Bedienroutinen auswählbar, deren Startadressen im Bereich 0 - 38H liegen. Wegen dieser eingeschränkten Möglichkeit wird man diesen Mode also nur verwenden, wenn bereits vorhandene 8080-Programme ablaufen sollen.

Mode 1: Auch dieser Mode dient zur Herstellung der Kompatibilität zum 8080 und verzweigt zu einem Bedienprogramm, das bei Adresse 38H beginnen muß.

Mode 2: Dieser Mode bietet die umfassendsten Verzweigungsmöglichkeiten und muß in Verbindung mit den E/A-Bausteinen des Z80 gesehen werden (PIO, SIO, CTC, DMA). Auch im Mode 2 muß beim Interrupt der Datenbus bedient werden. Da die externen Geräte dazu meist nicht in der Lage sind, wird der Interrupt zuerst zu einem dieser programmierbaren E/A-Bausteine geleitet und diese Bausteine übernehmen dann die Weitergabe des Interrupts an die CPU und die Bedienung des Datenbuses. Eine weitere Funktion liegt noch in diesen E/A-Bausteinen: nämlich die Priorisierung. Diese Bausteine sind ähnlich den Perlen einer Kette durch eine Leitung miteinander verbunden, wobei der Baustein, der in dieser Kette an oberster Stelle liegt, den höchstwertigen Interrupt erzeugt und gleichzeitig alle darunterliegenden Bausteine daran hindert, gleichzeitig einen Interrupt zur CPU zu senden. Damit ist das Problem der Priorisierung hardwaremäßig gelöst.

An einem Beispiel soll jetzt der weitere Ablauf einer Interruptmeldung beschrieben werden. Eine PIO (paralleler E/A-Baustein) erhält von einem externem Gerät die Interruptmeldung. Die Hardware-Priorisierung bestimmt jetzt, ob dieser Interrupt zur CPU weitergeleitet werden

kann. Ist also kein höherwertiger Interrupt in Bearbeitung, so sendet unsere PIO das "INT"-Signal und legt anschließend ihr Vektor-Register auf den Datenbus. Dieses Vektorregister der PIO muß in der Initialisierungsphase vorgebelegt werden. Die CPU, die jetzt auf den Interrupt reagiert, liest dieses Byte und interpretiert es als das niederwertige Byte einer Adresse. Das höherwertige Byte dieser Adresse holt sich die CPU aus ihrem I-Register (Interrupt-Register), das ebenfalls entsprechend vorgebelegt sein muß. Damit wird eine 16-bit Adresse generiert, die dann auf den Adreßbus gelegt wird. Diese Adresse zeigt in eine Sprungtabelle und dort steht erst die Anfangsadresse der Interrupt-Service-Routine. Somit ist also eine echte speicher-indirekte Adressierung mit Hilfe einer Sprungtabelle möglich. Der Umweg über diese Sprungtabelle bietet folgende Vorteile:

- 1.) Die dort abgelegte, 16-bit Adresse der Interrupt-Service-Routine (ISR) kann zu jedem beliebigen Punkt des 64k-Speicherraumes führen.
- 2.) Es können beliebig viele ISR angesprungen werden.
- 3.) Die Anfangsadresse der ISR in der Sprungtabelle kann während der Programmabarbeitung geändert werden. Dies ermöglicht eine dynamische Zuweisung zwischen Peripherie und ISR.

Zum Schluß noch eine Zusammenstellung der Initialisierungsschritte, die beim Betrieb im Interruptmode 2 notwendig sind:

```
DI Disable Int., für geordnete Anfangsverhältnisse
OUT ... Interrupt-Vektor in E/A-Baustein laden
LD I,A Interruptregister der CPU laden
IM 2 Interruptmode 2 einstellen
EI Interrupt freigeben
```

Die Beschreibung sollte einen Einblick in die Interruptbehandlung von Mikroprozessoren geben. Wenngleich auch im zweiten Abschnitt sehr speziell auf die Struktur des Z80 eingegangen wurde, so kann doch mit dem Verständnis dieses Typs relativ leicht die Interruptbehandlung anderer Typen verstanden werden. Die Feinheiten liegen, wie auch hier schon deutlich wurde, nicht mehr in der CPU selbst, sondern werden in zunehmendem Maße in andere Bausteine verlagert. Ob dies, wie beim Z80, die E/A-Bausteine oder spezielle Interrupt-Controller sind, eines haben sie gemeinsam: Die Komplexität ihrer Struktur kann es fast mit der der CPU

aufnehmen und dementsprechend umfangreich sind auch die Beschreibungen solcher Bausteine, durch die man sich wohl oder übel durchbeißen muss. Viel Spass dabei !

## Datenausgabe im E- u. F-Format von J. Kwasnitza

Bekanntlich hat das Nascom-Microsoft-Basic keine direkt zugaenglichen Befehle fuer die formatierte Datenausgabe und waehlt je nach Zahlenwert selbststaendig ein passendes Ausgabeformat. Das hat zur Folge, dass die Ausgabe ordentlicher Zahlentabellen ohne besondere Massnahmen nicht moeglich ist. Anhand des Testprogrammes Nr. 1 wird das noch einmal demonstriert.

Wer das Arbeiten in Fortran gewoehnt ist wuenscht sich da die Verfuegbarkeit der E- und F-Format-Befehle. Fuer alle, die mit Fortran nicht vertraut sind, sei hier kurz die Funktion dieser Befehle beschrieben.

Im E-Format wird eine Variable in folgender Form dargestellt:

nBlanks, Signum, Mantisse, ., Mantisse, E, Signum, Exponent, 2-stelliger Exponent

Beispiel: " -.123456E+21"

Durch Befehle kann die Stellenzahl der Mantisse DE und die gesamte Feldbreite WE einschliesslich der Blanks festgelegt werden.

Beispiel: DE=5, WE=12 gibt die Zahl  
X=-1.23456 als XE\$=" -.12346E+01" aus.

Im F-Format wird eine Zahl wie folgt dargestellt:

nBlanks, Signum, Zahl, Dezimalzahl

Mit dem Befehl DF kann die Anzahl der Dezimalstellen einer Zahl bestimmt werden, waehrend die gesamte Feldbreite der Zahl (einschliesslich der Blanks) ueber WF festgelegt wird.

Beispiel: DF=2, WF=7 gibt die Zahl  
X=-1.23456 als XF\$=" -1.23" aus.

Mit den Basic-Unterprogrammen E-Format (GO-

# Folien-Service

SUB1000) und F-Format (GOSUB1400) koennen nun die Fortran- Befehle simuliert werden. Durch den Aufruf GOSUB1000 wird die Variable X in die Variable XE\$ im E- Format umgewandelt. Vor dem Unterprogramm- Aufruf sind die Variablen WE und DE auf zulaessige Werte zu setzen:

WE (= 15, DE (= 6, WE - DE )= 6

Mit Hilfe des Unterprogramm- Aufrufs GOSUB-1400 erfolgt die Umwandlung der Variablen X in die Variable XF\$ im F- Format. Vor dem Aufruf ist WF UND DF auf zulaessige Werte festzulegen:

WF (= 15, DF (=14

Bei fehlerhafter Vorgabe von WF,DF,WE und DE geben die Unterprogramme entsprechende Meldungen aus. Zu beachten ist, dass in der F-Formatdarstellung eine Variable die vorgegebene Feldbreite ueberschreiten kann. In diesem Fall werden Sterne anstelle der Variablen ausgegeben.

Abschliessend wird nun die Anwendung der formatierten Datenausgabe anhand des Testbeispiels Nr. 2 (Testprogramm Nr. 1 mit Datenformatierung) demonstriert.

```

RUN10
0          0          0
1000      31.6228    1E+06
2000      44.7214    4E+06
3000      54.7723    9E+06
4000      63.2456    1.6E+07
5000      70.7107    2.5E+07
6000      77.4597    3.6E+07
7000      83.6661    4.9E+07
8000      89.4427    6.4E+07
9000      94.8684    8.1E+07
10000     100        1E+08
Break in 50
Ok

```

```

RUN60
.0          .0000    .000000E+00
1000.0     31.6228    .100000E+07
2000.0     44.7214    .400000E+07
3000.0     54.7723    .900000E+07
4000.0     63.2456    .160000E+08
5000.0     70.7107    .250000E+08
6000.0     77.4597    .360000E+08
7000.0     83.6660    .490000E+08
8000.0     89.4427    .640000E+08
9000.0     94.8684    .810000E+08
10000.0    100.0000    .100000E+09
Break in 130
Ok

```

Folgende Atzfolien sind augenblicklich lieferbar:

- MDCR Interface für Mini-Cassettenrecorder
- Grafikerweiterung (N2 Grafik für NASCOM 1)
- Kansas-City-Cassetteninterface
- A/D- Wandler
- Soundgenerator
- Monitor-Umschaltkarte (für 2 versch. Betriebssysteme)
- Video- Karte (80 X 24 Bildschirm)

Eine fertige Platine für EPROM-Programmer ist über Rüdiger Maurer erhältlich.

Der Unkostenbeitrag für eine normale Folie incl. Porto und Verpackung beträgt DM 7,50. Die großen Folien (doopelseitige Platine) für die Videokarte sind für DM 12,50 zu beziehen.

Da es mehrfach Schwierigkeiten durch das Versenden von Bargeld im Briefumschlag gab, bitte ich Sie, von nun an die Kosten für Folien an folgendes Postscheckkonto zu überweisen:

Günter Böhm XXXXXXXXXX PSchA Kirh

Die Überweisung mit Angabe der Folie genügt als Bestellung.

## SONDERAKTION

Da es sich herausgestellt hat, daß manche Leser Schwierigkeiten beim Ätzen und vor allem beim Durchkontaktieren der Video--Platine hatten, haben wir vor, eine fertige Platine zum Selbstkostenpreis anzubieten. Peter Keuntje hat mit einer entsprechenden Firma nun folgenden Preis ausgehandelt:

Epoxy FR4, 1,5mm, 2 x 35 um Bleizinn aufgeschmolzen mit Lötstoomaske

5 Stück Je 152,19

10 Stück Je 101,59

15 Stück Je 92,73 incl.MWSt

Falls Sie an solch einer Platine interessiert sind, schreiben Sie bitte, bis zu welchem Betrag Sie an der gemeinsamen Bestellung teilnehmen würden. Bis zum Ende des Jahres sammeln wir dann die Bestellungen. Sind nicht genügend Interessenten vorhanden, müßten wir die Aktion natürlich einstellen.

Günter Böhm

# Leserbrief

G.T.Klement

Da wäre dann noch ein Thema: Man liest immer wieder von diversen Computerclubs, in denen Software gehandelt wird. Selbstverständlich halte ich es für richtig, einem Software Autor die Materialkosten und Portospesen zu ersetzen. Aber ich wundere mich oft über den Krämergeist, mit dem Computer-hackers ihre oft kläglichen Leistungen versilbern wollen. Ich meine, dass wir die Computerei als Hobby betreiben und Idealisten - so sie es sind - sich gegenseitig nichts verkaufen sollten. Ein geradezu klassisches Beispiel ist unsere Maschine. Da der Benutzerkreis sicher ein atypischer ist, steht bei uns - Gott sei Dank - der Erfahrungsaustausch an erster Stelle. Ein Software Autor sollte sich im Klaren sein, dass es gegen Kopieren ohnehin keinen Schutz gibt. Ich werde es jedenfalls weiterhin so halten, dass ich jedes meiner Programme jedem zur Verfügung stelle. Um nicht in Portokosten zu ersticken, wähle ich gerne den Weg über das NASCOM-Jour-

nal. Ja noch etwas, das mir am Herzen liegt. Ich gebe jedem, der deutsche Beschreibungen und Literatur moniert, zu bedenken, daß er sich in eine schwierige Lage begibt. Die Computersprachen und Hardware Bezeichnungen sind nun mal englisch. Man hat beim Studium von Fachliteratur das Problem, englische Abkürzungen mit gewaltsam eingedeutschten Worten in Verbindung zu bringen, was in der Regel nicht einfach ist. IORQ als Eingabe-Ausgabe Anforderung vielleicht mit EAANF zu übersetzen, gehört zu meinen schlimmsten Alpträumen. Ich habe vom IS, Kellerspeicher, Silospeicher (!) bis zur Schwabbelplatte schon alles durchmachen müssen, und es packt mich das Grauen, wenn ich vom Urbinder höre. Das Fachenglisch hat nur einen begrenzten Wortschatz und auch bei sporadischen Englischkenntnissen hat man sich sehr schnell eingeleesen. Vielleicht könnte man die Sache im Journal einmal diskutieren.  
Gerhard T. Klement (Weichwaren Autor), Wien

## Zu DATALINE und SCROLL

von G.T. Klement

Einige Leser hatten Schwierigkeiten, die BASIC Programme DATALINE und SCROLL aus Heft 7/8-82 praktisch anzuwenden. Deshalb hier einige Tips. Leider bin ich im Moment etwas gestreßt und halte daher meinen Brief kurz ich denke aber mit der TRIAL AND ERROR Methode kann ein NASCOM Freak schon etwas anfangen.

### DATALINE :

Es wird ein RAM bis 8000 vorausgesetzt. Andernfalls muß das M-Code Programm neu assembliert werden.

Nach dem Aktivieren des Maschinenprogrammes mit dem BASIC Befehl SET wird jeder Programm Output in einen Buffer unterhalb 8000 geschrieben. (Die Verwendung von SET ist nur ein Gag, mit USSR ginge es genau so.) Je nach Umfang der Daten wird mehr oder weniger Buffer Bereich benötigt. Es ist daher beim Kaltstart von BASIC eine entsprechende Memsize Begrenzung vorzusehen. (In der vorliegenden Version etwa

15000.) Mit dem Programm sind auch andere BASIC Erweiterungen möglich. Beispielsweise können Rechenergebnisse als sondern findige Köpfe,

Rechenergebnisse als DATA Lines gespeichert werden oder REMARKS werden in Abhängigkeit vom Rechengang in ein Programm eingefügt.

Ich gebe Ihnen als Beispiel weiter unten eine simple Routine, die die ersten Bytes von NAS-SYS 3 als Data Statements generiert. NASCOM Besitzer sind ja keine A\*\*LE, T\*S-80 oder P\*T User, die ihr Gerät zum Entenabschießen verwenden, sondern findige Köpfe, und ich bin selbst gespannt auf weitere Vorschläge aus dem Leserkreis.

### SCROLL :

Das Programm erzeugt einen partiellen Bildschirm SCROLL, d.h. es können im oberen Bildschirmteil Graphiken oder Informationen still gehalten werden, während der untere Teil des Bildschirms normal scrollt. Für einen NASCOM Besitzer ist das keine weltbewegende Sache, denn mit SCREEN kann man ähnliche Effekte erzielen, wenn auch ein klein wenig umständlicher. Einfach einmal probieren und dann weitersehen!

# 2732/64 - EPROM-Programmierer 1

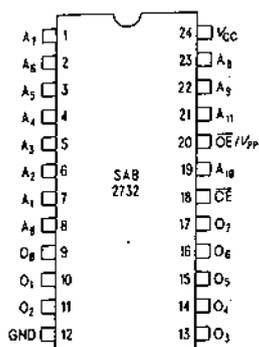
von B. Schuhmacher

Die EPROM's vom Typ 2732 und 2764 werden inzwischen auch für den Hobbybereich immer interessanter. Aus diesem Grunde soll hier ein EPROM-Programmierer für diese Typen beschrieben werden, der mit geringem Hardwareaufwand und unter 1K Software auskommt.

Da nicht jeder über ein teures Datenbuch verfügt oder für die vorgenannten EPROM-Typen anschaffen möchte, stelle ich zunächst die charakteristischen technischen Daten vor. Die Pinbelegungen, Zeitdiagramme und Kurzbeschreibungen für Auslesen und Programmieren erlauben eventuell bereits vorhandene 2708/2716-EPROM-Programmierer selbst zu modifizieren. 2732/2764-EPROM's sind dazu "ausreichend" kompatibel.

## KURZBESCHREIBUNG EPROM 2732

### Pinbelegung



### Charakteristische technische Daten

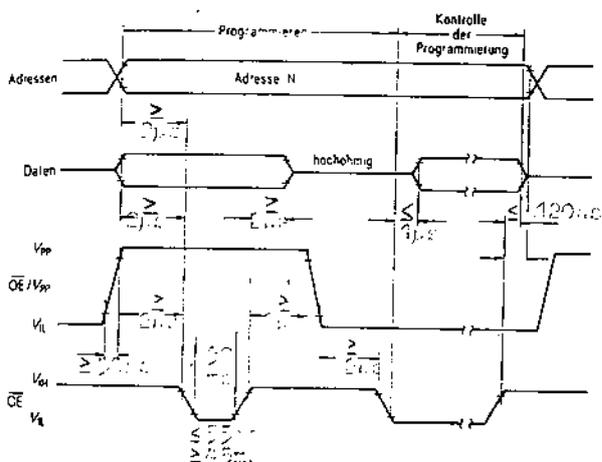
Organisation.....4K x 8  
 Versorgungsspannung Vcc.....5V+-5%  
 Versorgungsspannung Vpp.....25V+-1V  
 Stromaufnahme Vcc(aktiv).....85mA typ.  
 Stromaufnahme Vcc(standby)...15mA typ.  
 Stromaufnahme Vpp.....30mA max.  
 Verlustleistung aktiv.....525mW max.  
 Verlustleistung standby.....132mW max.  
 Die Abstufungen hinsichtlich Verzögerungszeiten, Gehäuseformen, Temperaturgrenzdaten usw. weichen nicht wesentlich von den bei 2708/2716-EPROM's bekannten Werten ab.

Anschlüsse	CE (18)	OE/Vpp (20)	Vcc (24)	Ausgänge (9-11, 13-17)
Lesen	V <sub>IL</sub>	V <sub>IL</sub>	+5V	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	bedeutungslos	+5V	hochohmiger Zustand
Programmieren	V <sub>IL</sub>	V <sub>PP</sub>	+5V	D <sub>IN</sub>
Kontrolle der Programmierung	V <sub>IL</sub>	V <sub>IL</sub>	+5V	D <sub>OUT</sub>
Sperre der Programmierung	V <sub>IH</sub>	V <sub>PP</sub>	+5V	hochohmiger Zustand

Die vorliegende Betriebsartenübersicht beschreibt den Zustand der Steuereingänge CE und OE/Vpp in der jeweils erwünschten Betriebsart.

### Betriebsart "Programmieren" des 2732

Wenn Adressen und Daten stabil anliegen, wird zuerst der Eingang OE/Vpp auf 25V vorbereitet. Durch einen Low-Impuls von ca. 50ms auf den Eingang CE erfolgt dann die Programmierung der eingestellten Adresse, wahlweise sequentiell oder einzeln. Mit OE/Vpp und CE auf Low-Potential) kann eine Prog.-Kontrolle durchgeführt

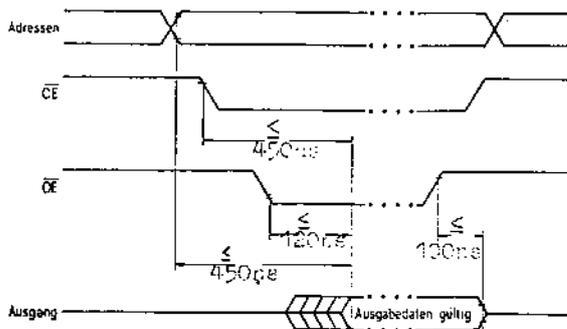


werden. Die Grenzzeiten aufeinander folgender Signale sind im Zeitdiagramm eingetragen. Das Löschen des EPROM's erfolgt wie von den Typen 2708/2716 her bekannt.

### Betriebsart "Lesen" des 2732

Im Zeitdiagramm sind die Bedingungen für die Chip-Enable- und Output-Enable-Eingänge (CE und OE) sowie max. Verzögerungszeiten für stabile Daten am Ausgang eingetragen, im Grunde nichts Neues.

(Fortsetzung im 80-Bus-Journal 1/83)



```

1 REM *** LOST DUTCHMAN'S GOLD *** 31.10.82
2 REM
3 REM COPYRIGHT 1980 BY TERI LI.
4 REM AUS BYTE DEZEMBER 1980
5 REM
6 REM BEARBEITET FÜR NASCOM VOM MICHAEL BACH
7 REM
8 REM
9 CLEAR 1000: DEF FNG(I)=-I*(I+1)
10 CLS: SCREEN 10,16
11 PRINT "*** LOST DUTCHMAN'S GOLD ***"
12 SCREEN 1,4
13 PRINT "Welcome to the search for the lost";
14 PRINT "Dutchman's"
15 PRINT "gold. I'm the ghost of back-pack Sam.";
16 PRINT "yer guide": PRINT "on this adventure.";
17 PRINT "Ya can use two-word commands to find ";
18 PRINT "clues and": PRINT "move about.";
19 PRINT "GET, GO, PUT, DROP, SCORE, INVENTORY ";
20 PRINT "n bout": PRINT "100 more words make ";
21 PRINT "sense to me."
22 PRINT "Ask fer HELP and '?' if ya get stumped."
23 PRINT "I hope ya don't end up a ghost, like me!"
24 PRINT "GOOD LUCK!": PRINT
25 DIM N$(26,7),O$(32,3),R$(46),SA(32,3)
26 B=0: IN=B: LN=50: LM=LN: XT=B: X2=B: T$="": M1$=T$
27 M2$=T$: OMQ$=T$: U=B: U1=B: IM=2: BD=21: L=L-1
28 C$=CHR$(13)
29 FOR J=B TO 26: FOR I=B TO 7: READ N$(J,I)
30 NEXT I: NEXT J
31 FOR I=B TO 32: READ O$(I,0),O$(I,1),O$(I,2)
32 O$(I,3)="": NEXT
33 FOR I=B TO 46: READ R$(I): NEXT
34 INPUT "Aites Spiel weiterspielen": QM$
35 IF LEFT$(QM$,1)="Y" GOTO 74
36 IF LEFT$(QM$,2)="Y" GOTO 100
37 PRINT "Kassettenrekorder auf Wiedergabe stellen."
38 LOAD*SA
39 PRINT "Fertig, Kassettenrekorder abstellen."
40 FOR I=0 TO 32
41 FOR J=1 TO 3: O$=STR$(SA(I,J)): O$(I,J)=O$
42 IF LEFT$(O$,1)=" " THEN O$=RIGHT$(O$,LEN(O$)-1)
43 O$(I,J)=O$
44 NEXT J: NEXT I
45 L=SA(0,0): CL=SA(1,0): LN=SA(2,0)
46 IN=SA(3,0): LM=SA(4,0): U=SA(5,0)
47 U1=SA(6,0): IM=SA(7,0)
48 IF VAL(O$(12,2))=L AND CL=0 GOTO 150
49 IF O$(12,3)="1" AND CL=0 GOTO 150
50 IF L(15) GOTO 150
51 PRINT "It's too dark ta see!"
52 IF L(15) OR O$(6,3)="1" OR O$(4,3)="" GOTO 110
53 PRINT "Ya hear a strange noise!"
54 GOTO 350
55 T$="OK": RETURN
56 B=1: PRINT
57 PRINT "Ya see: ";: RETURN
58 IF CL=1 THEN LN=LN-1: IF LN(1) THEN CL=0
59 L1=L
60 B=0000

```

```

172 PRINT: PRINT "YER "N$(L,0)";"
173 FOR J=0 TO 32: IF VAL(O$(J,2)){}L GOTO 240
174 IF B=0 THEN GOSUB 140
175 IF O$(J,3)="-7" OR O$(J,3)="-2" THEN PRINT "burnt";
176 IF J() OR VAL(O$(1,2)){}L THEN 200
177 IF O$(1,3)="" THEN PRINT "untied ";: GOTO 200
178 IF O$(1,3)="-9" THEN PRINT "dead ";
179 IF O$(J,3)="-3" THEN PRINT "smashed ";
180 IF L(26) OR J()32 THEN 220
181 IF U() THEN PRINT "un";
182 PRINT "locked ";
183 IF CL=1 AND J=12 THEN PRINT "lit ";
184 PRINT O$(J,0)";";
185 NEXT J
186 IF U1=1 AND L=1 THEN PRINT "Open TRAP DOOR. ";
187 IF N$(L,1)="" THEN PRINT: GOTO 330
188 IF B=0 THEN GOSUB 140
189 IF L()16 THEN 280
190 IF U() THEN PRINT "un";
191 PRINT "locked ";: GOTO 290
192 IF L()17 THEN 290
193 IF U=1 THEN PRINT "unlocked ";: GOTO 290
194 PRINT "blocked ";
195 PRINT N$(L,1)";"
196 IF L=17 AND U=0 THEN N$(L,2)=" "+RIGHT$(N$(L,2),3)
197 IF L=16 AND U=1 THEN N$(L,2)="E"+RIGHT$(N$(L,2),3)
198 PRINT: IF N$(L,2)="" THEN 350
199 PRINT "Obvious exits: ";
200 FOR I=1 TO 4: P$=MID$(N$(L,2),I,1)
201 IF P$="N" THEN PRINT "North ";
202 IF P$="S" THEN PRINT "South ";
203 IF P$="E" THEN PRINT "East ";
204 IF P$="W" THEN PRINT "West ";
205 NEXT I: PRINT
206 B=0
207 PRINT "=====
208 IF L()9 OR O$(22,2)(){}1 GOTO 370
209 PRINT "Thar's noise up ahead, ";
210 PRINT "sounds like inJuns."
211 IF L=9 AND O$(22,2)="" THEN O$(22,2)="11"
212 INPUT " -- Now what, QM$
213 IF LN=10 OR CL()1 GOTO 415
214 PRINT "Yer runnin' low on kerosene."
215 IF QM$="SAVE" THEN 2520
216 IF QM$="SCORE" THEN 450
217 IF L=6 THEN 426
218 T$="Ya get nothin' for bein' here": GOTO 1900
219 Z=0: FOR I=14 TO 17
220 IF VAL(O$(I,2))=6 THEN Z=Z+1
221 NEXT
222 PRINT "Ya got 'z' treasures, tot'lin' Z/4*100%"
223 IF Z=4 THEN PRINT "Ya made it!": GOTO 2220
224 T$="Ya missed 'sum treasure!": GOTO 1900
225 IF CL()10 OR L()15 OR O$(6,3)(){}1 GOTO 460
226 PRINT "Ya gadi": PRINT "Ya been bit by a rattler."
227 PRINT "Yer dead.": GOTO 2220
228 IF L()17 OR U()1 OR RND(1)*9+1)=2 GOTO 480
229 PRINT "&#188 CAVE IN !!?;+*#"
230 PRINT "The iron door's blocked, ";
231 PRINT "Yer trapped!": U=0

```

```

480 IF QM$="" THEN 2600
481 X=VAL(N$(L,3))
482 X=VAL(N$(L,3))
484 ON X GOSUB 2300,2310,2320,2330,2350,2370
485 ON FNG(X-6) GOSUB 2380,2390
486 GOTO 100
490 X1=LEN(QM$)
491 IF X1{3 THEN T$="What?":GOTO 1900
493 X3=0:FOR X2=3 TO X1:M1$=MID$(QM$,X2,1)
494 IF M1$="" THEN X3=X2: X2=X1
495 NEXT X2: X2=X3:IF X3=0 THEN X2=X1
500 P$=LEFT$(QM$,3)
501 IF X2{X1 AND X2+3}X1 THEN 610
502 I1=-1:FOR I=0 TO 42:IF P$=R$(I) THEN I1=I
503 NEXT I:IF I1=-1 THEN I=I:GOTO 520
510 IF X1=X2 THEN T$="What?": GOTO 1900
511 T$="Sorry, but ya can't "+LEFT$(QM$,X2)
512 IF X2{X1 THEN T$=T$+RIGHT$(QM$,X1-X2)
513 T$=T$+ ".":GOTO 1900
520 IF I1=8 OR X1{X2 GOTO 540
521 W=I
522 ON W+1 GOSUB 2010,2010,2010,2020,680,2020
524 ON FNG(W-5) GOSUB 1920,2000
526 GOTO 1900
540 JJ=-1:M1$=MID$(QM$,X2+1,3)
541 FOR J=0 TO 32:IF M1$=LEFT$(Q$(J,0),3) THEN JJ=J
542 NEXT J:J=JJ:IF JJ=-1 THEN 630
543 IF M1$="IRO" THEN 580
545 IF I=36 THEN 940
550 IF W(L,1)=" THEN 560
551 Z=I: A=Z: X1=LEN(N$(L,1)): FOR X2=5 TO X1
552 IF MID$(N$(L,1),X2,1)=" OR X2=X1 GOTO 554
553 NEXT: GOTO 560
554 M2$=MID$(N$(L,1),Z,3): IF M2$=M1$ THEN 556
555 Z=X2+2:A=A+1: NEXT: GOTO 560
556 X2=X1: NEXT: B=A
557 IF I=10 THEN PRINT "Ya see nothin' special.":GOTO 160
560 FOR X=43 TO 46
561 IF M1$=R$(X) THEN X=46: NEXT: GOTO 650
565 NEXT
570 IF Q$(4,3){} AND Q$(6,3){} GOTO 580
572 IF I{23 OR M1$()} "SEL" GOTO 580
574 PRINT "Ok":PRINT "Yer dead": GOTO 2220
580 IF I=32 THEN GOSUB 780: GOTO 1900
590 IF I=33 THEN GOSUB 860: GOTO 1900
600 T$="I can't tell watcha want.": GOTO 1900
610 T$="I must be dumb, ya don't make sense"
612 GOTO 1900
630 IF I{9 THEN 650
633 IF J=33 OR J{=-1 GOTO 636
634 B=VAL(Q$(J,2))
635 IF B(L AND B) O THEN 1890
636 IF J}32 THEN J=29
640 W=I
642 ON FNG(W-8)GOSUB 1250,1390,1390,1250,1700
643 ON FNG(W-14)GOSUB 1700,1700,1820,1820,1250
644 ON FNG(W-20)GOSUB 700,700,1080,1060,1060,920
645 ON FNG(W-26)GOSUB 1150,1190,1210,730,730,780
646 ON FNG(W-32)GOSUB 860,1250,900,940,1700,1690
647 ON FNG(W-38)GOSUB 1320,1340,1820,2500
650 IF I}=9 GOTO 660
651 W=I
652 ON W+1 GOSUB 970,970,2220,680,970
654 ON FNG(W-5) GOSUB 1920,2000,1320
660 GOTO 1900
680 IF Q$(3,3){} THEN 1890
681 PRINT "Ok": FOR I=13 TO 17
682 IF VAL(Q$(I,2)){}-L GOTO 685
683 T$="Eureka! We struck gold!?!?"
684 Q$(1,2)=STR$(L):RETURN
685 NEXT
686 T$="Dag nab it! Thar's nothin' here!":RETURN
700 IF J{23 THEN 1060
701 IF L=14 THEN Q$(23,2)="15": GOTO 120
702 IF L=15 THEN Q$(23,2)="16": GOTO 120
703 IF L=16 AND U=1 THEN Q$(23,2)="17": GOTO 120
704 IF L=17 THEN Q$(23,2)="19": GOTO 120
705 IF L=19 THEN Q$(23,2)="23": Q$(23,3)="-3"
710 GOTO 120
730 IF J=3 OR J=4 OR J=6 OR J=8 GOTO 740
731 IF J=9 OR J=11 OR J=12 OR J=13 GOTO 740
732 IF J=19 OR J=27 OR J=28 GOTO 740
735 GOTO 1050
740 IF Q$(J,3)="1" THEN IN=IN-1
750 Q$(J,3)="-3":Q$(J,2)=STR$(L):GOTO 120
770 T$="Don't havta.":RETURN
780 IFL{16 THEN 800
783 IF U{0 THEN 770
786 IF Q$(10,3)="1" THEN U=1:GOTO 120
788 GOTO 1890
800 IF L{26 THEN 820
803 IF U{0 THEN 770
806 IF Q$(10,3)="1" THEN U}=1: RETURN
808 GOTO 1890
820 IF Q$(1,3){} THEN 840
825 IF Q$(0,3){} "2" GOTO 830
826 Q$(0,3)="" :Q$(0,2)=STR$(L):GOTO 120
830 GOTO 1050
840 IF CL=1 AND Q$(12,3)="1" THEN CL=0: GOTO 120
860 IF (L=16 OR L=17) AND U=1 THEN U=0:GOTO 120
870 IF L=26 AND U}=1 THEN U}=0: GOTO 120
880 GOTO 1050
900 T$="ya hear nothin' speacial.": RETURN
920 IF L{11 OR Q$(9,3){} THEN 1320
930 PRINT "They took the fire water 'n ran."
931 IN=IN-1:Q$(9,2)="" :Q$(9,3)="" :Q$(22,2)=""
932 RETURN
940 IF M1${"YOH" GOTO 950
941 PRINT "Sorry, yer in the wrong adventure."
942 T$="Turkey!":GOTO 1900
950 GOSUB 120
951 T$=T$+C$+RIGHT$(QM$,X1-X2)+ ". "
952 GOTO 1900
970 IF B=0 THEN 980
971 L=VAL(N$(L,B+3))
972 IF Q$(1,3){} THEN GOTO 976
973 IF NOT(L=1 OR L=6 OR L=15)GOTO 976
974 T$="He won't go":L=L:RETURN
976 IF L=17 AND U=0 THEN L=L: GOTO 1890

```

```

480 IF QM$="" THEN 2600
481 X=VAL(N$(L,3))
482 X=VAL(N$(L,3))
484 ON X GOSUB 2300,2310,2320,2330,2350,2370
485 ON FNG(X-6) GOSUB 2380,2390
486 GOTO 100
490 X1=LEN(QM$)
491 IF X1{3 THEN T$="What?":GOTO 1900
493 X3=0:FOR X2=3 TO X1:M1$=MID$(QM$,X2,1)
494 IF M1$="" THEN X3=X2: X2=X1
495 NEXT X2: X2=X3:IF X3=0 THEN X2=X1
500 P$=LEFT$(QM$,3)
501 IF X2{X1 AND X2+3}X1 THEN 610
502 I1=-1:FOR I=0 TO 42:IF P$=R$(I) THEN I1=I
503 NEXT I:IF I1=-1 THEN I=I:GOTO 520
510 IF X1=X2 THEN T$="What?": GOTO 1900
511 T$="Sorry, but ya can't "+LEFT$(QM$,X2)
512 IF X2{X1 THEN T$=T$+RIGHT$(QM$,X1-X2)
513 T$=T$+ ".":GOTO 1900
520 IF I1=8 OR X1{X2 GOTO 540
521 W=I
522 ON W+1 GOSUB 2010,2010,2010,2020,680,2020
524 ON FNG(W-5) GOSUB 1920,2000
526 GOTO 1900
540 JJ=-1:M1$=MID$(QM$,X2+1,3)
541 FOR J=0 TO 32:IF M1$=LEFT$(Q$(J,0),3) THEN JJ=J
542 NEXT J:J=JJ:IF JJ=-1 THEN 630
543 IF M1$="IRO" THEN 580
545 IF I=36 THEN 940
550 IF W(L,1)=" THEN 560
551 Z=I: A=Z: X1=LEN(N$(L,1)): FOR X2=5 TO X1
552 IF MID$(N$(L,1),X2,1)=" OR X2=X1 GOTO 554
553 NEXT: GOTO 560
554 M2$=MID$(N$(L,1),Z,3): IF M2$=M1$ THEN 556
555 Z=X2+2:A=A+1: NEXT: GOTO 560
556 X2=X1: NEXT: B=A
557 IF I=10 THEN PRINT "Ya see nothin' special.":GOTO 160
560 FOR X=43 TO 46
561 IF M1$=R$(X) THEN X=46: NEXT: GOTO 650
565 NEXT
570 IF Q$(4,3){} AND Q$(6,3){} GOTO 580
572 IF I{23 OR M1$()} "SEL" GOTO 580
574 PRINT "Ok":PRINT "Yer dead": GOTO 2220
580 IF I=32 THEN GOSUB 780: GOTO 1900
590 IF I=33 THEN GOSUB 860: GOTO 1900
600 T$="I can't tell watcha want.": GOTO 1900
610 T$="I must be dumb, ya don't make sense"
612 GOTO 1900
630 IF I{9 THEN 650
633 IF J=33 OR J{=-1 GOTO 636
634 B=VAL(Q$(J,2))
635 IF B(L AND B) O THEN 1890
636 IF J}32 THEN J=29
640 W=I
642 ON FNG(W-8)GOSUB 1250,1390,1390,1250,1700
643 ON FNG(W-14)GOSUB 1700,1700,1820,1820,1250
644 ON FNG(W-20)GOSUB 700,700,1080,1060,1060,920
645 ON FNG(W-26)GOSUB 1150,1190,1210,730,730,780
646 ON FNG(W-32)GOSUB 860,1250,900,940,1700,1690

```

```

978 IF L1=26 AND L=1 AND U1=0 THEN L=L1:GOTO1890
979 GOTO 1060
980 IF U1=1 AND L=1 THEN L=26: GOTO 120
990 IF U1=1 AND M1$="TRA" AND L=26 THEN L=1:RETURN
1000 FOR X1=1 TO 4:M2$=M1$(X1,2),X1,1)
1001 IF M2$(LEFT$(M1$,1) THEN NEXT X1:GOTO 1010
1003 FOR X2=43 TO 46
1004 IF M1$(IR$(X2) THEN NEXT X2:GOTO 1010
1006 L=VAL(M$(L,X1+3)):IF L() THEN 1060
1010 IF L=23 THEN T$="It's too slippery.":RETURN
1020 IF L=25 THEN L=26:RETURN
1030 IF L=26 THEN L=25:RETURN
1040 IF L=19 THEN L=23:PRINT"High broke m'neck!":RETURN
1050 T$="Sorry, butcha can't do that.":RETURN
1060 B=0:GOTO 120
1080 IF O$(4,3){}1" AND O$(6,3){}1" THEN 1320
1083 IF J() THEN 1110
1086 Z=0:IF O$(J,3)="1" THEN Z=99
1090 O$(J,3)="9":O$(1,2)=STR$(L)
1100 IF Z=99 THEN IN=IN-1
1110 IF J()22 GOTO 1120
1111 PRINT"Ya got one!":PRINT"But the rest got you!"
1112 GOTO 2220
1120 IF J=9 THEN T$="Ya hit me!":RETURN
1130 GOTO 120
1150 IF O$(18,3){}1" THEN 1890
1152 IF J() THEN 511
1153 IF J=4 OR (J)5 AND J(11) GOTO 1050
1154 IF J()14 AND J(18) GOTO 1050
1155 IF J=22 OR J=29 OR J()30 THEN 1050
1156 IF O$(J,3)="1" THEN IN=IN-1
1157 O$(J,3)="7":O$(J,2)=STR$(L)
1158 IF J() THEN 1060
1160 IF O$(0,3){}1" THEN 1060
1163 O$(0,2)=STR$(L):FOR J=5 TO 21
1164 IF O$(J,2)="1" THEN O$(J,3)="7"
1170 NEXT: GOTO 1060
1190 IF J()12 THEN 1150
1193 IF O$(J,3){}1" OR O$(18,3){}1" THEN 1150
1195 CL=1: GOTO 120
1210 IF J()13 GOTO 1220
1212 PRINT"Yeeeh!!": PRINT"It's kerosene!"
1214 PRINT"Ya just poisoned yourself.":GOTO 2220
1220 IF J()9 THEN 1050
1230 PRINT"Whewee!!"
1232 PRINT"Ya got plastered and lost a day."
1234 IF CL=1 THEN LN=LN-10
1235 RETURN
1250 IF J)32 OR J()0 THEN 1260
1253 B=VAL(O$(J,2))
1254 IF O$(J,3)="2" OR B=L OR B=-1 THEN 1260
1255 IF J()22 THEN 1890
1257 T$="It ain't here.": RETURN
1260 IF J(=21 AND J()1 AND J()-1 GOTO 1270
1262 T$="Who the heck ya think ya is?"+C$
1264 T$=T$+"Paul Bunyon??! Ya ain't strong 'nuf."
1266 RETURN
1270 IF O$(J,3){}1" THEN 7" GOTO 1280
1272 T$="It's burnt up 'n ruined.": RETURN
1280 IF IN(=4 GOTO 1283

```

```

1281 T$="Ya can't! Yer hands 'r full": RETURN
1283 IF O$(J,3){}1" GOTO 1286
1284 PRINT"Ya already got it!":RETURN
1286 IF VAL(O$(J,1))=0 AND I()16 THEN 1290
1287 IN=IN+1:IF O$(J,2){}1" THEN 1289
1288 IM=IM-1:O$(J,2)="":O$(J,3)="1":GOTO 1290
1289 O$(J,3)="1":O$(J,2)="
1290 IF J()9 AND J()11 AND J()12 THEN 120
1293 IF J=9 AND O$(8,2)="6" THEN O$(8,2)="6"
1295 IF J=11 AND O$(12,2)="8" THEN O$(12,2)="8"
1297 IF J=2 AND O$(18,2)="14" THEN O$(18,2)="14"
1300 RETURN
1320 IF O$(J,3)="1" GOTO 1325
1322 T$="Ya don't have it!":RETURN
1325 T$="Ok":IN=IN-1:O$(J,2)=STR$(L):O$(J,3)="
1326 RETURN
1340 IF O$(5,3){}1" THEN 1890
1350 IF L=7 THEN L=8:GOTO 120
1360 IF L=12 THEN L=13
1370 GOTO 120
1390 GOSUB 120
1391 ON J+1 GOTO 1400,1430,1440,1460,1470
1392 ON ENG(J-4) GOTO 1480,1490,1460,1460,1500
1393 ON FNG(J-9) GOTO 1510,1520,1530,1540,1460
1394 ON FNG(J-14) GOTO 1460,1460,1460,1460,1560
1395 ON FNG(J-19) GOTO 1570,1580,1590,1600,1610
1396 ON FNG(J-24) GOTO 1620,1460,1630,1460,1460
1397 ON FNG(J-29) GOTO 1460,1640,1460
1399 GOTO 1460
1400 IF VAL(O$(0,3))=1 GOTO 1410
1402 T$="They's lumpy ole leather saddlebags."
1404 RETURN
1410 PRINT"They hold ";
1412 IF IM=0 THEN PRINT"nothin'":T$="":RETURN
1415 FOR I=6 TO 22
1416 IF O$(I,2)="1" THEN PRINT " O$(I,0)";
1420 NEXT I:T$=CHR$(8)+ "+C$:RETURN
1440 T$="Ya see a weary old grey mule.":RETURN
1442 IF O$(18,2)="14" THEN O$(18,2)="14"
1450 RETURN
1460 T$="Ya see nothin' special.":RETURN
1470 T$="It's an old winchester single-shot."
1472 RETURN
1480 T$="Looks like a map ta th' lost "
1481 T$=T$+"Dutchman's Mine! Thar's a sketch "
1482 T$=T$+"of sum boulders.": RETURN
1490 T$="Nothin' special. "+C$
1492 T$=T$+"Jus' a loaded six shooter.":RETURN
1500 IF O$(8,2){}1" THEN 6" GOTO 1505
1501 T$="Looks like sumthin's 'hind the bottles."
1502 O$(18,2)="6": RETURN
1505 GOTO 1460
1510 T$="Thar'r 3 keys, tied with a leather "
1512 T$=T$+"strap.": RETURN
1520 IF O$(12,2){}1" THEN 8" GOTO 1460
1521 T$="Looks like thar's glass under 'um."
1522 O$(12,2)="8": RETURN
1530 T$="It's an ole kerosene lamp. "+C$
1532 T$=T$+"Full ta the "+STR$(LN))

```

```

1534 T$=I$+" turns left' mark.": RETURN
1540 IF I=11 THEN T$="It sez 'KEROSENE'": RETURN
1550 T$="Thar's writing on it.": RETURN
1560 IF O$(20,2){}-17" GOTO 1460
1562 O$(20,2)=17":T$="It's a message !": RETURN
1570 IF I{11} GOTO 1550
1572 T$="It sez: "+C$+"Bring treasures to "
1574 T$=T$+"sailoon, say 'SCORE!": RETURN
1580 IF I{11} GOTO 1550
1582 T$="It sez: "+C$+"Watch for other Rider "
1584 T$=T$+"Fantasy adventures!": RETURN
1590 T$="They's a savage lookin' band, "+C$
1592 T$=T$+"n they seen ya!": RETURN
1600 T$="Jus' an ole ore cart full o' rocks.":
1602 RETURN
1610 IF O$(5,2){}-25" GOTO 1460
1612 O$(5,2)="1":T$="Thar's sumthin' here!":
1613 RETURN
1620 IF I{11} GOTO 1550
1622 T$="It sez: "+C$+"WELCOME TO FRONTIERTOWN."
1623 RETURN
1630 IF O$(10,2){}-6" GOTO 1460
1632 T$="Thar's a set of keys thar!"
1633 O$(10,2)="6": RETURN
1640 B=INT(RND(1)*3+1)
1642 ON B GOTO 1650,1660,1670
1650 T$="It's a barrel cactus.": RETURN
1660 T$="It's a cholla cactus.": RETURN
1670 T$="It's a saguaro cactus.": RETURN
1690 IF J=4 AND I=38 THEN 1780
1700 IF J=1 AND VAL (O$(1,3))=0 GOTO 1710
1702 T$="Sorry, b' that ain't possible.": RETURN
1710 IF I{38} THEN 1720
1715 IF O$(1,3){}="" GOTO 1780
1716 T$="Tendarfoot! Ya hav'ta lead 'im first."
1717 RETURN
1720 IF O$(1,3)="1" THEN 1320
1723 IF O$(7,3){}="1" OR I=16 GOTO 1730
1724 T$="Try 'LEAD'": RETURN
1730 ON I-13 GOTO 1740,1750,1760,1770
1740 T$="He got away from ya.": RETURN
1750 T$="Ye'r too slow, he got away.": RETURN
1760 IF O$(7,3)=1" GOTO 1763
1761 T$="Bad luck, ya tripped and he run off."
1762 O$(1,3)="": O$(1,2)=STR$(L): RETURN
1763 IF O$(1,3)=" THEN 1280
1766 GOTO 1320
1770 T$="He bucked ya off.": RETURN
1780 INPUT"with what";Q$; IF J=4 THEN 1800
1782 IF LEFT$(Q$+"",3)="SAD" GOTO 1790
1783 T$="It fell off. "+C$+"He got away."
1784 O$(1,3)="": IN=IN-1:O$(1,2)=STR$(L): RETURN
1790 IF O$(0,3){}="1" GOTO 1320
1791 O$(0,3)=2": IN=IN-1: GOTO 1060
1800 IF LEFT$(Q$+"",3){}="BUL" THEN 1050
1805 IF O$(8,3){}="1" OR O$(4,3){}="1" THEN 1320
1810 GOTO 120
1820 IF O$(J,3){}="1" THEN 1320
1822 INPUT"on or in what (ie 'ON TABLE')";Q$
1823 P$=LEFT$(Q$+"",2)

```

```

1824 IF P$(())"IN"AND P$(())"ON"THEN T$="Huh?": RETURN
1830 M1$=MID$(Q$+,4,3)
1840 IF P$(())"ON" THEN 1850
1841 IF M1$="MUL" THEN 1790
1843 IF J{13} THEN 1320
1845 IF LM=0 THEN T$="It's empty.": RETURN
1847 LM=0: GOTO 1320
1850 IF M1$(J)"SAD" THEN 1860
1852 IF J}=5 AND J{19 AND J{22} GOTO 1855
1853 T$="Sorry, it don't fit.": RETURN
1855 IF IM}4 THEN T$="The bags 'r full.": RETURN
1857 O$(J,2)="-1": O$(J,3)="": IN=IN-1
1858 IM=IM+1: T$="OK": RETURN
1860 IF M1$="MUL" THEN T$="Ya got kicked!": RETURN
1870 IF M1$(J)"LAN" THEN 1050
1872 IF O$(13,3){}="1" OR O$(12,3){}="1" THEN 1320
1875 IF LM=50 THEN LM=LN+50: LM=0: GOTO 120
1880 T$="Jar's empty.": RETURN
1890 T$="Ya can't do that... yet!"
1900 PRINT T$: T$="": GOTO 100
1920 PRINT"Ya got with ya.":
1922 IF IN{0} OR O$(1,3)="1" GOTO 1930
1924 T$="Nothin'": RETURN
1930 B=VAL(O$(0,3)):X1=B: IF B{1} THEN 1940
1933 PRINT:PRINT O$(0,0)", containing: ";:FOR I=1 TO 22
1934 IF O$(1,2)="-1" THEN PRINT O$(I,0). ";:X1=2
1936 NEXT: IF X1{2} THEN PRINT"Nothin'! ";
1940 IF O$(1,3){}="1" GOTO 1950
1942 PRINT:PRINT"The mule, which yer leadin. ";
1944 IF B=2 THEN PRINT "(carrying saddlebags) ";
1950 PRINT: X1=0: FOR I=2 TO 22
1952 IF VAL (O$(I,3))=1 THEN 1970
1955 IF CL=1 AND I=12 THEN PRINT "lit ";
1960 PRINT O$(1,0). ";
1970 NEXT
1980 T$="": RETURN
2000 IF L{19} GOTO 2005
2001 CLS:PRINT"Ya fell 100 feet 'n landed on rocks."
2002 PRINT"Yer dead.": GOTO 2220
2005 IF J=230R J=320R M1$="p" THEN T$="OK": RETURN
2006 GOTO 1050
2010 T$="Where?": RETURN
2020 IF L=25 THEN L=26: GOTO 120
2030 IF L=26 THEN L=25
2040 GOTO 120
2100 DATA" ", " ", " ", "0,0,0,0,0
2101 DATAIN A MINER'S SHACK
2102 DATAWINDOW, DOOR," ",1,3,3,0,0
2105 DATAIN A DESERT,ROAD, MOUNTAINS, DESERT.
2106 DATA"WNES",2,4,2,2,2
2107 DATAON A DIRT PATH
2108 DATAMINER'S SHACK, ROAD, MOUNTAINS
2109 DATA" W ",5,1,4,2,2
2110 DATAON A DIRT ROAD
2111 DATAMOUNTAINS, PATH, DESERT, TOWN
2112 DATA"NE S",2,7,3,2,5
2113 DATAIN A GHOST TOWN
2115 DATASALOON," N ",1,6,4,0,0

```

```

2116 DATAIN A SALOON," "W " ,1,5,0,0,0
2120 DATAAT THE SUPERSTITION MOUNTAINS
2121 DATAAT ROAD. DESERT."S N " ,4,4,2,2,2
2122 DATAAT WEAVER'S NEEDLE," "NS " ,1,9,7,0,0
2130 DATAIN A NARROW DEFILE,BUSHES. CAVES
2131 DATA " NS" ,1,11,10,12,8
2132 DATAIN A SMALL CAVE
2133 DATA " "E " ,1,9,0,0,0
2134 DATABEHIND A BUSH,BUSH, "W " ,1,9,0,0,0
2135 DATAIN A BOX CANYON
2136 DATABUSHES. TREES. BOULDERS," " S "
2137 DATA4,12,12,12,9
2140 DATAIN FRONT OF A HIDDEN MINE
2141 DATAMINE SHAFT,"ESNW"
2142 DATA8,14,12,12,12
2143 DATAIN THE MOUTH OF A DIM MINE
2144 DATAMINE SHAFT,"EW "
2145 DATA1,15,13,0,0
2146 DATAIN A MINE,DARK TUNNEL,"EW "
2147 DATA6,16,14,0,0
2150 DATAAT THE END OF A TUNNEL,IRON DOOR
2151 DATA " W " ,1,17,15,0,0
2152 DATAIN A LARGE CHAMBER,IRON DOOR
2153 DATA"MNES" ,1,16,19,18,20
2154 DATAIN A MAZE OF TUNNELS," " ,"NSEW"
2155 DATA6,18,18,18,18
2156 DATAIN FRONT OF A PIT,DARK HOLE," S "
2157 DATA7,23,17,0,0
2160 DATAAT AN INTERSECTION," "
2162 DATA " SNW" ,6,0,22,17,21
2163 DATAIN A DEAD END TUNNEL," "
2164 DATA "E " ,8,20,0,0,0
2165 DATAIN A BLOCKED TUNNEL," "
2166 DATA "N " ,8,20,0,0,0
2167 DATAAT THE BOTTOM OF A SHAFT,WALL
2168 DATA "S " ,8,24,0,0,0
2170 DATAIN A TUNNEL
2171 DATA " "NS " ,6,23,25,0,0
2172 DATAAT THE END OF A TUNNEL
2173 DATALADDER," N " ,6,26,24,0,0
2174 DATAAT THE TOP OF A LADDER
2175 DATA " " ,6,1,25,0,0
2180 DATASADDLEBAGS,2,1,MULE,0,3
2181 DATABURLAP SACK,4,14,SHOVEL,4,1
2182 DATARIFLE,4,1,MAP,1,-25
2183 DATAGUN,1,-1,CARROTS,1,-1
2184 DATABOX OF RIFLE BULLETS,1,-6
2185 DATAWHISKEY BOTTLES,2,6
2186 DATAKEYS,1,-6
2187 DATAPILE OF BONES (MINE),2,8
2190 DATALANTERN,2,-8,JAR OF LIQUID,1,-11
2192 DATA*SPANISH COINS*,1,-10
2193 DATA*TOURNOUISE*,1,-23
2194 DATA$SILVER$,1,-21
2195 DATA$GOLD$,1,-22
2196 DATAMATCHES,1,-14,CRATES,3,17
2197 DATANOTE,1,-17
2198 DATAPAPER,1,6
2199 DATAINDIANS,0,11
2200 DATAORE CART,0,14,BED,0,1

```

```

2201 DATASIGN,0,5,BROKEN GLASS,0,6
2202 DATATABLES,0,6,CHAIRS,0,6
2203 DATAROCKS,0,10,WOODEN RAILS,0,15
2204 DATACACTUS,0,2,TRAP DOOR,0,26
2210 DATAGO ,ENT, RUN, QUI, DIG, CLI, INV, JUM
2211 DATADRO,GET,EXA,REA,LOO,MDV,CAT,CHA
2212 DATALEA,RID,PUT,PLA,PIC,PUS,PUL,SHO
2213 DATAOPE,CLO,GIV,BUR,LIG,DRI,BRE,HIT
2214 DATAUNL,LOC,TAK,LIS,SAY,FEE,LOA,UNT
2215 DATAFOL,POU,EAT,NOR,SOU,EAS,WES
2220 INPUT"do you want ta try again";QMS
2222 IF LEFT$(QMS+" ",1)="Y" THEN 10
2230 END
2300 PRINT"Try exam'nin thin's":RETURN
2310 PRINT"Roads go places.":RETURN
2320 PRINT"Maybe the trail goes sumwhar.":RETURN
2330 IF O$(5,3){}"1"THEN PRINT"Ya got a map?"
2335 RETURN
2340 PRINT"Try 'FOLLOW'":RETURN
2350 IF VAL(O$(1,2)){L GOTO 2300
2351 PRINT"The mule looks thin.": RETURN
2360 GOTO 2300
2370 PRINT"Keep goin'":RETURN
2380 PRINT"It's slipry, butcha might make it down
2382 RETURN
2390 PRINT"This here's a mine, ya know":RETURN
2500 IF J=7THEN T$="They taste purty good.":RETURN
2502 RETURN
2510 T$="Can't do that... weirdo!":RETURN
2520 REM ABSPEICHERN
2530 PRINT"Ist der Kassettenrekorde auf";
2532 INPUT "Aufnahme";QMS:QMS=LEFT$(QMS+" ",1)
2534 IF QMS{"J" AND QMS{"Y"} GOTO 2530
2540 FOR I=0 TO 32
2542 FOR J=1 TO 3:SA(I,J)=VAL(O$(I,J)):NEXT J
2544 NEXT I
2550 SA(0,0)=L:SA(1,0)=CL:SA(2,0)=LN
2552 SA(3,0)=IN:SA(4,0)=LM:SA(5,0)=U
2554 SA(6,0)=U1:SA(7,0)=IM
2560 CSAVE*SA
2570 PRINT"Fertig.": GOTO 390
2600 PRINT"Sum of the words I understand are:"
2610 PRINT" go, enter, run, quit, dig,";
2615 PRINT" climb, jump,"
2620 PRINT" inventory, drop, get, examine,";
2625 PRINT" read, look,"
2630 PRINT" move, catch, chase, lead, ride,";
2635 PRINT" put, place,"
2640 PRINT" push, pull, shoot, open, close,";
2645 PRINT" pick, give,"
2650 PRINT" burn, light, drink, break, hit,";
2655 PRINT" unlock, lock,"
2660 PRINT" take, listen, say, feed, load,";
2665 PRINT" untie, follow,"
2670 PRINT" pour, eat, score, help, ..."
2672 INPUT"Ok";QMS
2675 PRINT"Hope that helps you a bit."
2680 GOTO 100
OK

```

```

1510 IF DF)13 THEN PRINT "DF)13 ";:GOTO 1340
1520 IF WF-DF(2) THEN PRINT "WF-DF(2 ";:GOTO 1340
1530 IF X(0) GOTO 1550
1540 XX=0:XM$="" :.000000"-:IF DF=0 THEN XM$="0."
1550 RD=DF+XX:NL$=NL$+NL$:BL$=BL$+BL$
1560 IF RD)5 OR RD(0 OR XX)5 OR X=0 GOTO 1610
1570 MR=SGN(VAL(XM$))*ABS(VAL(XM$))+10:(-RD)/2)
1580 IF ABS(MR)(<1 GOTO 1600
1590 XM$="" :100000":XX=XX+1:GOTO 1610
1600 XM$=STR$(MR)
1610 IF XX)0 GOTO 1680
1620 XF$=LEFT$(BL$,WF-DF-2)+LEFT$(XM$,2)
1630 XF$=XF$+LEFT$(NL$,ABS(XX))+MID$(XM$,3,6)
1640 RD=LEN(XF$)
1650 IF RD(MF) THEN XF$=XF$+LEFT$(NL$,WF-RD)
1660 IF RD)WF THEN XF$=LEFT$(XF$,WF)
1670 GOTO 1770
1680 XV=1:IF X(0) THEN XV=2
1690 BL=WF-DF-XV-XX:IF BL(0) GOTO 1760
1700 XF$=LEFT$(BL$,BL):IF X(0) THEN XF$=XF$+"-"
1710 IF XX)5 GOTO 1740
1720 XF$=XF$+MID$(XM$,3,XX)+"."
1730 XF$=XF$+MID$(XM$,3+XX,6):GOTO 1640
1740 XF$=XF$+MID$(XM$,3,6)+LEFT$(NL$,XX-6)+"."
1750 XF$=XF$+LEFT$(NL$,DF):GOTO 1770
1760 XF$=LEFT$(OV$,WF-DF-1)+"."+LEFT$(OV$,DF)
1770 RETURN :REM 17.07.82 J. KWASNITZA
1780 REM ENDE F-FORMAT *****
OK

```

1 REM TV KAMERA INTERFACE

```

2 REM VERSION 1.0/3.9.82
3 REM G. BOEHM
8 CLS
10 OUT62,207 : REM CONTROLMODE
15 OUT62,192 : REM PORT A CONTR. BIT 6/7 IN
30 OUT60,32 : REM AUSGANGSPOSITION
40 FOR I=0 TO 47 : REM ZEILEN
50 OUT60,50 : REM ZEILENRUMMER *
60 OUT60,51 : REM *
70 OUT60,50 : REM *
80 OUT60,54 : REM ZEILE SPEICHERN **
85 IF (INP(60)AND64)=0 THEN85
90 OUT60,34 : REM ZAEHLER-FREIMACHEN ***
100 REM DARSTELLEN DER ZEILEN AUF SCHIRM
110 FOR J=0 TO 95
120 OUT60,42 : REM TAKT ****
130 OUT60,34 : REM *****
140 A=INP(60) AND 128
142 IF A=128 THEN GOTO 150:REM INVERT. A=0
143 Y=I: IF Y=2 THEN Y=Y+48
145 SET (J,Y-3) : REM WEISS SETZEN
150 NEXT J
160 OUT60,18:OUT60,34:NEXT I:REM CLEAR RAMZ.
170 GOTO 170
OK

```

```

10 REM TESTPROGRAMM NR. 1
20 FOR I=0 TO 10000 STEP 1000
30 PRINT I,SQR(I),I*I
40 NEXT I
50 STOP
60 REM TESTPROGRAMM NR. 2
70 CLEAR(300)
80 FOR I=0 TO 10000 STEP 1000
90 WF=9:DF=1:X=I:GOSUB1400:PRINT XF$:
100 WF=12:DF=4:X=SQR(I):GOSUB1400:PRINT XF$:
110 WE=14:DE=6:X=1*I:GOSUB1000:PRINT XE$
120 NEXT I
130 STOP
1000 REM *** SUBR. E-FORMAT (X NACH XE$) ***
1010 REM WE = FELDBREITE
1020 REM DE = ANZAHL DER STELLEN NACH DEM PUNKT
1030 REM BELEGTE VARIABLEN: BL,DE,EE,LX,MR,X,XE
1040 REM XE,XR,XX,BL$,XE$,XX$,XM$,X1$
1050 EE=0: BL$="" :NL$="0000000" :XR=X
1060 IF WE)15 THEN PRINT"WE)15 ";:GOTO 1340
1070 IF DE)6 THEN PRINT"DE)6 ";:GOTO 1340
1080 IF WE-DE)6 THEN PRINT"WE-DE(6 ";:GOTO 1340
1090 BL=WE-DE-5:IF X=0 GOTO 1320
1100 XE$=STR$(X):IF LEN(XE$)(4) GOTO 1130
1110 X1$=MID$(XE$,LEN(XE$)-3,1)
1120 IF X1$="E" GOTO 1140
1130 X=X*1E+10:EE=1:GOTO 1100
1140 X1$=MID$(XE$,3,1):IF X1$="" GOTO 1170
1150 XM$=LEFT$(XE$,1)+"." :+MID$(XE$,2,1)
1160 GOTO 1190
1170 XM$=LEFT$(XE$,1)+"." :+MID$(XE$,2,1)
1180 XM$=XM$+MID$(XE$,4,LEN(XE$)-7)
1190 IF DE)5 THEN MR=VAL(XM$):GOTO 1210
1200 MR=SGN(VAL(XM$))*ABS(VAL(XM$))+10:(-DE)/2)
1210 XX=VAL(RIGHT$(XE$,3))+1-EE*10
1220 IF ABS(MR)=1 THEN MR=SGN(MR)*.1:XX=XX+1
1230 XX$=STR$(XX):XT$=LEFT$(XX$,1)
1240 IF XT$="" GOTO 1260
1250 XX$="" :+RIGHT$(XX$,LEN(XX$)-1)
1260 IF LEN(XX$)=3 GOTO 1280
1270 XX$=LEFT$(XX$,1)+"0"+RIGHT$(XX$,1)
1280 XE$=STR$(MR)+NL$:XM$=LEFT$(XE$,DE+2)
1290 XF$=XM$+"E"+XX$
1300 IF LEN(XE$)=WE GOTO 1330
1310 XE$=LEFT$(BL$,WF-LEN(XE$))+XE$:GOTO 1330
1320 XE$=LEFT$(BL$,BL)+"." :+LEFT$(NL$,DE)+"E+00"
1330 X=XR:RETURN
1340 PRINT "UNZULAESSIG ":STOP
1350 REM ENDE E-FORMAT *****
1400 REM *** SUBR. F-FORMAT (X NACH XF$) ***
1410 REM WF = FELDBREITE
1420 REM DF = ANZAHL DER STELLEN NACH DEM PUNKT
1430 REM BELEGTE VARIABLEN: DF,RD,XV,Wf,OV$,XF$
1440 REM UMD WIE IN GOSUB1000
1450 OV$=""*****:WE=12:DE=6:GOSUB1050
1460 IF WF)15 THEN PRINT "WF)15 ";:GOTO 1340
1470 IF X(0 OR XX(=0) GOTO 1510
1480 IF DF)14 THEN PRINT "DF)14 ";:GOTO 1340
1490 IF WF-DF(1) THEN PRINT "WF-DF(1 ";:GOTO 1340
1500 GOTO 1530

```

```

10 REM -- DATALINE DEMO --
20 ;
30 REM *****
40 REM * Erzeugung von DATA Statements, die *
50 REM * die ersten 100 Bytes von NAS-SYS 3 *
60 REM * enthalten. A808-A885 Maschinenpro- *
61 REM * gramm.
70 REM *****
80 ;
90 CLS:PRINT** D A T A   L I N E   G E " ;
100 PRINT" N E R A T O R **:PRINT
110 PRINT" MEMSIZE SOLLTE AUF ETWA DIE " ;
120 PRINT" HAELFTE DES FREI"
130 PRINT" VERFUEGBAREN RAMS BEGRENZT SEIN. "
140 PRINT" Wenn diese zutrifft dann'OK' " ;
150 INPUT" eingeben";A$
160 IFA$="OK" THEN 180
170 MONITOR
180 PRINT" A C H T U N G : "
190 PRINT" SET und RESET ist aktiv !!!"
200 PRINT" BEGINN und ENDE des Speicher";
210 PRINT" bereiches, der"
220 PRINT" umgewandelt werden soll in Zeile";
230 PRINT" 320 einsetzen"
240 PRINT" und Programm mit RUN 280 starten"
250 GOSUB 430
260 DOKE4100, -22520:A=USR(0):END
270 ;
280 : DOKE4100, -22469:WIDTH49
290 LI=5000:REM *** STARTLINE ***
300 SET
310 REM * In naechste Zeile Bereich einsetzen
320 FL=0:FORX=0TO100STEP2
330 Y$=STR$(DEK(X))
340 ILEFT$(Y$,1)=" " THEN Y$=RIGHT$(Y$,LEN(Y$)-1)
350 DA=DA+LEN(Y$): IFFL=0 THEN FL=1:GOTO380
360 IF DA=38 THEN PRINTCHR$(8):GOTO380
370 PRINTY$ " ; : DA=DA+1:GOTO400
380 DA=LEN(Y$):PRINTCHR$(17)LI"DATA"Y$ " ;
390 LI=LI+2:REM *** LINE INCREMENT ***
400 NEXT:PRINTCHR$(8)
410 RESET:A=USR(0):END
420 ;
430 RESTORE460
440 FORX= -22520TO-22350STEP2
450 READI:DOKEX,I:NEXT:RETURN
460 DATA18721,8872,4181,23329,8872,4184
470 DATA17135,26229,27750,28261,15648,32
480 DATA33,-4696,-20645,4880,21229,-21043
490 DATA8097,-22379,31522,8460,-22415,30754
500 DATA-8436,8538,-192,21794,8464,-171
510 DATA22562,-8432,-6822,-20959,-8280,8817
520 DATA-22522,-20694,8976,546,-7768,-6711
530 DATA1578,-8280,8561,-22352,29407,1570

```

```

540 DATA10920,-22526,1058,6312,-2591,26079
550 DATA10993,-22526,9079,546,4520,-22528
560 DATA21229,10968,-22522,29151,3567,30018
570 DATA26214,20256,3414,-8448,10843,-22526
580 DATA-4635,1115,-4696,-7854,14462,10758
590 DATA-22522,29407,9161,546,-13912,117
600 DATA118,30208
610 ;
620 REM = AB HIER ENTSTEHEN PROGRAMMZEILEN =
630 ;
5000 DATA49,-10480,-15608,946,25311,6360,-15365
5002 DATA22,-7707,9185,-15387,1392,-7707,9185
5004 DATA-15387,1408,11235,-15389,1050,0,32483
5006 DATA-18653,1568,-13853,-15387,1882,6391
5008 DATA242,-14275,-3595,-1512,18351,-1,-1008
5010 DATA-6711,33,-20980,211,-11394,-7936,16073
5012 DATA-6896,33,-20980,6263,-2574,467,731
5014 DATA30667,-1496,-13839
OK
1081 REM NACHTRAG ZU HEFT 7/8-82
1082 REM "Entfernungs- und Winkelprogramm"
1083 REM von Wolfgang v. Jan
1084 REM aus unerfindlichen Gründen war
1085 REM dort das Listing nur bis Zeile
1086 REM 1080 abgedruckt -
1087 REM Hier folgt der Rest:
1090 IF PS)=PE GOTO1130:REM 4.QU
1100 WS=MS:RETURN:REM 1.QU
1110 WS=180-WS:RETURN:REM 2.QU
1120 WS=180+WS:RETURN:REM 3.QU
1130 WS=360-WS:RETURN:REM 4.QU
1140 PRINT" Nord":RETURN
1150 PRINT" Nordost":RETURN
1160 PRINT" Ost":RETURN
1170 PRINT" Suedost":RETURN
1180 PRINT" Sued":RETURN
1190 PRINT" Suedwest":RETURN
1200 PRINT" West":RETURN
1210 PRINT" Nordwest":RETURN
1220 PRINT" Nord":RETURN
1230 REM UPRO TABELLENAUSGABE 5/81 v. Jan --
1240 REM ZEILE, 1-15
1250 REM SPSPALTE, 1-48
1260 REM WEITERSPRUNG PRO TABELLENWEITE
1270 IF ZE(1) THEN ZE=1
1280 IF SP(1) THEN SP=1
1290 SCREEN SP,ZE:REM CURSOR GESETZT!
1300 SP=SP+WE:REM WEITERSPRUNG UM WE
1310 IF SP>(48-WE) THEN ZE=ZE+1:SP=1
1320 RETURN
OK

```

```

10 REM == STRINGSAVE V6 FILE A ==
20 REM 30.7.82 22h32
30 REM Memory C00 - E00
40 :
50 REM G.T.KLEMENT 'Teddysoft'
60 REM ██████████ WIEN
70 :
80 REM *****
90 REM * Es werden Stringarrays gesaved
100 REM * die in DIM Statements definiert
110 REM * wurden.Tape Format wie numerische
120 REM * Arrays.
130 REM * AUFRUF U=USR(AX$)S fuer SAVE
140 REM * U=USR(AX$L) fuer LOAD
150 REM * hier im Beispiel die Variable
160 REM * AX$.Multiple Statements moeglich
170 REM *****
180 :
190 GOSUB220
200 END
210 :
220 RESTORE240:F0RX=3264T03597STEP2:READY
230 DOKEX,Y:NEXT:D0KE4100,3264:RETURN
240 DATA-727,-10779,32299,10494,-1504,32291
250 DATA6706,8973,-386,8228,-20735,-32570,4914
260 DATA-12019,32481,-6877,-299,10323,-504
270 DATA-13748,3454,-21053,-8221,-6793,-12827
280 DATA3332,-12863,3373,-8251,-8353,-16035
290 DATA19661,-15603,3474,-10198,-4848,-9637
300 DATA-6896,21229,-3359,3365,-32706,9150
310 DATA7362,15885,-16831,24099,22051,-14301
320 DATA6169,-7710,-7711,29151,-16445,18147
330 DATA9080,9054,-10922,-1176,7714,-7924,3096
340 DATA19780,-2607,2331,31515,8370,-3590,8253
350 DATA-13839,-81,1286,28639,-11970,-1520
360 DATA19437,3102,2563,863,-6909,28426,2563
370 DATA-15001,-20669,31675,28639,28639,30410
380 DATA-8435,31085,28639,-7743,31787,8373
390 DATA-13859,30687,-8219,-6792,1229,-8435
400 DATA-12961,3373,-25907,-7923,29407,-8223
410 DATA-8335,-15521,-3854,-4635,7771,-12532
420 DATA-11778,-1248,774,-305,8401,4340,11001
430 DATA4291,20431,5071,4882,-18157,11808
440 DATA-18001,10792,-20665,17133,8747,4291
450 DATA-10973,-4635,23131,-4848,-7854,14545
460 DATA32039,31762,4627,16667,14,30671,20353
470 DATA4131,-12295,-15687,3572,-16109,-20725
480 DATA8377,-18430,-14904,-18152,205,-15602
490 DATA-2709,205,-15602,-7262,-7727,-7711
500 DATA29407,-8223,-8335,-10913,201
50K

```

```

10 REM = SWINGHOUSE FILE M =
20 REM 31.8.82 Memsize Enter 8193
30 REM AB .....Betrachtungsabstand
40 REM X0,Y0,Z0..Blickrichtung
50 REM X1,Y1,Z1..Augenpunkt
60 AB=250:PI=4*ATN(1)
70 NP=35:K=25:DIMP(NP).X4(K).Y4(K).Z4(K)
80 GOSUB690:FORI=0T023:P(I)=8192+1024*I:NEXT
90 FORI=0T011:P(I+24)=-32768+1024*I:NEXT
100 DEFFNI(X)=SGN(X)*(ABS(INT(X)+.5))
110 DEFFNX(X)=95/(2*AB)*(X+AB)
120 DEFFNY(Y)=(44/(2*AB))*(AB-Y))
130 DEFFNA(BI)=(B1*X1-A1*Y1)/SQR((A1*A1+B1*B1+C1*C1)*(X1*X1+Y1*Y1))
140 DEFFNB(MI)=(1+M1*M2+N1*N2)/SQR((1+M1*M1+N1*N1)*(1+M2*M2+N2*N2))
150 DEFFNAC(X)=-ATN(X/SQR(-X*X+1))+PI*.5
160 DEFFNAS(X)=ATN(X/SQR(-X*X+1))
170 DEFFNC(D)=ABS(AB*TAN(FNAC(D)))
180 DEFFND(R)=R*COS(P):DEFFNE(R)=R*SIN(P)
190 FORJ=0T0NP:CLS:SCREEN23,1:PRINTJ
200 X0=5.95:Y0=4.35:Z0=2.51
210 REM** Augenpunktbewegung **
220 X1=7*SIN(J*PI/18)+.1+X0
221 Y1=7*COS(J*PI/18)+.1+Y0
230 Z1=5+10*SIN(J*PI/18)+.1
240 :X1=X1-X0:Y1=Y1-Y0:Z1=Z1-Z0
250 M1=Y1/X1:N1=Z1/X1
260 RESTORE570:FORI=1T0K:READX2,Y2,Z2
270 X2=X2-X0:Y2=Y2-Y0:Z2=Z2-Z0
280 A1=Y2*Z1-Y1*Z2:B1=X1*Z2-X2*Z1
290 C1=X2*Y1-Y1*Y2:D=FNA(B1)
300 IFABS(D)>1THEND=-.9999999*SGN(D):GOTO320
310 IFD=0THEND=.00001
320 P=FNAS(D):IFC1=0THENP=PI-P
330 M2=(Y2-Y1)/(X2-X1):N2=(Z2-Z1)/(X2-X1)
340 D=FNB(M1):R=FNC(D)
350 X4(I)=FND(R):Y4(I)=FNE(R):NEXT
360 X1=X1E38:Y1=X1:X2=X1:Y2=X2:FORI=1T0K
370 IFX4(I){X1THENX1=X4(I)
380 IFX4(I){X2THENX2=X4(I)
390 IFY4(I){Y1THENY1=Y4(I)
400 IFY4(I){Y2THENY2=Y4(I):NEXT
410 FORI=7T010:GOSUB480:NEXT
420 FORI=11T012:GOSUB480:NEXT
430 FORI=14T018STEP2:GOSUB480:NEXT
440 FORI=20T025:GOSUB480:NEXT
450 Y4(O)=Y4(21):Y4(1)=Y4(24)
460 Y4(O)=Y4(21):Y4(1)=Y4(24)
470 D0KE3631,P(J):U=USR(O):NEXTJ:GOTO650
480 :X8=FNX(X4(I-1)):X9=FNX(X4(I))
490 Y8=FNX(Y4(I-1)):Y9=FNX(Y4(I))
500 IFX9=X8THENKS=1E5-SF=1:GOTO520
510 KS=(Y9-Y8)/(X9-X8):SF=SGN(X9-X8)
520 :FORXS=X8T0X9STEP.99*SF/SQR(1+KS*KS)
530 XD=FNI(XS):YD=FNI(Y8+(XS-X8)*KS)
540 IFXD{0ORXD}{940RYD}{00RYD}44THEN560
550 SET(XD,YD)
560 :NEXT:RETURN
570 DATA0,0,0,12.8,0,0,12.8,19.5,0,0,19.5,0
580 DATA0,0,0,2.3,0,0,9.4,2.3,0,9.4,6.4,0
590 DATA2.5,6.4,0,2.5,2.3,0,2.5,2.3,0

```

```

600 DATA 2.5,2.3,3.7,9.4,2.3,0.9,4.2,3,3.7
610 DATA 9.4,6.4,0.9,4.6,4.3,7.2,5.6,4.0
620 DATA 2.5,6.4,3.7,2.6,9.3,5.9,6.9,3.5
630 DATA 9.4,3.5,5.9,1.8,3.5,2.1,8,3.5
640 DATA 2.4,3.5,5.2,6.9,3.5
650 *FORJEOTOMP:DOKE3631,P(J)
660 U=USR(O):FORI=I030:NEXTI,J
670 :FORJ=2TOMP:DOKE3631,P(J)
680 U=USR(O):FORI=I030:NEXTI,J:GOTO670
690 :RESTORE720:FORI=3584T03668STEP2:READX
700 DOKEI,X:NEXT:DOKE4100,3548:U=USR(O)
710 DOKE4100,3614:RETURN
720 DATA4385,8718,3195,20769,8718,3189,18125
730 DATA-14066,14848,3667,8359,-8441,-12163
740 DATA16638,-16329,10538
750 DATA32268,24574,9512.1,-8956.33
760 DATA-760.33,-8800,86,24317,-8960,115,29437
770 DATA2816.9181,9213,-20104,-5344,12975,3667
780 DATA16073,12801,3667,30409,0,8319

```

0k

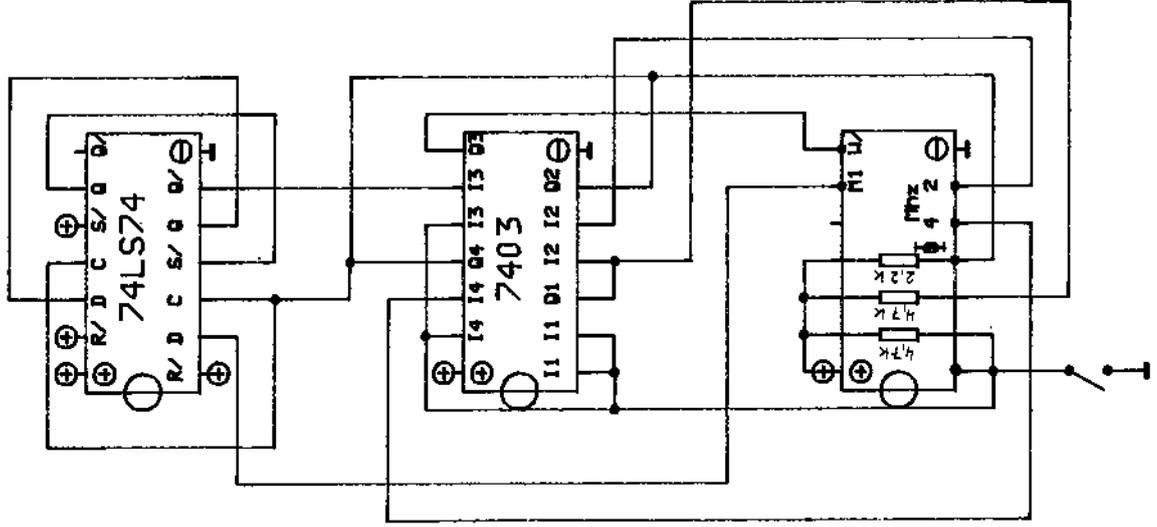
# VARPROTECT

```

0F00 21 12 0F 22 7B 0C 21 A9 C4
0F08 0F 22 75 0C AF 32 85 0F 6E
0F10 DF 5A DF 7D D0 F5 3A B5 68
0F18 0F A7 C4 7B 0F F1 FE 0D 27
0F20 37 C0 3E 17 DF 65 2A 29 12
0F28 0C 2B 23 7E FE 20 28 FA 4F
0F30 FE 3A 30 05 FE 30 D4 3D EB
0F38 0F 3E 0D 37 C9 2A D8 10 B3
0F40 ED 4B D6 10 ED 42 22 AD 6B
0F48 0F 2A DA 10 23 ED 42 22 EE
0F50 AB 0F ED 73 B3 0F 2A B3 18
0F58 0F 11 EC FF 19 22 AF 0F 6B
0F60 23 ED 4B AB 0F 08 ED 42 BE
0F68 03 22 B1 0F 2A D6 10 ED 59
0F70 5B B1 0F DF 49 3E 01 32 33
0F78 B5 0F C9 2A AF 0F 22 C3 E1
0F80 10 2A D6 10 ED 4B AD 0F A3
0F88 ED 4A 22 D8 10 2A D6 10 E8
0F90 ED 4B AB 0F 08 ED 4A 22 F5
0F98 DA 10 2A B1 0F ED 5B D6 99
0FA0 10 03 DF 49 AF 32 85 0F 8F
0FA8 C9 76 00 0D 00 0C 00 9B AA
0FBO AF 8F AF AF AF 00 00 0A

```

## 2/4Mhz + M1-Wait-State





# Stringsave

von G.T. Klement

The mad scientist has done it again!!

Das Maschinenprogramm liegt von CCO aufwärts (um nicht mit TOOLKIT zu kollidieren) und ist eine wesentlich verbesserte Version von STRINGSAVE, das ich den Freunden bei INMC vor längerer Zeit geschickt habe. Es können String Arrays von BASIC gespeichert und gelesen werden, wobei ich mich an die simple Checksum Prüfung von Microsoft gehalten habe. Der Aufruf zum Speichern erfolgt mit U=USR(A\$)S. A\$ oder jedes beliebige andere String Array wird auf Band gespeichert und kann mit U=USR(A\$)L wieder geladen werden. Es wurden die Error Meldungen von BASIC verwendet.

# VARPROTECT

von G.T. Klement

Wir kennen alle die Situation: 1h30min am Morgen. Das Programm läuft zum ersten Male fehlerfrei. 25 Minuten Rechenzeit. Im Mittelwellenempfänger, mit dem wir den Lauf des Programmes kontrollieren, kündigt sich das Ende der Rechnung an. DIE ERSTE ZAHL ..... und Error in 2040 ! Ein einfacher Syntaxfehler. Wieviele NASCOMS sind in dieser Situation zertrümmert worden? VARPROTECT sitzt in der Input Routine und speichert bei Eingabe einer Zahl die Variablen. Nachdem das Programm geändert wurde, werden die Variablen wieder herangerückt, und mit GOTO kann das Programm fortgesetzt werden. (Programmstart von VARPROTECT mit E F00)

Verkaufe NASCOM-1  
im Gehäuse komplett mit Netzteil  
Hardware: Europakarten-Bus (3 Steckplätze),  
I/O-Karte, 4K RAM, Kansas-City-Interface.  
Software: NASSYS-i, T2, NASPEN, BASIC  
plus umfangreicher Dokumentation  
Boye Hamkens  
Tel.: [REDACTED]

Mini-Puffer-Karte  
wie in NJ 10/82 beschrieben  
fertig aufgebaut und getestet: DM 105,-  
Platine: DM 65,-  
Bei größere Auflage entsprechend billiger  
U. Forke  
[REDACTED]

# Swinghouse

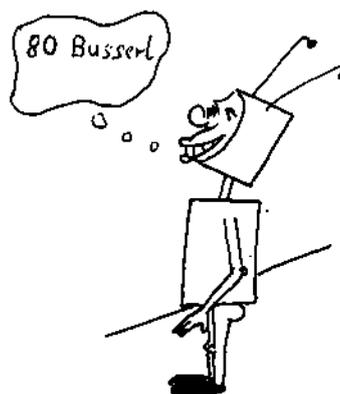
von G.T. Klement

Etwas für Tüftler! Voraussetzung RAM bis 8000. Es geht auch mit weniger, aber ist ohne Änderung nicht so hübsch. Memsize mit 8193 begrenzen. Die Vorgeschichte: Beim Editieren von Programmen sollte mit der Taste a) der Bildschirminhalt mit einem Teil des RAMS gewapt werden, so daß verschiedene Teile des Programmes ohne das ärgerliche LIST verglichen werden können. Während der Arbeit kam mir die Idee, möglichst viele Screens zu swapen und so eine Animation von Graphic zu erzeugen. Auf Deutsch : das Maschinenprogramm wurde etwas vergewaltigt und sollte von einem NASCOM Freak leicht optimiert werden können. Wenn Sie das Programm starten, haben Sie noch Zeit, die Katze aus der Wohnung zu werfen und Ihrer Frau den Gute-Nacht Kuss zu geben. Wenn alle 36 Bilder gezeichnet sind, erleben sie einen unvergleichlichen Flug um mein Sommerhäuschen in perspektivischer Darstellung, der Sie sogar unter das Erdreich führt. Beim nächsten Wien-Besuch haben Sie den Stadtrundflug gespart!

SUCHE  
ZEAP 2.0 Assembler, NAS-DIS, DEBUG Programmierhilfe für NAS-SYS 1 in Eproms 2708, komplett oder einzeln  
Ewald K. Rieger  
[REDACTED]

SUCHE Schachprogramm  
Version "G" (mit Graphik).  
BIETE ZEAP 2.0, NAS-DIS, DEBUGGER und BASIC  
PROGRAMMIERE EPROMS  
2708, 2716, 2532, 2732, 2764  
H. G. Schäfer  
[REDACTED]

Tel.: [REDACTED] dienstl.: [REDACTED]



# Mini Buffer Board

von H.G. Ingelaat und U. Forke

Wie schon im letzten Journal angekündigt, hier ein Vorschlag zur besseren Pufferung des NI ohne aufwendige Umbauten. NI-Benutzer, die schon eine Original-Pufferkarte haben, müssen zum Einbau der Mini-Karte den Grundzustand des Systems wiederherstellen. Verfahren Sie dabei wie folgt:

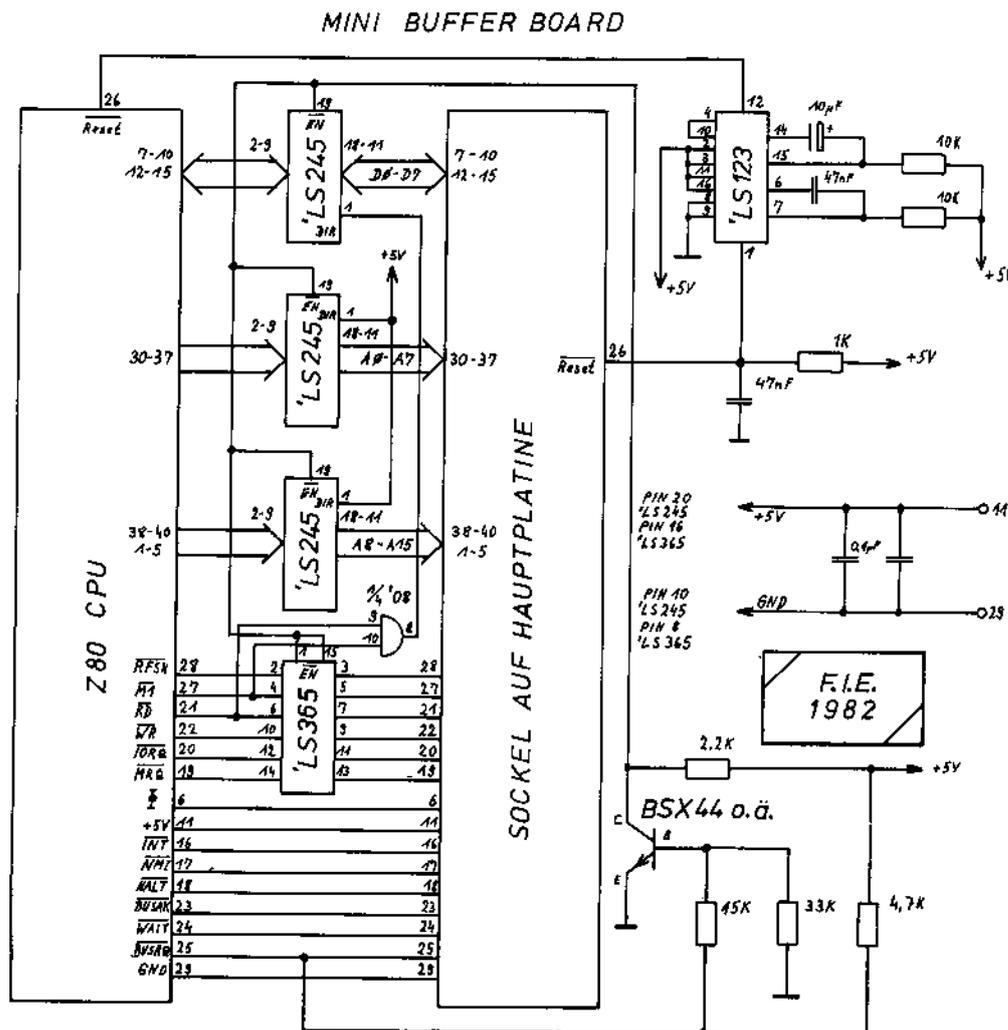
- Entfernen Sie die Verbindung von IC 31, Pin 13 nach 0V. (Diese mußte auf Originalkarte hergestellt werden.)
- Legen Sie die Brücke LK6 auf die gewünschte Systemfrequenz.
- Legen Sie die Resetleitung, die von der Tastatur kommt, wieder auf den entsprechenden Anschluß des Sockels "KBD".

Damit ist der benötigte Grundzustand hergestellt. Nehmen Sie jetzt die CPU aus dem Sockel der Grundplatte, und stecken Sie diese in den vorgesehenen Sockel auf der Mini-Puffer-Karte. Setzen Sie die Mini-Karte nun anstelle der CPU

in den Sockel auf der Grundplatte. Verdrahten Sie die Anschlüsse der Steckerleiste PL1 auf der Grundplatte zu den entsprechenden Anschlüssen des NASCOM-Motherboards. Die Anschlußbelegungen dazu entnehmen Sie den Heften "Nasbus Funktionsbeschreibung" und "Buffer-board Functional Specification".

Falls Sie Zusatzkarten anderer Z80-Systeme benutzen (z.B. den ECB-Bus), muß die Verdrahtung natürlich entsprechend dem System erfolgen. Die Mini-Karte kann aber auch in andere Z80-Systeme eingebaut werden, wenn eine Pufferung nachträglich erforderlich wird.

Bei der Inbetriebnahme des NI werden Sie jetzt feststellen, daß dieser völlig unanfällig gegen Störungen ist, die z.B. beim Einschalten elektrischer Verbraucher entstehen. Bei meinem System ergab sich zudem der Vorteil, daß die 32K-RAM-Karte mit 4 MHz Systemfrequenz und ohne Waitzyklen problemlos arbeitet.





# Drehzahl-Messung für Cassettenrecorder von H. Gnirck

(Herr Gnirck hat eine interessante Möglichkeit gefunden, die ganze Bandlänge einer Compact-Cassette für Daten und Programme auszunutzen, ohne stundenlang die richtige Bandstelle suchen zu müssen. Die Beschreibung ist vielleicht etwas knapp geraten - Interessenten wenden sich bitte bei Rückfragen direkt an den Verfasser. RED )

## Betrieb

Der "Dorn" wird auf die Cassetten-Recorder-Spindel gesteckt (Cassette ist eingelegt). Danach Deckel mit eingebauter Lichtschranke schließen. Jetzt Zähler "RESET". Nach Aufschreiben der File-Länge beim Auflesen kann mit schnellem Vorlauf das gesuchte File angefahren werden.

## Einbau der Lichtschranke

Werkstoff : Kunststoff (schwarz) bzw. Holz (Innenflächen schwarz gestrichen)

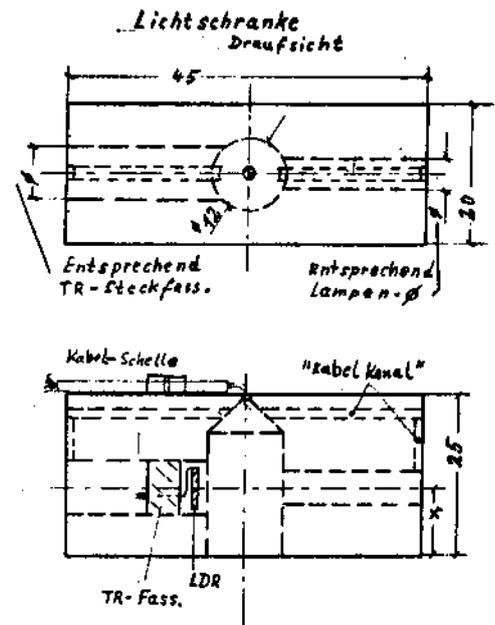
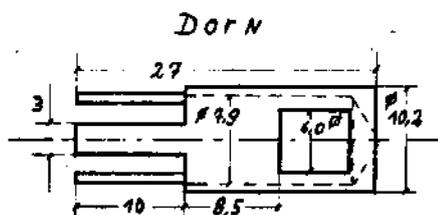
Montage: Maß x ermitteln (Höhe = Deckel Cassettenfach bis Mitte Dornfenster). Mitte Durchmesser 12 muß Mitte Cassettenspindel sein. Danach Ausschnitt im Cassettendeckel machen (ich habe zum Ausschneiden eine Schablone der Lichtschranke auf den Deckel geklebt). Lichtschranke mit UHU-Plast einkleben.

Zähler: Nach CHIP 12/81. Taste Up/Down als Mikroschalter bei Cassetten-Recorder-Taste "Schneller Rücklauf" eingebaut. Achtung: muß schalten, bevor Motor die Spindel bewegt! Leitung IC VII Pin 7 zu IC I, III, V Pin 15 auftrennen. Mit Teiler /2 Takt halbieren.

## Herstellung des "Dorns"

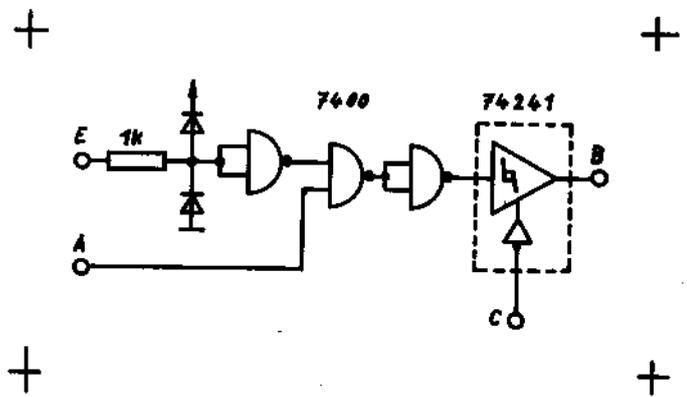
Material: Hirschmann-Buchse für Büchelstecker (Kabelanschluß), Farbe schwarz.

Arbeiten: Einsatz entfernen, Schraube eindrehen. Auf der Handbohrmaschine mit Schmirgelpapier auf Durchmesser 10,2 schleifen. Innendurchmesser aufbohren auf 7,9 (Zapfenmaß der Treibspindel des Cass.-Rec.). 3 Stück Stege mit 3mm Breite (Maß zwischen 6-fach-Stern der Cassette und 3-fach-Stern der Spindel) sägen bzw. feilen. Fenster für LDR aussägen. Feinarbeiten bis Einsatz "leichtgängig" auf Spindel geht.



# Handshake für UART von Peter Urban

Nach der Anschaffung eines Druckers stellte sich heraus, daß die UART im NASCOM2 keine Busy-Leitung hat. Abhilfe schafft die untenstehende kleine Schaltung. Man trennt dazu Pin 22 der UART (Transmitter Buffer Register Empty) vom Datenbus und verbindet ihn mit Punkt A. Punkt B wird mit dem Datenbus D6 verbunden. Punkt C wird mit Pin 16 der UART verbunden. Liegt nun an E ein High an wird dem Monitor ein volles Buffer Register vorgegaukelt und die Ausgabe unterbrochen.



# NASCOM- Sonderangebote

<b>NASCOM II</b> mit 8k BASIC und NAS-SYS Monitor, mit 1k RAM-Speicher, erweiterbar auf der Platine bis auf 8k RAM mit MK 4118	<b>1395,-</b>
<b>Netzteil</b> zum NASCOM-Computer 3A bei 5V, 12V 1A, -12V 1A, -5V 1 A	<b>115,-</b>
<b>PIO</b>	<b>18,-</b>
<b>Tastaturerweiterung</b> für NASCOM 1 auf NASCOM 2	<b>69,-</b>
<b>Minigrafik</b> ohne Grafikrom, läuft jedoch mit NASCOM 2 Grafikrom	<b>64,-</b>
<b>NASCOM III</b> mit 48k Speicher, Gehäuse, Lüfter	<b>2295,-</b>
<b>VEROFRAME</b> für NASCOM 1 oder 2	<b>85,-</b>
<b>Steckleiste</b> für NASBUS	<b>24,-</b>
<b>Farbbänder</b> für IMP-Drucker	Stück <b>19,-</b>
<b>Motherboard</b> für NASCOM, 14 Steckplätze	<b>35,-</b>
<b>Disassembler</b> für NAS-SYS 1	<b>75,-</b>
<b>Disassembler</b> für NASBUG T 4	<b>49,-</b>
<b>NASPEN</b> wahlweise für NAS-SYS oder NASBUG T 4 / T 2	<b>49,-</b>
<b>Schach</b> wahlweise für NAS-SYS 1 oder NASBUG T 4 — bitte angeben!	<b>25,-</b>
<b>Umschaltkarte</b> zum Umschalten zwischen 2 Betriebssystemen	<b>49,-</b>
<b>NASCOM 1</b> mit NAS-SYS	<b>495,-</b>
<b>80-Zeichen-Karte</b> für NASCOM II	<b>625,-</b>
<b>43-polige Steckleiste</b> für NASCOM I	<b>19,-</b>