Experimente für Anfänger mit KIM-I

Experimente für Anfänger mit KIM-1

In unserer letzten Folge haben wir uns mit der Programmierung des Ein-/Ausgabeports beschäftigt. Dabei haben wir Interessante Experimente durchgeführt, bei denen die Ports für Lauflicht, Würfel oder auch einen Rechteckgenerator verwendet wurden.



Als nächste Stufe unserer KIM-Einführung wollen wir ausschließlich das Display (Anzeige) benutzen.

Grundsätzlich wird dabei zwischen dem Segment und dem Digit (Ziffernanzeige) unterschieden, Das KIM-System hat 6 Digits. Jedes Digit hat 7 Segmente.

Digit und Segmente lassen sich getrennt voneinander ansteuern.

Wichtige Adressen zur Ansteuerung:

0009	1. Digit
000B	2. Digit
000D	3. Digit
000F	4. Digit
0011	5. Digit
0013	6. Digit
00F9	Input-Buffer Hi-Display (56. Digit)
00FA	Pointer Low - Display (34. Digit)

00FB	Pointer High-Display (12. Digit)
1740	SAD Datenregister (Segment)
1741	SADD Datenrichtungsregister (Seg.)
1742	SBD Datenregister (Digit)
1743	SBDD Datenrichtungsregister (Digit)

Wichtige Unterroutinen mit Anfangsadresse

1F19	SCAND Ausgabe auf 7-Segm.Display
1 F 1 F	SCANDS Aufleuchten des 7-Segment
	Displays
1 F E 7	Tabelle - Hex in 7 Segment

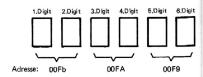
Erklärung:

Die Unterroutine SCANDS wird nur in Zusammenhang mit der Displayansteuerung durch die Adressen 00F9, 00FA, 00FB benutzt.

Die Unterroutine Tabelle — Hex, in 7-Segment wandelt die Zahl, mit der in die Routine gesprungen wird, in diejenige Bitinformation um, die zur Anzeige auf dem 7-Segment Display benötigt wird, d. h. die Binärzahl 0000 0001 wird in 0000 0110 verschlüsselt. Nähere Erläuterung folgt.

Paarweise Digitansteuerung

DISPLAY



Größte Hex-Zahl, die in jede beliebige, frei verfügbare Adresse gespeichert werden kann, ist FF

Diese Information benötigt 2 Digits. Damit ist die paarweise Digit-Ansteuerung des Displays von Vorteil.

Dazu nun 2 Programmbeispiele:

Wichtig: Es ist immer darauf zu achten, daß

bei der paarweisen Digit-Ansteuerung die Unterroutine SCAND benötigt wird. In diese muß immer wieder gesprungen werden, um eine konstante Anzeige zu erreichen.

LDA	IM	12	200	A9	12	
STA	ZP	FB	202	85	FB	
LDA	IM	34	204	A9	34	
STA	ZP	FA	206	85	FA	
LDA	IM	56	208	A9	56	
STA	ZP	F9	20A	85	F9	
JRS	SCA	NDS	20C	20	IF	IF
JMP		E3	20F	4C	OC.	02

Erklärung in dem Programm:

E3

Durch jeweiliges Laden einer zweistelligen Hex-Zahl zwischen 00 \rightarrow FF in die Adressen der paarweisen Digitansteuerung des Displays wird im gewählten Beispiel die Zahl 12 34 56 pro-

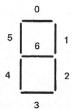
Wie schon erwähnt, wollen wir aber jedes Digit einzeln ansteuern. Wichtig dabei ist aber, daß damit verbunden auch die Segmente angesteuert werden müssen.

DISPLAY

	1.Digit	2.Digit	3.Digit	4.Digit	5.Digit	6.Digit
			1 1	1 1		1 1
				\Box		
Adres	se: 0009	000b	000d	000F	0011	0013

Man kann nun sehr schön diese Digits hintereinander ansprechen, da ihre Adressen die Differenz 2 aufweisen.

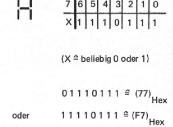
Wichtig: Spreche ich die Digits alleine an, so muß ich die Segmente, die aufleuchten sollen, im Programm bestimmen.



Die Zahlen geben die Reihenfolge der Binärzahl an, die das gewünschte Zeichen aufleuchten läßt.

Als Beispiel nun der Buchstabe A:

Folgender Code:



Dazu nun das Programmbeispiel:

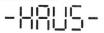
Erklärung:

Zuerst wird das Segmentdatenrichtungsregister auf Ausgang aller Segmente gesetzt. Es folgt die Ansteuerung des gewählten Digits. In dieses Digit wird nun die Information der Segmente gebracht.

	LDA		7F		200	A9	7F	
	STA	ABS	SADD	(1741)	202	8D	41	17
E1	LDA		09		205	A9	09	
	STA	ABS	SBD	(1742)	207	8D	42	17
	LDA		77		20A	A9	77	
	STA	ABS	SAD	(1740)	20C	8D	40	17
	JMP	ABS	E1		20F	4C	05	02

Wir wollen nun das ganze Display zur Anzeige eines Wortes benutzen. Man könnte auch nur 1 Segment auf jedem Digit aufleuchten lassen.

Ich wähle dazu das Wort



Programm:

	LDA	7F			200	A9	7F	
	STA	ABS	SADD	(1741)	202	8D	41	17
E1	LDA	IM	09		205	A9	09	
	STA	ABS	SBD	(1742)	207	8D	42	17
	LDA	IM	40		20A	A9	40	
	STA	ABS	SAD	(1740)	20C	8D	40	17
	JSR		VERZ		20F	20	00	03
	LDA	IM	OB		212	A9	OB	
	STA	ABS	SBD	(1742)	214	8D	42	17
	LDA	IM	76		217	A9	76	
	STA	ABS	SAD	(1740)	219	8D	40	17
	JSR		VERZ		21C	20	00	03
	LDA	IM	0D		21F	A9	0D	
	STA	ABS	SBD	(1742)	221	8D	42	17
	LDA	IM	77		224	A9	77	
	STA	ABS	SAD	(1740)	226	8D	40	17
	JSR		VERZ		229	20	00	03
	LDA	IM	0F		22C	A9	0F	
	STA	ABS	SBD	(1742)	22E	8D	42	17
	LDA	IM	3E		231	A9	3E	
	STA	ABS	SAD	(1740)	233	8D	40	17
	J\$R		VERZ		236	20	00	03
	LDA	IM	11		239	A9	11	
	STA	ABS	SBD	(1742)	23B	8D	42	17
	LDA	1M	6D		23E	A9	6D	
	STA	ABS	SAD	(1740)	240	8D	00	17
	JSR		VERZ		243	20	00	03
	LDA	1M	13		246	A9	13	
	STA	ABS	SBD	(1742)	248	8D	42	17
	LDA	IM	40		24B	A9	40	
	STA	ABS	SAD	(1740)	24D	8D	00	17
	JSR		VERZ		250	20	00	03
	JMP	ABS	E1		253	46	05	02
VERZ	LDY	ZP	t		300	A4	01	
E2	LDX	ZP	Z		302	A6	02	
E3	DEX	IMP			304	CA		
	BNE	R	E3		305	D0	FD	
	DEY	IMP			307	88		
	BNE	R	E2		308	D0	F8	
	RTS	IMP			30A	60		

Beim Programmablauf ergab es erhebliche Schwierigkeiten. Die letzten drei Zeichen waren nicht zu erkennen. Sie bildeten ein nicht gewünschtes Zeichen. Die Digits reagierten nicht mehr genau auf die schnelle Ansteuerung durch das Programm. Deswegen war das Erstellen eines Verzögerungsunterprogrammes nötig. Bei maximaler Verzögerung zeigte sich dann auch ein sehr schöner Effekt. Das Aufleuchten der Digits bildete eine Laufschrift. Natürlich kann man die Verzögerung auch so gestalten, daß eine konstante Anzeige erreicht wird.

Merke: Verzögerungsprogramm immer einbauen

Wie wir sehen, ist dieses Programm sehr aufwendig, da wir jedes Digit und die entsprechenden Segmente getrennt ansteuern müssen.

Um sich diese Mühe zu sparen, bietet sich die Unterroutine Tabelle-Hex in 7-Segment an.

Diese Routine liefert sofort den richtigen Binärcode für das Digit. Es lassen sich aber nur Hex-Zahlen umwandeln, keine anderen Segmentkonfigurationen (z.B. H,t u.s. w.)

Nachfolgendes Programm zeigt die Anwendung dieser Routine.

Das Programm arbeitet als Dezimalzähler. Von 0 beginnend bis 999 in Schritten zu 1. Ist 999 erreicht, beginnt es von vorne

999 erreicht, beginnt es von vorne.									
EO	LDA	IM	BF			200	A9	BF	
	STA	ZP	AA			202	85	AA	
E1	LDA	IM	1F			204	A9	1F	
	STA	ABS	43	17		206	8D	43	17
	LDA	IM	12			209	A9	12	
	STA		42	17		20B	8D	42	17
	LDY	IM	03			20E	A0	03	
E2	LDX	ZP,Y	E6			210	B6	E6	
	LDA	X	E7	1F		212	BD	E7	1F
	STA		40	17		215	8D	40	17
	LDA	IM	7F			218	A9	7F	
	STA		41	17		21A	8D	41	17
	LDA	IM	FF			21D	A9	FF	
E3	SBC		01			21F	E9	01	
	BNE		E3			221	D0	FC	
	CLD					223	D8		
	STA		40	17		224	8D	40	17
	DEC		42	17		227	CE	42	17
	DEC		42	17		22A	CE	42	17
	DEY					22D	88		
	BNE		E2			22E	DO	EO	
	DEC		AA			230	C6	AA	
	BNE		E1			232	D0	DO	
	INC		E9			234	E6	E9	
	LDA	IM	0A			336	A9	0A	
	CMP		E9			238	C5	E9	
	BNE		EO			23A	D0	C4	
	STY		E9			236	84	E9	
	INC		E8			23E	E6	E8	
	LDA	IM	0A			240	A9	0A	
	CMP		E8			242	C5	E8	
	BNE		EO			244	D0	BA	
	STY		E8			246	84	E8	
	INC		E7			248	E6	E7	
	LDA CMP	IM	0A			24A	A9	0A	
	BNE		E7 E0			24C 24E	C5	E7	
	STY		E7				D0	BO	
	JMP		E0			250	84	E7	00
	JIVIF		EU			252	4C	00	02
						E7		00	
						E8		00	
						E9		00	
						AA		XX	
						~~		^^	

Wir wollen nun ein praktisches Beispiel anwenden. Der Inhalt des Akkumulators soll auf dem Display angezeigt werden. Dies bedeutet, daß wir Rechenoperationen einfacherer Art zur Anzeige bringen.

Wichtig ist, wie aus vorherigen Beispielen ersichtlich, daß wir zeitliche Verzögerungen in das Programm einbauen. Der Grund liegt in der Zykluszeit der CPU.

Ich wähle dazu ein Hauptprogramm, das nur Zu-

fallszahlen erzeugt. Dies will ich nun zur Anzeige bringen.

Im KIM-1 System lassen sich Hex-Zahlen von 00 bis FF im Akku abspeichern. Dies bedeutet, daß wir 2 Anzeigen dazu benötigen:

Ich wähle daher die paarweise Ansteuerung des Displays, um die langwierigen Überlaufabfragen zu umgehen.

Gleichzeitig wurde das Digitdatenregister auf Ausgang des 5. und 6. Digits gelegt. Damit werden nur die benötigten Digits angesteuert. Die anderen sind nicht erleuchtet.

E0	LDA	IM	08		200	A9	80	
E1	PHA				202	48		
	LDA	ZP	00		203	A5	00	
	R06	Α			205	2A		
	EOR	ZP	00		206	45	00	
	ROL	Α			208	2A		
	ROL	Α			209	2A		
	ROL	ZP	01		20A	26	01	
	ROL	ZP	00		20C	26	00	
	PLA				20E	68		
	CLC				20F	18		
	ADC	IM	FF		210	69	FF	
	BNE		E1		212	DO	EE	
	LDA	ZP	00		214	A5	00	
	STA	ZP	F9		216	85	F9	
	LDA	IM	FF		218	A9	FF	
	STA	ZP	AA		21A	85	AA	
E2	JSR		SCANI	DS	21C	20	IF	IF
	LDA	IM	OF		21F	A9	OF	
	STA	ABS	43	17	221	8D	43	17
	DEC	ZP	AA		224	C6	AA	
	BNE		E2		226	D0	F3	
	JMP		EO		228	4C	00	02
					AA	xx		
					AA	^^		

