

ELCOMP[®]

FACHZEITSCHRIFT FÜR MICROCOMPUTER

2. Jahrgang Volume II

April 1979

DM 4,50

sFr 4,50

F 4,50

ÖS 35,-

Nr. 4

AUS DEM INHALT:

- 01 KIM-Software – Es lebe der KIM-1
- 13 Disassembler und Editor für KIM-1
- 16 CAI Computer Assisted Instruction
- 19 Optimieren Sie Ihre Programmier-technik
- 23 Jetzt kommen die Roboter
- 26 Zeilenveränderungsprogramm f. PET
- 34 Was ist eine Computersprache
- 36 Kurz notiert
- 37 Messebericht Consumer Show in Las Vegas
- 39 Dreidimensionale Graphik mit dem PET
- 41 Wie lernt man BASIC, Teil VI

- 45 Hardwareneuheiten
- 48 Softwareneuheiten
- 49 Microcomputer Book Store
- 55 Software Shop
- 58 Buchbesprechungen

SN-Nr. 0171-0958

KIM-Software – Es lebe der KIM-I

KIM-1 für Anfänger

Dieser Artikel soll mit einer kurzgefaßten Einführung eine ständige Arbeitsunterlage für alle KIM-1 Besitzer sein. Falls Sie zu den Leuten gehören, die sich einen KIM-1 gekauft haben, aber an der großen Problematik fast verzweifeln, daß Sie sich Ihre Informationen aus vier verschiedenen Büchern holen müssen und am Schluß doch nicht so richtig schlau daraus werden, dann sollten Sie hier unbedingt weiterlesen.

Dieser Artikel soll Sie wieder auf den richtigen Weg führen.

Alles was Sie dazu benötigen ist:

1. KIM-1
2. Cassettenrecorder und dadurch bedingte 12V, 0,1A Spannungsquelle
3. 8 x LED's
4. 8 x Widerstände 180Ω , 0,25 W
5. 1 x Widerstand $3,3K\Omega$, 0,25 W
6. 2 x IC 7404
7. 6 x Einpolige Umschalter
1 x Spannungsquelle 5V, 1,2A $\pm 5\%$ für den Betrieb des KIM-1
8. 1 x Lautsprecher 8Ω (Lochraster)
9. Powerace Experimentiergerät von AP-Products
10. Anschlußlitze
11. DIN-Stecker für die Verbindung zwischen Cassettenrecorder und KIM-1 (Lautsprecherstecker, Diodenstecker 5polig)
12. Lötkolben, Lötzinn
13. Softwarecassette
14. 1 x Steckerleiste für den Anschluß an das KIM-System

Wenn Sie keinen Cassettenrecorder in Betrieb nehmen wollen, so spielt es keine Rolle; Sie können trotzdem die Programmbeispiele eingeben.

A) Anschlüsse des Systems

Für unsere Versuche ist nur der Applikationsstecker des KIM-1 von Bedeutung, der bei der Draufsicht unten sichtbar ist. Gekennzeichnet ist er mit A1 am oberen Ende und A22 am unteren Ende.

Oberhalb wird durchnummeriert von A1 bis A22
Unterhalb wird durchnummeriert von AA bis AZ (AG, AI, AO, AQ ausgeschlossen)

Mit der Netzgerätemasse wird der PIN A1 und AK verbunden.

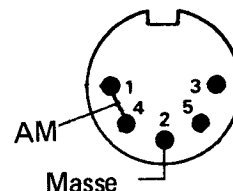
Mit + 5V, 1,2A $\pm 5\%$ wird PIN AA verbunden.
Mit + 12V, 0,1 A $\pm 5\%$ wird PIN AN verbunden.

Der 12V Anschluß ist nur für die Inbetriebnahme eines Cassettenrecorders nötig.

B) Zum Cassettenrecorder noch die Anschlüsse:

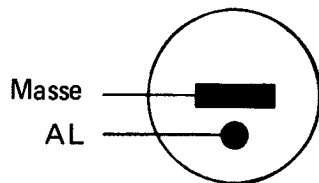
Ich gehe davon aus, daß Ihr Cassettengerät DIN-Anschlußbuchsen hat.

Wenn Sie den 5poligen Diodenstecker aufmachen, haben Sie das Kernstück mit den 5 Anschlüssen in der Hand. Sie verbinden dabei die Anschlüsse 1 und 4 und führen diesen gemeinsamen Anschluß an das PIN AM. Den Steckeranschluß 2 legen Sie an die Netzgerätemasse.



Zu der Dateneingabe von Cassette müssen wir den Lautsprecher- oder Kopfhöreranschluß des Cassettengerätes heranziehen.

Nur dieser liefert einen genügend großen Pegel für das Übertragungssignal. Den Lautsprecherstecker müssen Sie folgendermaßen anschließen.



Die „Zunge“ schließen Sie an Netzteilmasse an. Die Stiftverbindung verbinden Sie mit dem PIN AL.

Somit wären Sie für Programmabspeicherung und Programmwiedergabe gerüstet.

C) Zur Programmabspeicherung auf Cassette

Sie müssen nachstehende Adressen mit folgenden Inhalten belegen:

Adresse	Inhalt
00F1	00
17F5	LB – Startadresse
17F6	HB – Startadresse
17F7	LB – Endadresse + 1
17F8	HB – Endadresse
1FF9	Kennziffer (zw. 01 und FE)

Erklärung:

LB → Lower Byte

HB → Higher Byte

Zu der Startadresse

Man beginnt meist mit der Adresse 0200. Dabei sind die letzten beiden Adressenstellen das LB und die zwei ersten Adressenstellen das HB. In unserem Falle

$$00 = LB ; 02 = HB$$

Zu der Endadresse:

Es ist immer die Endadresse + 1 notwendig. Angenommen, das Programm endet bei Adresse 03FF, 1 dazu addiert ergibt 0400. Das LB=00 das HB = 04

Für alle anderen Adressenkonfigurationen gilt das Entsprechende.

Nachdem Sie die Bedingungen eingegeben haben, geben Sie die Adresse 1800 XX ein. (XX = beliebiger Inhalt).

Lassen Sie nun Ihren Cassettenrecorder aufnahmebereit abspielen. Nach ein paar Sekunden drücken Sie die GO-Taste Ihres KIM-1 und somit ist die Datenübertragung auf Cassette im Gange. Das Display erlischt. Die Wartezeit ist deswegen notwendig, um einen exakten Gleichlauf der Bandgeschwindigkeit zu erreichen.

Nachdem das Programm bis zu der Endadresse abgespeichert ist, meldet sich Ihr KIM-1 mit 0000 XX oder FFFF XX. Ersteres bedeutet eine fehlerfreie Übertragung; letzteres eine fehlerhafte Übertragung.

Mögliche Fehlerquellen sind:

1. Sie haben entweder eine zu hohe oder zu niedere Lautstärke am Cassettenrecorder eingestellt. (Beste Einstellung: Mittlere Tonstärke, Tonblende ganz hoch einstellen)
2. Sie haben die + 12V Versorgungsspannung nicht angeschlossen

Möglichkeit des Vergleichens nach Programmabspeicherung von Kassettendaten mit Speicherinhalt von KIM-1

Folgendes muß getan werden:

Adressen 17E7 und 17E8 mit Dateninhalt 00 versehen.

Eingabe der Routine:

17EC	CD	00	00
17EF	DO	03	
17F1	4C	0F	19
17F4	4C	29	19

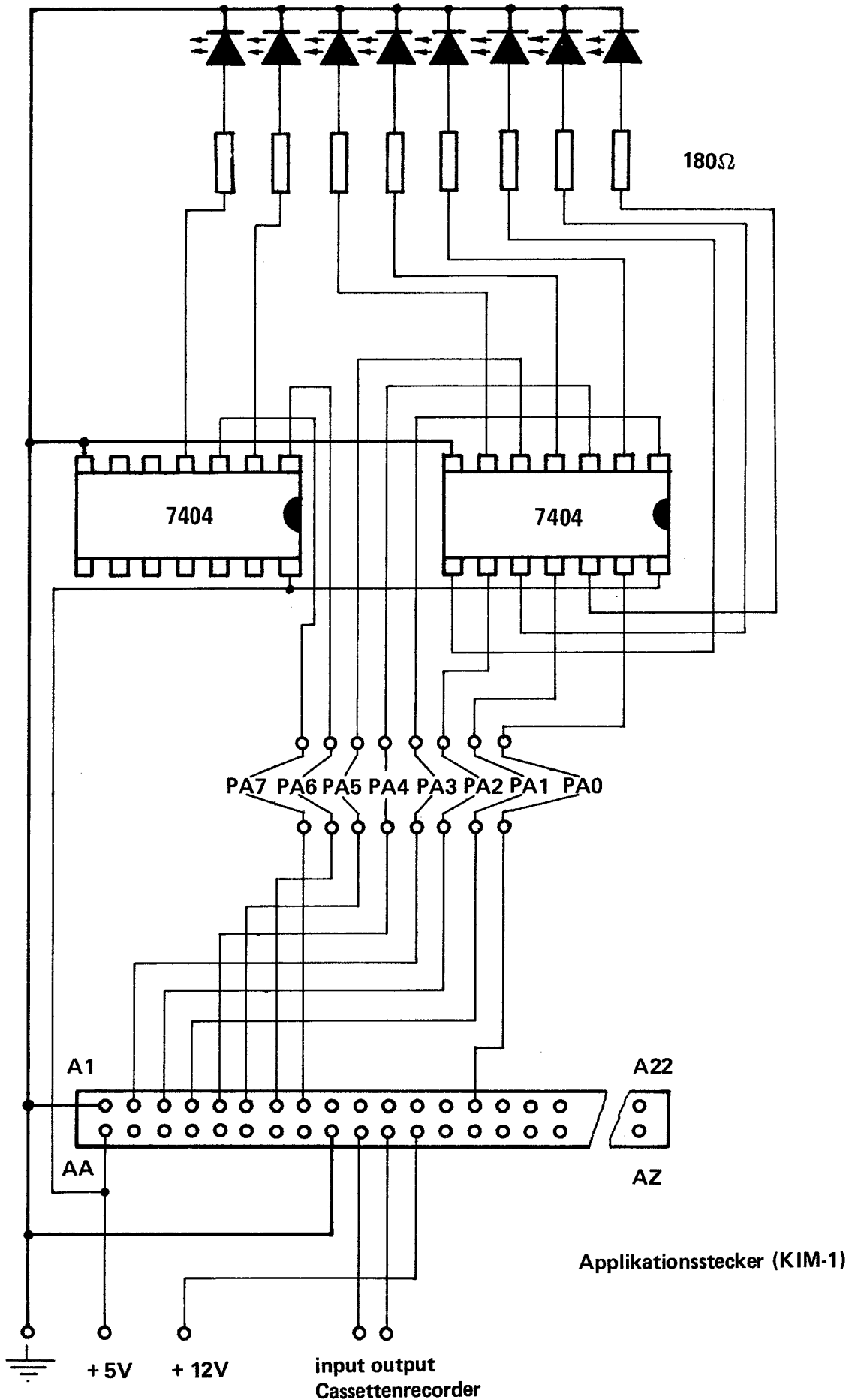
Cassette zurückspulen

Eingabe der Adresse 188C XX. GO-Taste drücken. Display erlischt. Cassettenrecorder abspielen.

Nach dem Vergleich meldet sich KIM-1 mit 0000 XX oder FFFF XX.

Zur Programmwiedergabe von Cassette
 Hierbei ist folgendes zu beachten:
 Adresse 00F1 ist mit Dateninhalt 00 zu ver-
Praktischer Aufbau (Hardware)

sehen. Adresse 17F9 muß dieselbe Kennziffer
 als Inhalt haben, wie das abgespielte Programm.



Nach diesen Eingaben geben Sie die Adresse 1873 XX ein und drücken die GO-Taste. Das Display erlischt. Somit erwartet Ihr KIM Daten von Cassette. Spielen Sie nun Ihren zurückgespulten Cassettenrecorder ab.

Nachdem das Programm mit der entsprechenden Kennziffer gefunden wurde, wird es im Arbeitsspeicher abgespeichert. Bei Programmende meldet sich Ihr KIM-1 wieder mit 0000 XX oder FFFF XX. Bedeutung wie bei Programmspeicherung.

Mögliche Fehlerquellen bei FFFF XX:

1. Falsche Kennziffer wurde eingegeben. Somit wird Ihr Programm nicht identifiziert.
2. Lautstärkeregelung und Tonblendeinstellung am Cassettengerät nicht optimal
3. Nicht Lautsprecher Ausgang des Cassettengerätes sondern Buchse zum überspielen auf andere Tonaufzeichnungsgeräte wurde verwendet. Hier ist das Signal zu schwach. Es müßte verstärkt werden.

Der KIM-1 besitzt insgesamt 15 Anschlüsse für Dateneingabe oder Datenausgabe. Durch Programm können diese Anschlüsse als Ein- oder Ausgänge deklariert werden.

Die 15 Anschlüsse teilen sich auf in 8 I/O-Ports PA und 7 I/O-Ports PB.

Nach dem Einschalten des KIM-1 sind alle 15 Ports auf Dateneingang geschaltet. Dies ist immer der Fall.

Das Register, welches die Ein- oder Ausgänge bestimmt, ist das Datenrichtungsregister. PADD für PA und PBDD für PB.

Ihre Adressen sind 1701 und 1703. Der Inhalt der Adressen ist zu Beginn 00. Das dazugehörige Datenregister PAD für PA und (PBD) für PB hat zu Anfang FF. Ihre Adressen sind 1700 und 1702.

In unseren folgenden Beispielen wollen wir nur das A-Register mit seinen 8 I/O-Ports verwenden.

Mathematische Behandlung der Umrechnung zwischen den Codes

Von Binär in Dezimal nach dem Horner Schema:

z.B. 00 10 00 01

$$\begin{array}{r} 00100001 \\ \times 2 \\ \hline 00100001 \\ 00200002 \\ 00400004 \\ 00800008 \\ 01600016 \\ \hline 00100012 \\ 00200024 \\ 00400048 \\ 00800096 \\ \hline 00100123 \rightarrow \text{dez.} \end{array}$$

Von Dezimal in Hexadezimal
33 : 16 = Rest 1 → 21

Umgekehrt, von Hexadezimal in Dezimal

$$\begin{array}{r} 21 \\ \times 16 \\ \hline 32 \\ 200 \\ \hline 232 \end{array}$$

Von Dezimal in Binär

$$\begin{array}{l} 33 : 2 = 16 \text{ (Rest 1)} \\ 16 : 2 = 8 \text{ (Rest 0)} \\ 8 : 2 = 4 \text{ (Rest 0)} \\ 4 : 2 = 2 \text{ (Rest 0)} \\ 2 : 2 = 1 \text{ (Rest 0)} \\ 1 : 2 = 0 \text{ (Rest 1)} \end{array} \left. \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{array} \right\} 0010\ 0001$$

Postulat: Wenn an einem Port log. „0“ ist, so entspricht dies einem Spannungspegel von 0V. Ist ein Port log. „1“, so entspricht dies einem Spannungspegel von +5V.

Wir müssen diese Bedingungen bei dem Gebrauch der verwendeten ICs als Inverter berücksichtigen. Die Inverter kehren bekanntlich jedes Signal um. So werden 0V in +5V und umgekehrt.

Wenn ein invertierendes Signal vorliegt, so wird es graphisch durch einen Schrägstrich über der Bezeichnung sichtbar gemacht.

Ein praktisches Beispiel:

Wir wollen PA0 und PA5 als Ausgänge und PA1, PA2, PA3, PA4, PA6 und PA7 als Eingänge schalten.

Wir müssen also erst das A-Datenrichtungsregister richtig setzen.

Binärcode: $0010\ 0001 \hat{=} (33)_{10} \hat{=} (21)_{Hex}$.

Wenn Sie in das A-Datenrichtungsregister, Adresse 1701 die 21 eingeben, so sind die Ein- und Ausgänge bestimmt.

Es leuchten zunächst die LEDs 1 und 5. Das hat aber noch nichts zu bedeuten, da das Datenregister hier zur Anzeige kommt. Schauen wir in das Datenregister Adresse 1700, so steht dort $(DE)_{Hex} \hat{=} 222 \hat{=} 1101\ 1110$

Man kann also sagen, daß im Datenregister das Äquivalent von Datenrichtungsregister zu $(255)_{10} \hat{=} 1111\ 1111 \hat{=} (FF)_{hex}$ steht.

Dies gilt jedoch nur zu Anfang.

Wollen wir Daten nach außen geben und somit das Datenregister verändern, so werden Sie feststellen, daß sich der Inhalt des DR plötzlich auf FF und FE verändert, obwohl im DRR die 21 steht und somit die Ausgänge festlegt. Es leuchtet auch nur noch das PIN PA0 auf, bei R-Inhalt $(FE)_{hex} \hat{=} (1111\ 1110) \hat{=} (254)_{10}$.

Allgemein kann man daher sagen:

Durch deklarieren der Ein- und Ausgänge kann man keine Daten über Tastatur ausgeben. Dies ist nur über Programm möglich.

Mit nachstehendem Programm wollen wir es verdeutlichen.

0300	A9	FF		Lade Accu mit FF
0302	8D	01	17	Speichere < Accu > in Adresse 1901
0305	EE	00	17	inkrementiere < Adr. 1701 > um 1
0308	4C	05	03	Absoluter Sprung zu Adresse 305

Durch dieses kleine Programm werden alle Ports als Ausgänge geschaltet (erste beiden Befehle). Nun steht im PAD Adr. 1700 00; diesem Wert 00 wird eine 1 dazuaddiert. Dann steht im PAD 01. Nun wird dem Wert 01 eine 1 hinzuaddiert u. s. w.

Durch Single Step Befehle können wir nun sehr schön sehen, wie das Programm abläuft. Dazu stellen Sie den kleinen Schalter in der oberen linken Ecke des Tastenfeldes auf ON. Geben Sie die Adresse 0300 ein und drücken Sie die GO-Taste.

Nun wird der erste Befehl abgearbeitet. Drücken Sie nochmals die GO-Taste, so wird der 2. Befehl in der Reihenfolge des Programmes abgearbeitet. Alle LEDs leuchten auf; wegen der Inverter, d. h. im PAD steht 00.

Sie können nach jedem Schritt in die Adresse 1700 einsehen. Danach gehen Sie wieder an die Adresse, an der Sie unterbrochen haben und drücken wieder die GO-Taste.

Sie werden feststellen, daß sich nun 1. die LED-Anzeige und 2. der Inhalt von Adr. 1700 (PAD) entsprechend verändert.

Somit haben wir es erreicht, verschiedene Daten auszugeben. Der LED-Anzeige entspricht der Wert in der Adresse 1700; die Inverter berücksichtigt.

Wenn Sie keine Single-Step-Befehle eingeben, sondern das Programm im Normalbetrieb arbeiten lassen, so sehen Sie wegen der kleinen Zykluszeit alle LEDs leuchten.

Es wäre also vorteilhaft, dem Programm eine Verzögerung beizufügen, um so den Betrieb einem Single-Step-Befehl anzunähern.

Dazu nun das Programm mit Zeitschleife

Zähler ohne Verzögerung

	LDA	IM	FF		200	A9	FF	
	STA	ABS	01	17	202	8D	01	17
E1	INC	ABS	00	17	205	EE	00	17
	JMP	ABS	E1		208	4C	05	02

Zähler mit Verzögerung

	LDA	IM	FF		200	A9	FF	
	STA	ABS	01	17	202	8D	01	17
E1	INC	ABS	00	17	205	EE	00	17
	LDA	ZP	t1		208	A5	00	
	STA	ZP	t		20A	85	01	
	JSR	ABS	VERZ		20C	20	20	02
	JMP	ABS	E1		20F	4C	05	02
VERZ	LDY	ZP	t		220	A4	01	
E2	LDX	ZP	Z		222	A6	02	
E3	DEX	IMP			224	CA		
	BNE	R	M3		225	D0	FD	
	DEY	IMP			227	88		
	BNE	R	M2		228	D0	F8	
	RTS	IMP			22A	60		

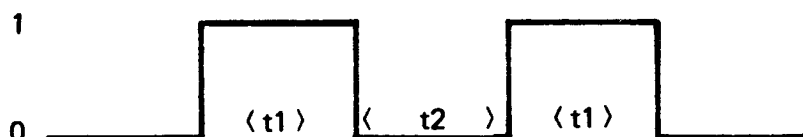
Zeitliche Festlegung der Verzögerung	00	t1		
Speicheradresse	01	XX	≅	t
Zeitfaktor	02	Z		

gewählt: t1 = FF
Z = FF

Wenn Sie nun die Adresse 0200 eingeben und RUN drücken, so können Sie die Verschiebungen im Datenregister direkt beobachten.

quenz beliebig einstellbar ist zwischen 2 Grenzen (der normalen Zykluszeit und der maximalen Verzögerung) durch das Programm:

In diesem Zusammenhang noch ein programmierbarer Rechteckgenerator, der in seiner Fre-



Die CPU benötigt zur Ausführung eines Befehls eine bestimmte Zeit, die allerdings sehr kurz ist. Wir entwerfen ein Programm, das die CPU eine genau festgelegte Anzahl von Befehlen ausführen läßt. Dieses Programm wirkt sich wie eine Verzögerung mit einer bestimmten Zeitkonstanten aus.

Es wird eine bestimmte Zahl t (Zeitkonstante) in das X-Register geladen. Die CPU durchläuft t mal die Schleife, wenn X = 0 ist, verläßt sie die Programmschleife.

Ein einfaches Programm, das eine kurze Zeitverzögerung bewirkt ist folgende Programmschleife:

	LDX	IM	Zeitkonstante
M	DEX	IMP	
	BNE	R	M

Will man eine Zeitverzögerung im Bereich von Zentelsekunden programmieren, muß man eine Doppelschleife anlegen. Da wir zweimal eine Verzögerung programmieren müßten, formulieren wir ein entsprechendes UP.

Zeitschleife

	LDA	IM	FF		200	A9	FF	
	STA	AB	01	17	202	8D	01	17
M1	LDA	IM	FF		205	A9	FF	
	STA	AB	00	17	207	8D	00	17
	LDA	Z8	t1		20A	A5	00	
	STA	ZP	t		20C	85	01	
	JSR	AB	VERZ		20E	20	20	02
	LDA	IM	00		211	A9	00	
	STA	AB	00	17	213	8D	00	17
	LDA	ZP	t2		216	A5	03	
	STA	ZP	t		218	85	01	
	JSR	AB	VERZ		21A	20	20	02
	JMP	AB	M1		21D	4C	05	02
	LDY	ZP	t		220	A4	01	
M2	LDX	ZP	Z		222	A6	02	
M3	DEX	IMP			224	CA		
	BNE	R	M3		225	D0	FD	
	DEY	IMP			227	88		
	BNE	R	M2		228	D0	F8	
	RTS	IMP			22A	60		

Zeitliche Festlegung d. log. „1“ Pegels	0000	t1	
Speicheradresse für t	0001	XX \triangleq t	
Zeitfaktor	0002	Z	
Zeitliche Festlegung d. log. „0“-Pegels	0003	t2	

gewählt für	t1	=	FF
	Z	=	FF
	t2	=	FF

in diesem Fall minimal mögliche Frequenz an den Ausgängen.

gewählt für	t1	=	1F
	Z	=	FF
	t2	=	FF

in diesem Fall ist der log. „1“-Pegel ziemlich kurz (sichtbar).
Der log. „0“-Pegel dementsprechend länger.

Achtung: Wenn hier von log. „0“ und log. „1“ die Rede ist, so ist der Pegel an den entsprechenden Outports gemeint – nicht an den LEDs.

Aufbauend auf diesen Rechteckgenerator können wir ein Lauflicht konzipieren. (Alles mit der gewohnten Hardware).

Hier wird der log. „1“ und log. „0“-Pegel hintereinander jeweils auf die verschiedenen Outports POA bis PA7 und bei PA0 wieder beginnend, gebracht.

Lauflicht

	LDA	IM	01		200	A9	01	
	STA	AB	01	17	202	8D	01	17
	STA	ZP	K		205	85	04	
M1	LDA	IM	FF		207	A9	FF	
	STA	AB	00	17	209	8D	00	17
	LDA	ZP	t1		20C	A5	00	
	STA	ZP	t		20E	85	01	
	JSR	AB	VERZ		210	20	00	03
	LDA	IM	00		213	A9	00	
	STA	AB	00	17	215	8D	00	17
	LDA	ZP	t2		218	A5	03	
	STA	ZP	t		21A	85	01	
	JSR	AB	VERZ		21C	20	00	03
	LDA	ZP	K		21F	A5	04	
	ADC	ZP	K		221	65	04	
	STA	ZP	K		223	85	04	
	STA	AB	01	17	225	8D	01	17
	JMP	AB	M1		228	4C	07	02

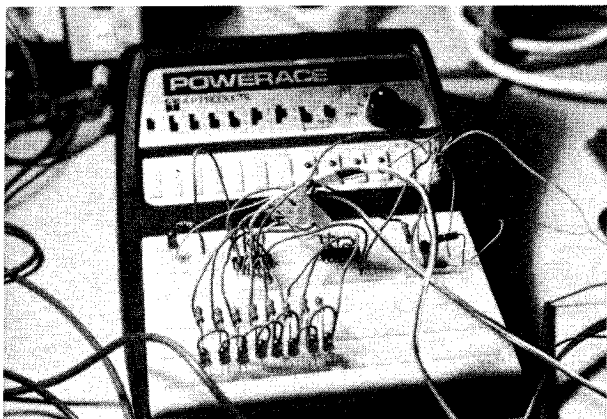
VERZ	LDY	ZP	t		300	A4	01
M2	LDX	ZP	Z		302	A6	02
M3	DEX	IMP			304	CA	
	BNE	R	M3		305	D0	FD
	DEY	IMP			307	88	
	BNE	R	M2		308	D0	F8
	RTS	IMP			30A	60	

Zeitliche Festlegung des log. „1“-Pegels	0000	t1
Speicheradresse für t	0001	XX $\hat{=}$ t
Zeitfaktor	0002	Z
Zeitliche Festlegung des log. „0“-Pegels	0003	t2
Speicheradresse für k	0004	XX $\hat{=}$ K

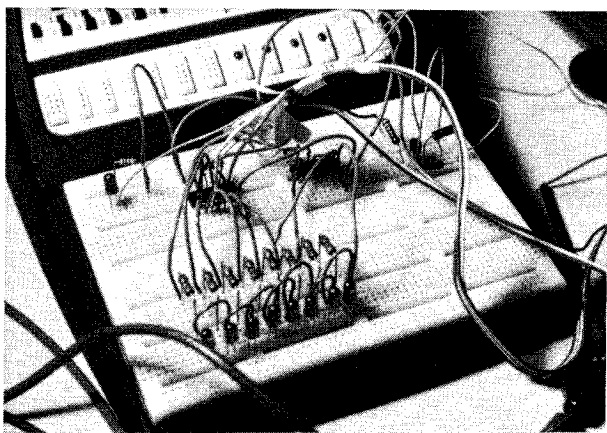
gewählt wurde für

t	=	01
Z	=	19
t2	=	FF

Mit demselben Schaltungsaufbau können wir ein einfaches Spiel betreiben.



Powerace als sehr praktisches Experimentierhilfsmittel



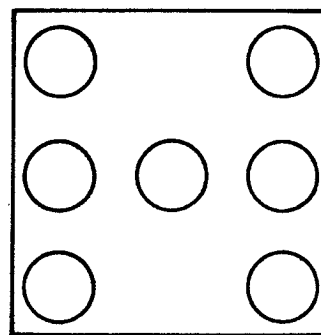
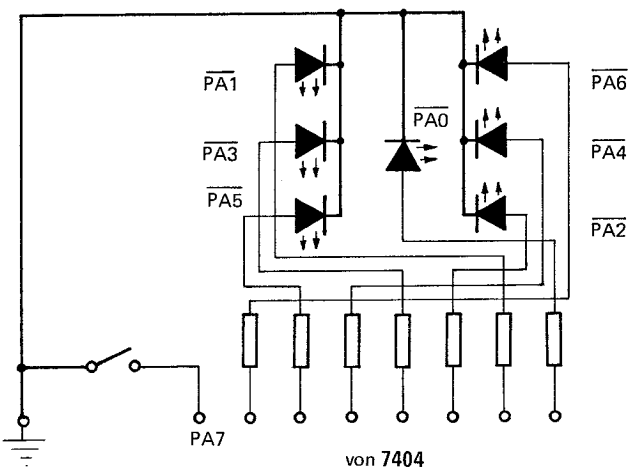
Die Experimentierschaltung kann durch Stecken leicht aufgebaut werden

Bei diesem Programm sind PN PA0 – PA6 Output und PIN PA7 als Input geschaltet. Wir steuern also von außen, ohne die Tastatur zu betätigen, ein Programm.

Legen wir an PA7 Masse, so entspricht es einem log. „1“, keine Verbindung entspricht log. „0“. Diese Schalterabfrage ist im Programm eingebaut.

Sobald der Taster T gedrückt wird, zählt die CPU in einem Zähler die Zahlen 0 bis 5 zyklisch durch. Wird der Taster losgelassen, verbleibt

Zur Demonstration sollten Sie allerdings die LEDs folgendermaßen anordnen:



Würfel

die CPU in einer Warteschleife, um bei erneutem Drücken von T weiterzuzählen. Gleichzeitig wird die Zufallszahl auf dem LED-Feld angezeigt. Wenn T gedrückt wird, leuchten also alle LEDs auf. Wenn T in den Ruhezustand zurückgeht, leuchtet ein bestimmtes Würfelmuster auf.

Zu der Bit-Kombination der 6 Zahlen noch ein paar Überlegungen:

Bei jeder Würfelzahl soll ein bestimmtes LED-Feld sichtbar sein. Unter Berücksichtigung der Inverter ergeben sich folgende Werte. Die Hex-Zahl ist also der Wert, der auf die Outports PIN PA1 – PA7 gegeben wird.

Würfelzahl	Logische Zustände an den A-Ausgängen des KIM-1							Hex-Code
	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
—								—
1	1	1	1	1	1	1	0	7E
2	1	1	1	1	0	0	1	79
3	0	0	1	1	1	1	0	1E
4	0	0	1	1	0	0	1	19
5	0	0	1	1	0	0	0	18
6	0	0	0	0	0	0	1	01

Würfelspiel

E1	LDA	IM	FF		200	A9	FF	
	STA	ABS	01	17	202	8D	01	17
	LDX	IM	05		205	A2	05	
E2	LDA	ABS	00	17	207	AD	00	17
	AND	IM	80		20A	29	80	
	BNE	R	E2		20C	D0	F9	
	LDA	ZPX	10		20E	B5	10	
	STA	ABS	00	17	210	8D	00	17
	DEX	IMP			213	CA		
	BPL	R	E2		214	10	F1	
	BMI	R	E1		216	30	E8	
					0010	7E		
					0011	79		
					0012	1E		
					0013	19		
					0014	18		
					0015	01		

Bei dem nun folgenden Programm müssen wir unseren Hardwareaufbau verlassen.

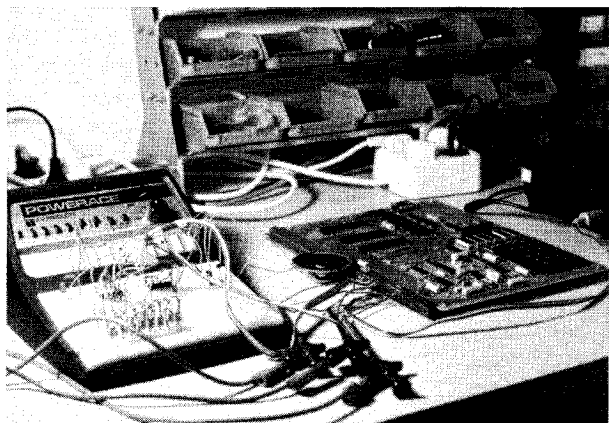
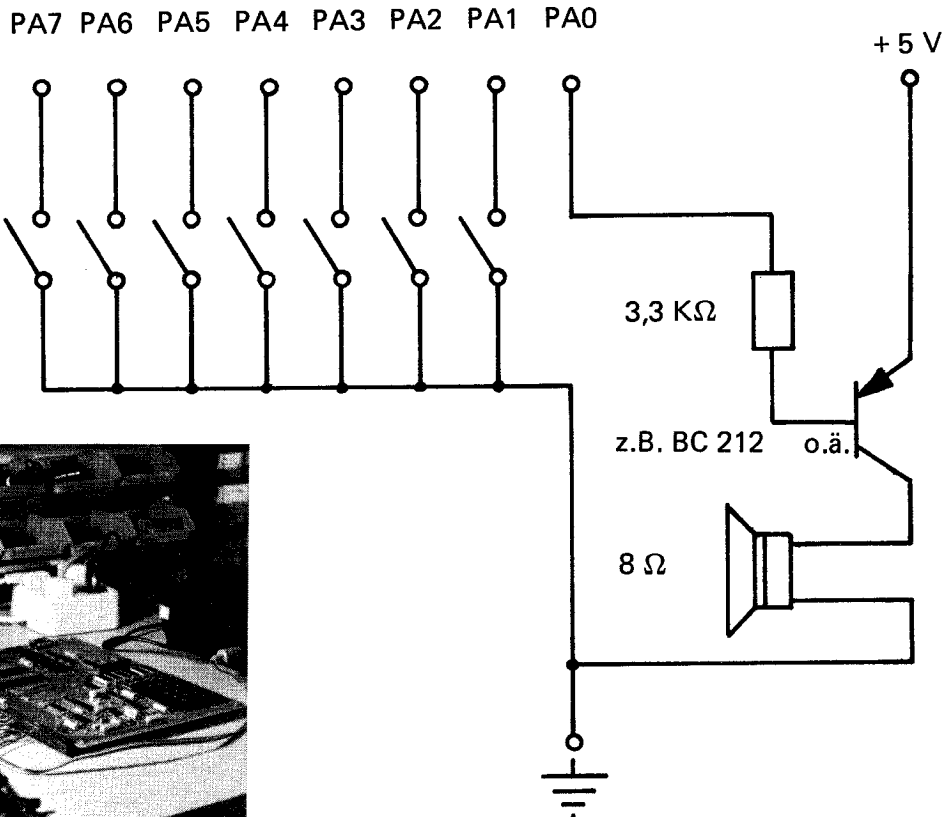
Unser Tonerzeuger ist ein Rechteckgenerator, dessen Frequenz einstellbar sein soll. Es sollen 7 verschiedene Töne erzeugt werden.

Wir benötigen nun 1 Lautsprecher, 8Ω, 1 x 3,3 KΩ, 1 BC 219 oder ähnlich.

Wenn eine der Tasten gedrückt wird, erklingt der entsprechende Ton; wird keine Taste gedrückt, erklingt kein Ton. Drückt man zufällig 2 Tasten, so soll der höhere Ton erklingen.

Wir wollen mit unserem KIM-1 Töne erzeugen.

Hardwareaufbau:



Gesamtaufbau: KIM, Lautsprecher und Powerace

Ton:	LDA	IM	01		200	A9	01	
	STA	ABS	01	17	202	8D	01	17
E1	LDA	ABS	00	17	205	AD	00	17
	CMP	IM	FF		208	C9	FF	
	BEQ	R	E1		20A	F0	F9	
	CLC	IMP			20C	18		
	LDY	IM	00		20D	A0	00	
E2	INY	IMP			20F	C8		
	ASL	AC			210	0A		
	BCS	R	E2		211	B0	FC	
	LDX	ABY	FF	02	213	BE	FF	02
	INC	ABS	00	17	216	EE	00	17
E3	LDY	IM	0F		219	A0	0F	
E4	DEY	IMP			21B	88		
	BPL	R	E4		21C	10	FD	
	DEX	IMP			21E	CA		
	BPL	R	E2		21F	10	F8	
	JMP	ABS	E1		221	4C	05	02

Quellenverzeichnis und Literaturhinweis:

1. R6500 Mikrocomputer System Programming Manual von Rockwell	300	07
	301	06
	302	05
2. KIM-1 Microcomputer Module User Manual von Rockwell	303	04
	304	03
	305	02
3. KIM-1 ist eingetragenes Warenzeichen / von MOS Technology	306	01
	307	00
4. 6502 Microcomputerprogrammierung von Peter Heuer, Hofacker-Verlag		

E. Schmid

