Alphatronic P2 ● *Inside 1* Keyboard Display Portliste (update 10-jun-.2016)

Mit dieser kleinen Serie "**Alphatronic P2 ● *Inside***" , zeige ich viele interne Hard- und Softwarestrukturen, Funktionen und den Zusammenspiel mit dem MOS. Oft habe ich kleine Beispiele dargestellt mit konkreten Daten und Fakten wie man die Abläufe nachvollziehen kann und verstehen kann.

Es gibt viele nützliche Hinweise auf meiner kleinen Web-Site (nicht schön - aber fast alles zur P2) und weiteren Informationen. Die Bereiche Keyboard, Display und die Port-Liste werden hier abgehandelt.

http://www.waltroper-aufbruch.de/AlphatronicP2.php (zum Verzeichnis)

Dort finden Sie Unterlagen und Programme sowie viele Hilfswerkzeuge, Anleitungen um eine **Alphatronic P2** von Triumph Adler oder ähnliche Maschinen zum „**ERLEBEN**" mit Software sich selbst zu besorgen. Fast aus dem Nichts, nur mit einem PC-Internet und über meine kleine Web-Site. Dazu benötigen Sie nur ein V24 – Drehkabel schon können Sie loslegen. Ach ja, eine Alphatronic P2 haben Sie doch oder?
Viel Erfolg.

Dipl.-Ing. H. Wiertalla

- So arbeitet die Tastaturtabelle mit dem MOS
- Wie geht das, bei einer Alphatronic P2 mit der MOS Checksumme in EPROM's?
- Eine TASTE (key), der Weg bis zur Anzeige (display).
- Ein kokretes Beispiel für eine Taste
- Scan – CODEs Übersicht der Tastatur - mit dem Keyboard - controller
- Den eigenen Zeichensatz auf dem Display anzeigen und was bedeutet das
- Wie arbeitet das Display-Interface und wo finde ich dazu mehr Informationen
- Eine eigene TAS-Tabelle einbauen. Hier zeige ich genau wie?
- Oder Hilfe, kleine Operation im EPROM und Teil-Schaltbild – zur Spannungsversorgung
- Zeichensatz - selbst anzeigen, wie mache ich das?

Die **verwendeten Ports** für eine „Alphatronic P2" , KISS

Baugruppe	**Funktionen**	**Basisport (hex)**	**Bedeutung**	**Buchsen**
CPU	V24 - Druckeranschluss	00h	DSR, RTS	I: 25 pol.
CPU	Drucker senden/empfang	SIM und RIM	Tx und Rx	
CPU	V24 – voller Anschluss	04h , und 05h	volle Schnittst.	II: 25 pol.
CPU	(EPROM, RAM)	78h	Banking switch	
Tastatur	Tastatureingabe	10h		
Floppy	Diskettensteuerung	50h		
Festplatte	Extern - Interface	80h	Steckplatz 9	III:37 pol.
Dyn.48 kB RAM	mit Adapter	78h	Banking switch	
Display	VIDEO Speicher	3000h	Bildspeicher	
Display	RAM 2kB	78h	Banking switch	

(to top)

Tastatur-Tabelle im MOS

Hier ist die interne Arbeits-Tastatur-TABELLE P) vom MOS als HEXA Dump.
Im EPROM 0800-0FFFh ist die P-CODE Tabelle im **Floppydisk –Driver** EPROM.
Ab der Adresse 0EF0 bis 0FEF (hex) befindet sich die TABELLE und eine Länge von 256 (100h) Byte. Dort ist der mittleren EPROM 2716 (typ **TMS**) auf der CPU Baukarte.

Der **Display- und Tastaturtreiber** befindet sich ab 1000h-17FFh in dem dritten EPROM.
Dazu den DISPLAY-TASTATUR Treiber als pdf bitte durchsehen. **Wer suchet – der findet.**

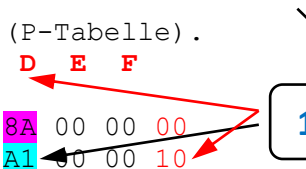
Der **D**ump-Teil vom EPROM prom02p.bin wurde per MOS erstellt.

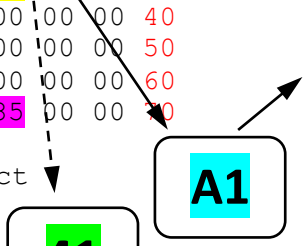
Taste →

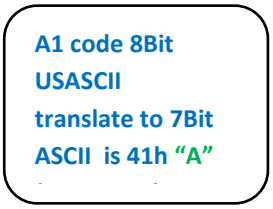
key

(to top)

```
.D0EE0,FFF(CR)   (CR=Return key),  .(red point MOS)
0EE0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Hier ist der Anfang zum nicht shift Bereich (P-Tabelle).
0000: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

0EF0 C4 B7 B3 B8 81 B1 00 00 51 A5 A4 A3 40 8A 00 00 00
0F00 52 B2 A6 B6 82 8C 00 00 53 B4 A7 A2 8B A1 00 00 10
0F10 54 BA A8 AE 00 C0 00 00 55 B5 AA AD 89 B9 00 00 20
0F20 56 A9 AB 4C 00 5C 00 00 57 AF AC 4E 8F C1 00 00 result->(0EF0h +1Dh) =A1
0F30 58 B0 BC 4D 84 00 00 00 59 BD BB C1 50 00 00 00 40
0F40 50 4B 43 FF 4E 00 00 00 FE 09 0D 51 4B 00 00 00 50
0F50 47 57 54 52 00 00 00 00 88 4F 4A C2 00 00 00 00 60
0F60 87 59 56 4D 00 00 00 00 86 58 55 53 00 85 00 00 70

          If only 7Bit in P-Range used direct
```








```
Hier ist der shift Bereich.    (to top)
0080:
0F70 C4 F7 F3 F8 81 F1 00 00 41 E5 E4 E3 40 8A 00 00
0F80 42 F2 E6 F6 82 80 00 00 A0 F4 E7 E2 8B E1 00 00
0F90 44 FA E8 EE 00 C0 00 00 45 F5 EA ED 89 F9 00 00
0FA0 46 E9 EB 5B 00 5E 00 00 4F EF EC 5A 8F C1 00 00
0FB0 48 F0 FC BF 84 00 00 00 49 FD FB C1 50 00 00 00
0FC0 5D 4A BE FF 4E 00 00 00 5F 09 0D 51 4B 00 00 00
0FD0 E0 57 54 52 00 00 00 00 88 4F 4A C2 00 00 00 00
0FE0 87 59 56 4D 00 00 00 00 86 58 55 53 00 85 00 00

          Interne Tastatur TABELLE P.)

0FF0 FF FF FF FF 03 10 40 FF 5F EF 07 34 32 44 0C 1E
.Signatur Promversion in ASCII->     " 4  2  D"
          Checksumme  ->                (αα ββ)     (to top)
```

Display

Inside ● Alphatronic P2! Wie geht das mit der Checksumme im MOS ?

Der Anfangswert ist 0. Jedes Zeichen im EPROM wird relativ von 0 bis 7FDh auf ein 16 Bit Register auf addiert. Dann wird der CHECK Wert der letzten EPROM (o.Beispiel) Speicherstellen verarbeitet. Der bisherige Wert ist z.B. als 16 Bit W = xx yy (xx = high yy = low). (alles hexa !)

W=	xx	yy	von Hand habe ich gerechnet W=	E1	F4
+	ββ	αα	**nun von oben eintragen->**	**+ 1E**	**0C** **vom prom02p.**
Result:	**C 00**	**00**	(C ist carry)	**C 00**	**00**

Das Ergebnis muss NULL sein. **Das war's – Summe gleich NULL, dann ok**! Wer z.B. etwas in einem EPROM ändert – sollte dann seine neue CHECK – Werte berechnen und dort von Hand ablegt. Dann klappt das im und mit dem MOS prima. Beim KISS und dem DS2069 wird das auch so gemacht.

Aber alle reden vom MOS EPROM CHECK – Test – aber hier zeige ich den Rechenweg.

W(16bit) := 00 00 setzen. Nun jedes EPROM-Byte (8 Bit) von 000h bis 7FDh als 16 Bit aufaddieren . Sagen wir es wurde W als **E1 F4 berechnet**. Nun bildet man W das Komplement **(to top)**
W (von E1 ist 1E, von F4 ist 0B , in hexa).

Zur Überprüfung F4 = 1 1 1 1 0 1 0 0 als Komplemet 0 0 0 0 1 0 1 1 das ist hexa 0B. Nun noch eine 1 zu rechnen, also 1E 0B + 1 := 1E 0C .

Hier wird 0C als **αα** und 1E als **ββ** wie oben abgelegt. Das war's, das als Summe ist 16 Bit zu 00 00.

Bei **meiner** Alphatronic P2 (von Triumph Adler ausgeliefert) ist die Berechnungsroutine Adr: 027Dh mit einem hex 0C9h ge**patch**t. Das bedeutet, die Berechnung wird gar nicht ausgeführt, weil 0C9h ein RET Befehl steht ! Eine EPROM – Prüfung findet nicht statt. Bei anderen TA - EPROMS wird die Checksumme aber berechnet.
(to top)

Beschreibung Key-Ablauf:

Die Funktionscode sind von C0h…CxH mit der Tastenebene (shift, unshift…) zum Behandeln vorgesehen. Für die Cursorbewegungen und die F1 … F6 Tasten, sind die Code 80h…8xh verbunden. Für die **Control-Taste**-Behandlung ist der Funktionscode **0FFh** oder auch mit anderen Funktionen (TA, KISS, DS2069 ev. anders) eingerichtet.

Eine TASTE, der Weg bis zur Anzeige,
wie geht das bei einer Alphatronic P2?

Eine Tastenkappe ist mit Symbolen (Buchstaben, Ziffern, Sonderzeichen bedruckt oder eingraviert. Entscheidend ist die Position einer Taste - was damit gemacht wird. Die Position einer Taste liefert einen sogenannten Scan-CODE ab. Vom Keyboard Interface wird der Scan-CODE (meist lückenhaft von 00h-7Fh) über eine interne **TAS-Tabelle P.) (**MOS ADR: 0EF0h, 100h Länge, unshift von 00h-07Fh und ab 80H-0FFh mit shift) als Index zu einem **Anzeige** - oder **Funktion**scode verwendet.

Ein Anzeigecode liefert aus der MOS-Tabelle den Wert (**fast** direkt)zur Anzeige auf dem Display.

Aus **historischen Gründen**, befinden sich in einigen Alphatronic P2 MOS-TABELLE die Arbeitscode als **USASCII 8-Bit**! Daher wird im MOS eine CODE-Wandlung nach ASCII (7-Bit Zeichensatz) per Software gemacht. **Es wird auch bei anderen MOS – Varianten ein direkter ASCII 7-Bit Code verwendet**.
(to top)

		0	10	20	30	40	50	60	70
		0	16	32	48	64	80	96	112
00	0	NUL	DLE	SP	0	@	P	`	p
01	1	SOH	DC1	!	1	A	Q	a	q
02	2	STX	DC2	"	2	B	R	b	r
03	3	ETX	DC3	#	3	C	S	c	s
04	4	EOT	DC4	$	4	D	T	d	t
05	5	ENQ	NAK	%	5	E	U	e	u
06	6	ACK	SYN	&	6	F	V	f	v
07	7	BEL	ETB	'	7	G	W	g	w
08	8	BS	CAN	(	8	H	X	h	x
09	9	HT	EM	)	9	I	Y	i	y
0A	10	LF	SUB	*	:	J	Z	j	z
0B	11	VT	ESC	+	;	K	[	k	{
0C	12	FF	FS	,	<	L	\	l	|
0D	13	CR	GS	-	=	M	]	m	}
0E	14	SO	RS	.	>	N	^	n	~
0F	15	SI	US	/	?	O	_	o	DEL

Für einige nationale Länder werden im Display Interface der EPROM (Zeichengenerator) bei einigen Codepositionen als Bild (character) geändert.

http://www.waltroper-aufbruch.de/pdf/Nationale_Alphatronic%20P2%20Character%20set.pdf (Beispiel)

Beispiel Anzeigecode: (to top)

TASTE "**A**" liefert den Scan-CODE **1D**h.
Die TABELLE mit Basis 0EF0H + 1DH Scan-CODE:= 0F0D ->der Inhalt ist dort A1h. Im 8-Bit USASCII ist das Zeichen "A", aber es wird immer dann auf den 7-Bit ASCII Rahmen gewandelt.
Abgeliefert wird **41h** also endlich ein Zeichencode für "A". Damit wird im Displayinterface aus einem EPROM das Punktbild "A" gemacht, was wir sehen.

Es gibt zusätzlich einen definierter Satz von **Funktionscode**. Das sind Shift, Control, Reset(Software), SM Schreibmaschinen-Mode, Cursor, Bell, usw. Mit einer MOS-Displayfunktion ist eine eigene Tastaturtabelle anzumelden und dann zu verwenden. (Eigenbau). **Dazu den DISPLAY-TASTATUR Treiber durchsehen.**

Inside ● **Hier zeige ich, wie eine eigene Tastaturtabelle eingebaut wird.** (to top)
Über die Display-Treiber Schnittstelle (+ MOS) ist ein Zugriff mit der Adresse **0DAh** über Funktionen per A (Accu) möglich. Mit dem Accu **A=7** und mit dem Register **HL = neue TABELLE** wird eingehängt. Sie müssen im 48 kB Speichermodus, das MOS und die neue TABELLE (absolute Adresse > 4000h) verfügbar machen.

```
.....
GET48                ; mit einem MACRO RAM Banking  (ausführlich im BANKING >dort Booten einer P2.)
MVI    A,7           ; Funktionscode laut Beschreibung
LXI    H,nTAS        ; Adresse wo die neue TABELLE ist ( selbst bauen, oder kopieren, oder als File laden und,++
CALL   0DAH          ; Einsprung per Unterprogramm (Displaytreiber –MOS Eintrittspunkte – Beschreibung)
GET64                ; z.B. im cp/m unter 64 kB > fertig
                     ;ab jetzt verwendet das MOS also auch  alle key- cp/m Aufrufe, die neue TABELLE !
```

Es gibt zwei Arten, um die Standard-EPROM-TABELLE zu erzwingen (wieder benutzen).
Die elegante Art ist mit dem Beispielaufruf – aber mit HL = 0 -, dann verwendet das MOS wieder die eigene Tabelle. Mit der Alphatronic P2 **HARD – Reset Taste** ist sowie so immer die EPROM Tabelle eingeschleust beim MOS. Hier ein Beispiel um im EPROM eine Taste umzulegen – gut überlegen!
(to top)

Scan –CODE from keyboard Alphatronic Px (to top)

(to top) **SCAN CODE Alphatronic P2**

Hier ist das **Keyboard – Interface** mit dem Buzzer ( Bell).

(to top)

Alphatronic P2 Keyboard – Interface /same as KISS, DS2069-1

(to top)

Zeichensatz selbst anzeigen, wie mache ich das?

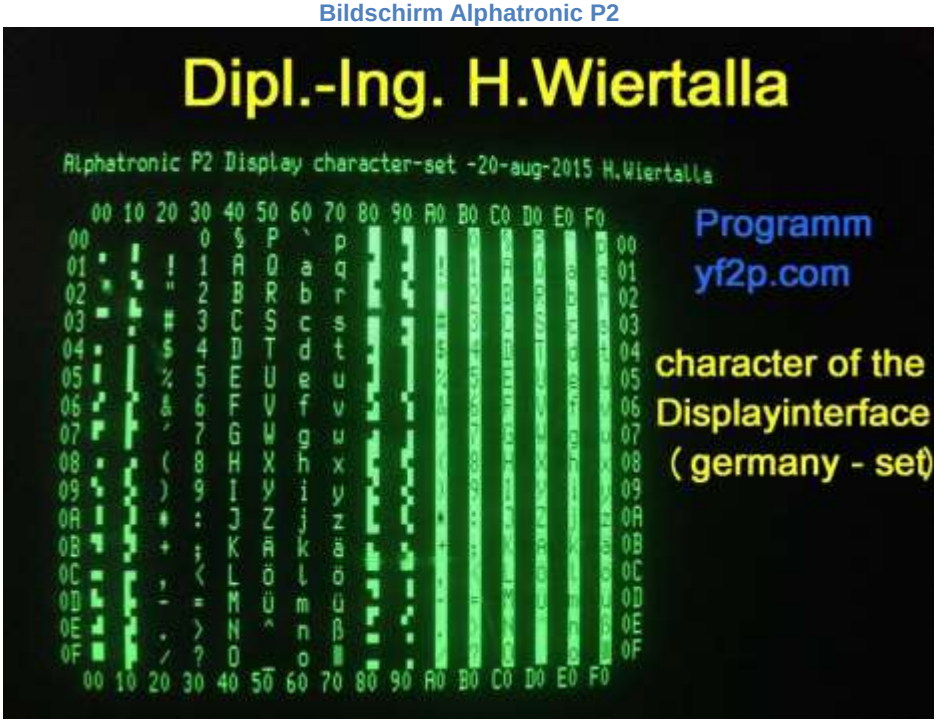
Hier sind mit dem kleinen **Programm yf2p.com** (Font anzeigen) jeweils die eigenen Zeichen wie diese im Display-Controller EPROM abgelegt sind.

Die CODEs 00-1Fh sind für eine einfache Blockgrafik nutzbar. Von 20h bis 7Fh ist der normale 7 Bit ASCII Rahmen. Je nach einer Tastaturschnittstelle werden die Zeichen bewertet. Dazu sehen Sie in meine pdf-Datei der Tastaturschnittstelle.

Aber im Displaycontroller EPROM werden oft national „Zeichen“ verwendet. Beachten Sie, wird ein Zeichen (character) mit dem Bit 2 hoch 7 auf eins gesetzt, so erzeugt der Displaycontroller ein inverses Zeichen.

(to top)

Bildschirm Alphatronic P2



7 Bit ASCII 00h-7Fh from Displaycontroller EPROM (national set). (to top)
Code 80h-FFh is inverse character from Controller (Hardware – Function).

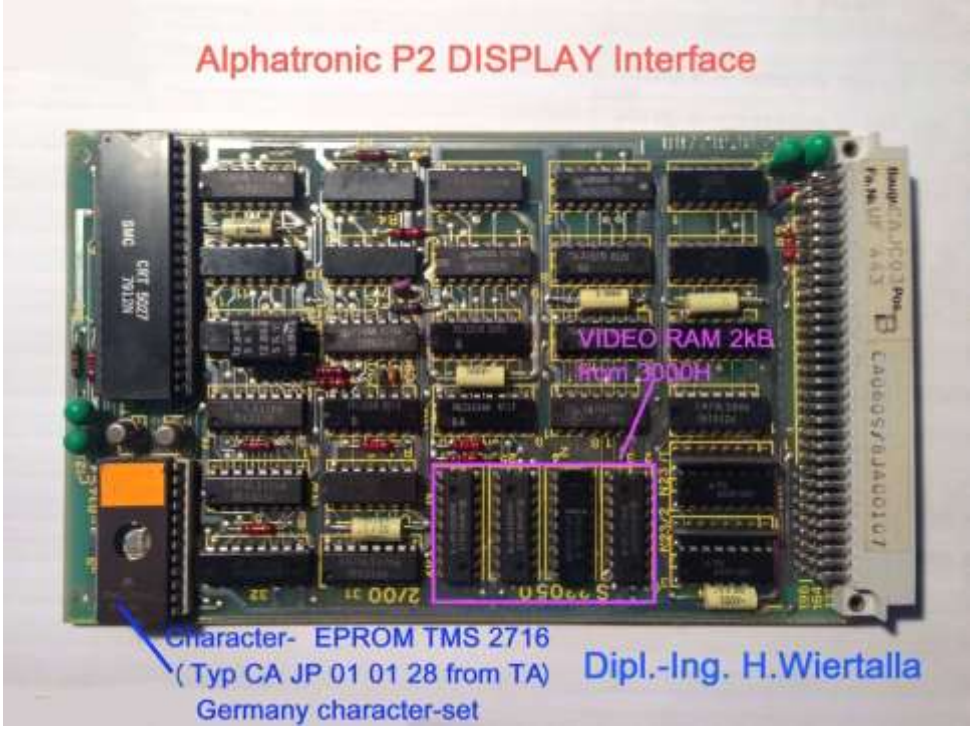
Use the program **yf2p.com**, is Font for your Alphatronic P2 or Px.

Schaltplan Display P2
Hier finden Sie den elektrischen Schaltplan für den von **Triumph Adler benutzten Alphatronic P2** verbauten Bilschirm mit einigen Grundeinstellungen dargestellt

http://www.waltroper-aufbruch.de/pdf/DISPLAY_P2_Manual_elekt_Schematic_hw_scann.pdf

. (to top)

Display – Interface mit dem VIDEO-RAM und dem EPROM mit dem Zeichensatz. Eine vollständige Dokumentation (siehe unten) enthält den Schaltplan und weitere Funktinsbeschreibungen. Mit Infos zur Programmierung also auch zur Initialisierung dort.
(to top)



http://www.waltroper-aufbruch.de/pdf/SKS_BC10_Schalt_Softwareunterlagen_auch%20Basis%20P2_Displaycontroll.pdf
Here is a programing part in this document.

Please check for a new version of this or other pdf's over this WEB-site.

Hilfe, eine TASTE geht dauern nicht!

Trotz allen guten Ratschlägen - geht nichts mehr.
Für Könner (-löter): HALL-Geber defekt (das ist das Element unter der Tastenkalotte)- ev. ein Austausch ist möglich.

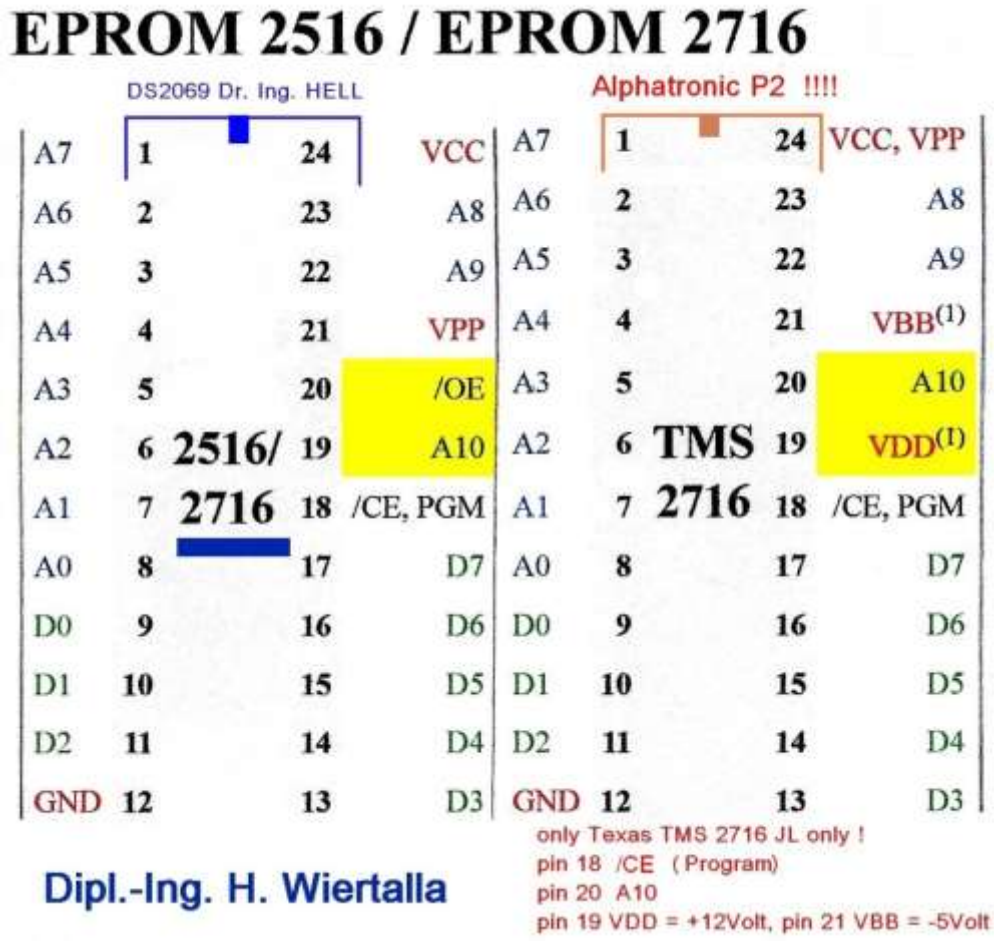
Oder eine kleine Operation, das geht so. Als Beispiel.

Eventuell wäre eine kleine Operation im EPROM 800h-0FFFh (P-Taste) erfolgreich. **Also z.B. die Taste "G" ganz wichtig – ist tot**. Unter dem Scancode **1A**h ist **A7**h („**G**" 8Bit USASCII) , und relativ zur P-TAB Anfang (000) + 1Ah + 80h (shift range) **9A**h steht **E7**h. Als Beispiel modifiziert man die P-TAB unter der TASTE „**F1**" den Ort **7D**h (scan-code) mit **A7**h und **FD**h mit **E7**h. Oder eine andere Taste (scan-code) verwenden.

Besitzt man einen Eprommer (auch noch für TMS 2716 beachten !!!) kann ein geänderter EPROM helfen. Bitte noch die Checksumme in diesem EPROM vor dem Brennen eintragen.
Jetzt über „F1" erreichen Sie „G" / „g".(shift beachten)

(to top)

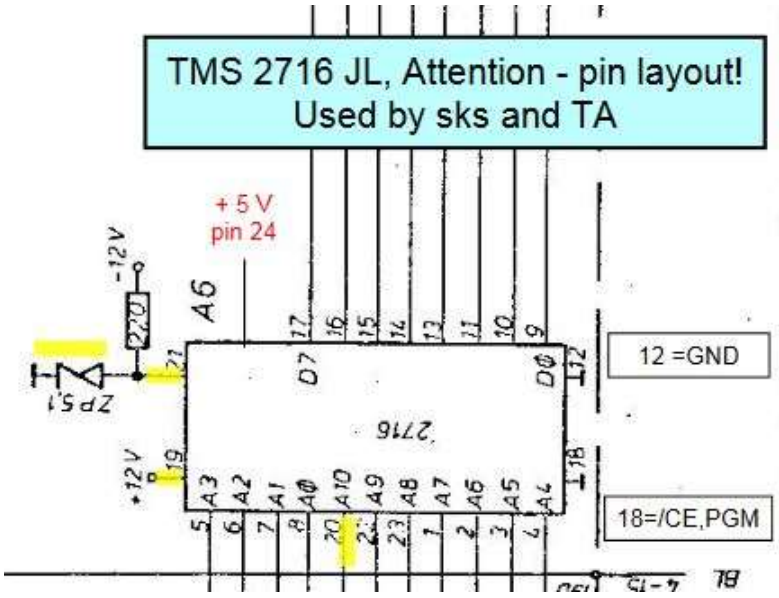
Bitte beachten Sie, der
EPROM von TMS 2716 ist mit den Spannungen von **+5 Volt, -5Volt und +12 Volt** zu versorgen.
Auch beachten auf A10 und die Texas PINs. (to top)



(to top)

Teil- Schaltbild TMS 2716 Beschaltung
Hier ist ein Auszug des Schaltbildes vom **Display – Interface** (pdf mit Programmparameter dort)

Hier sieht man welche Gleichspannungen erforderlich sind.



(to top)