

MTX *User-Club Deutschland*

Info 43
10.07.1991

Zweck: Zusammentragen und Austausch von Tips & Tricks u.s.w., Hilfestellung bei allen möglichen Problemen, Aufbau einer Programmbibliothek und Basteln von Hardware-Erweiterungen.

Programme (nur **Selbstgeschriebenes**): Tausch von kurzen und einfachen Routinen. Gute Programme (mit Dokumentation) können über den Club an alle Mitglieder verkauft werden. Wer solche Programme an uns schickt erhält ggf. Verbesserungshinweise und eine Besprechung im Info.

Mitglied kann jeder werden! Keine Beitragsgebühr! Anmeldung kostet DM 1.-.

Verpflichtungen: Einsendung unseres Anmeldeformulars.

Bitte: Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, Programmen, Beiträgen zum Info, Hinweisen auf preiswerte Hard- und Software, und was noch so zusammenkommt und andere interessieren könnte.

Club-Info, unser Blatt, verschicken wir ca. 8-wöchentlich. Inhalt ist alles was uns über den MTX/FDX (ohne Copyright) in die Hände fällt. Es kostet nicht über DM 12.- je Exemplar. Jeder kann dazu Beiträge liefern und hier gratis Kleinanzeigen veröffentlichen.

Kosten: Wir berechnen ausschließlich Selbstkosten und verschicken nichts, wenn Ihr persönliches Guthaben nicht reicht! (s.u.)
Schüler, Studenten, Auszubildende, Grundwehrdienstleistende, Rentner und Arbeitslose erhalten einen Nachlaß von 40% auf die zukünftigen Infos nach Einsendung einer entsprechenden Bescheinigung für deren Gültigkeitszeitraum.

Geld/Konto: Für jedes Mitglied führt Herbert zur Nedden ein Konto, von dem die jeweils entstehenden Kosten abgehen. Der Kontostand wird bei jeder Sendung mitgeteilt (**er steht über der Anschrift**) und kann selbstverständlich jederzeit erfragt werden! Wir verschicken nur gegen Vorkasse!

Einzahlungen bitte auf's Club-Konto: (oder V-Scheck)
(**Absender!** incl Name und Anschrift bitte nicht vergessen!)
Postgiroamt Hamburg, BLZ 200 100 20,
Herbert zur Nedden, Sonderkonto C, Nr. 3480 00-200

Kontaktadressen:

Herbert zur Nedden
Alte Landstraße 21
2071 Siek
(04107) 99 00

Hans Gras
Statenhoek 49
NL 1506 VM Zaandam
(0031-75) 17 49 91

Telefon-Sprechzeiten

Herbert zur Nedden: Do 18 - 21 Uhr, Sa 9 - 14 Uhr
(Etwas klingeln lassen oder nochmal versuchen!)

Inhaltsverzeichnis

C l u b	
Clubtreffen	diese Seite
Tips	Seite 2
Mitgliederliste	Anlage
N e w W o r d / W o r d S t a r	
Hartmut Traber, 5270	Seite 3
Erik d'Hondt, B-9320	Seite 4
Einige Anmerkungen zu WordStar	Seite 6
S o f t w a r e	
KLIX-Moni	Seite 7
T u r b o - P a s c a l	
Directoryzugriff	Seite 8
S o f t w a r e	
DIREX	Seite 10
H a r d w a r e	
Centronics-Schnittstelle	Seite 11
IBM-Tastatur	Seite 12
K o m i k	
Der Fortschritt (vom SVI/MSX-Club Deutschland)	Seite 17

Preis für dieses Info: DM 6.60

Neue PD (Herbert zur Nedden, 2071)
 KLICK.017 ist fertig. Es ist zwar in kB gemessen wenig drauf, aber dafür etwas wirklich schönes: Maus- & Tastatur-Treiber für Genius-Maus und Escom-Tastatur und **KLIX-Moni**! Ja, es gibt ihn wirklich, Olaf's Moni im KLICK... nein, eigentlich stimmt das nicht, da KLIX-Moni viel besser und mehr als Moni ist!

Clubtreffen (Herbert zur Nedden, 2071)
 Irgendjemand hat leider vergessen all das zu bezahlen, was er/sie konsumiert hat. Offen sind zwei Orangensaft und sechs Bier geblieben! Wer von den Teilnehmern hat denn O-Saft getrunken und nicht bezahlt; wer hat sich bei seinen Bierchen verzählt??? Bitte bei mir melden, damit ich das ausgelegte Geld zurückbekomme!

Kontostand (Herbert zur Nedden, 2071)
 Eine rote Markierung auf dem Umschlag bedeutet, daß er zu niedrig ist.

Anzeigetexte samt Absender bitte schriftlich an Herbert zur Nedden!

V E R K A U F (Preise sind i.a. ohne Porto & Verpackung)

Herbert zur Nedden, Alte Landstr. 21, 2071 Siek, 04107 - 9900:

--> Interessiert Dich einer der von mir angebotenen Posten: Mach ein Angebot!
Meine hier genannten Preise sind nicht unbedingt unumstößlich!
(Neue/geänderte Posten haben einen * statt des - vorne weg)

- Evtl. Grünmonitor: DM 100.-
- Rikadenki Plotter RY21, VB DM 1500,-
Flachbettplotter, DIN A4, Aufnahme für Rotringstifte, incl. Handbuch und Schaltplan, 8085 CPU (Z80-aufwärtskompatibel), Centronics-Schnittstelle. Positioniergenauigkeit: 0,1 mm, Zeichengeschwindigkeit 200 mm/s, 19 Kommandos wie u.a. Kreise, Kreisbögen, Rechtecke, verscheiden gestrichelte Linien, Textausgabe in 4 Richtungen und verschiedenen Größen, absolute und relative Koordinaten, Markierungen auf Linien und Graphen. Kann in Textmodus gesetzt werden, um als Drucker zu arbeiten.
- Ich habe RAMs für 768kB-Erweiterung!
- dBASE Version 2.41: DM 100.-
- NewWord Version 2.02 DEUTSCH, im Tausch gegen Original-NW-Diskette: DM 40.-
- Original Microsoft-BASIC-Lizenz incl., M80/L80 Assembler/Linker, Handbuch, auf dem MTX lauffähig: DM 180.- (Microsoft liefert dieses Teil nicht mehr aus!)
- NonDoc-Editor für CP/M incl. Source, zeilen- und bildschirmorientiert, Macro-Fähig, mit Handbuch im Ringordner: VB DM 40.-
- Z80ASM (Club-PD Z80-Ass.): Listing und Handbuch, ca. 1.4 cm DIN A4: VB DM 5.-
- Infos 11-36, gebraucht VB DM 60.- frei Haus
- Bücher
 - Programmierung des Z80, Rodney Zaks, 606 Seiten: DM 45.-
 - Mikroprozessor Interface-Techniken, A. Lesea/R. Zaks, 425 Seiten: DM 30.-
 - Operationsverstärker Anwendung, 164 Seiten, DM 10.-
 - ECA-Tabelle ttl-IC's, endet im Bereich der 74xx400-er: DM 20.-
 - ECA-Tabelle dat 1: Transistoren A..BUY: DM 5.-
 - ECA-Tabelle tht: Thyristoren, Triacs, ...: DM 5.-
 - * Turbo Pascal, 3. Auflage von Herschel, Oldenbourg-Verlag: DM 15.-
- Einbau-Drehspulmeßgerät 0-50uA: DM 10,-
- Solange der Vorrat reicht:
 - MTX-Tasten je DM 1.-, Tastenkappen je DM -.50
 - EPROMS 2564 für je DM 15.-
 - Dynamische RAMs 4116 (VRAMs) 8 Stück: DM 25.-
 - Dynamische RAMs 3732 (32k x 1Bit) 8 Stück: DM 1.50
 - Statische RAM's 2k x 8 Bit (6116): je DM 2.-
 - TTL-IC's: 74LS175, 74LS368, 74LS173, 74LS158, 74LS258 je DM 0.50;
74LS10, 74LS11, 74LS21 je DM 0.30
 - Original-Memotech-Spielecassetten: Toado, Kilopede, Knuckles, Draughts, Reversi, Snappo, Blobbo, Utilities, Demo, StarCommand je DM 4.-;
 - 10 Disketten FUJI HD 5 1/4", gebraucht & o.k. DM 35.-, Originalverpackt: DM 60.-

Joachim Keiser, Brinkgarten 23, 4925 Kalletal:

SDX, Aluminiumprofilgehäuse mit Schaltnetzteil, Lüfter und sämtlicher Elektronik, Platz für ECB-Bus (8 Steckplätze), 512+64 kB RAM, Hardware-Uhr, erw. 80Zeichen-Karte, neues 80Zeichen-PROM, 2 getrennte elektronisch umschaltbare Druckerschnittstellen, Tastatur mit AlphaLock-LED, 2 Floppy-Laufwerke EPSON SD 540, RAM 6.1, NewWord, SuperCalc, dBASE, TurboPascal, div. CLUB.- und KLICK.-PDs, alle Infos, evtl. mit DMX80-Drucker, ohne Monitor: VHB DM 1500.-

Günther Franke, Leithfeld 11, Hs. 8, 2000 Wedel, 04103-84040 oder 040-381403:
MTX, FDX, Bernstein-Monitor, leicht defekt bis bootunfreudig: DM 350.-

Liebe Leserin, lieber Leser,

das Clubtreffen hat viel Spaß gemacht! Abgesehen von diversen MTX-Neuigkeiten und -Vorführungen wie z.B. Claudios MENU mit Mausunterstützung, dem Debugger KLIX-Moni oder einer schwarzen IBM-kompatiblen Tastatur von ESCOM an meiner seriellen Schnittstelle war der Z280-Rechner von Tillmann Reh unter CP/M+ zu sehen: Er ist ca. doppelt so schnell wie ein 8 MHz MTX. Die Gruppe der Teilnehmer war sehr bunt (das hatten wir lange nicht mehr in dieser Ausprägung): Anfänger, Neulinge, Softwarecracs, Hardwarecracs und 'Leute wie Du und ich' waren vertreten! Eigentlich war die Zeit zu kurz: Ich kam nicht dazu mir alles was zu sehen war anzusehen oder zu hören war anzuhören. Leider hat auch dieses Mal nicht jeder seine Zeche bezahlt! Näheres siehe unter dem Inhaltsverzeichnis oben.

Tillman Reh hat übrigens eine ECB-Karte layoutet, über die man eine AT-Bus-Festplatte anschließen kann. Warum? AT-Bus-Festplatten haben den Controller gleich mit auf der Platte, und daher kennt der Controller 'seine' Platte 100%ig. Solche Dinge, wie die Steprate usw. ist damit kein Thema mehr, da der Controller das selbst festlegt. Mal sehen, vielleicht lernt RAM 6.x auch noch die Bedienung der einen oder anderen AT-Bus-Festplatte.

Einige Programme fragen die Pfeiltasten ab, indem sie sich den Tastencode über KbdIn holen. Diese Programme sollten auf jeden Fall in Zukunft auch die Codes CTRL-E, -S, -D und -X akzeptieren! Z.B. die IBM-kompatiblen Tastaturen haben zusätzlich zum 10-er Block mit dessen Pfeiltasten noch separate Pfeiltasten; und letzere liefern direkt Codes wie CTRL-E usw. Besonders interessant wird die Sache dann allerdings, wenn ein Programm SHIFT-Pfeile abfragen will (z.B., wenn es darum geht, die Maustasten einzulesen, Claudio). Bei mir liefern die geSHIFTeten Pfeile CTRL-A, -F, -W und -Z. Bei anderen sicherlich andere Codes. Daher sollte bei solchen Gelegenheiten ein Programm eine gewisse Toleranz aufweisen...

*Euer
Herbert zur Nedden*

Club: Tips**Tip: Druckerreparatur**

(Gerhard Kulenkampff, 2000)

Das Ingenieur-Büro H. Voigt GmbH, Schlader Weg 8, 2803 Weyhe, 0 42 03-59 49 repariert Drucker zu guten Preisen.

Tip: K2Dos

(Jan Brederke, 2000)

PERFORM geht von einem 2k-CCP aus und stürzt deshalb unter K2Dos ab. Kurzfristige Abhilfe: Dicht hinter dem Label START: steht ein LD DE,2056. Daraus habe ich ein 2000H = 8kB gemacht, indem ich an Adresse 119h/11Ah die Bytes 00h und 20h gepatcht habe. Langfristig wäre es aber sinnvoller, wenn sich alle Programme, die es benötigen, die Größe des CCP aus dem Environment holen. Ich hatte nur noch keine Lust, PERFORM neu zu übersetzen.

Tip: K2Dos

(Herbert zur Nedden, 2071)

Einige Programme wollen den CCP nicht überschreiben, sei es nun, um ihn selbst zu verwenden oder auch um statt mit einem JP 0 mit einem einfachen RET enden zu können da das schneller ist. Leider glauben die alten CP/M 2.2- und viele der älteren ZCPR-Programme noch an einen CCP, der popelige 2kB kurz ist! Seit K2Dos ist das nicht mehr der Fall, so daß eben diese Programme seltsame Resultate produzieren. Solltest Du über so ein Teil stolpern, laß es mich bitte wissen - ich will versuchen, Abhilfe zu schaffen.

New Word / WordStar: Leserbrief Hartmut Traber, 5270

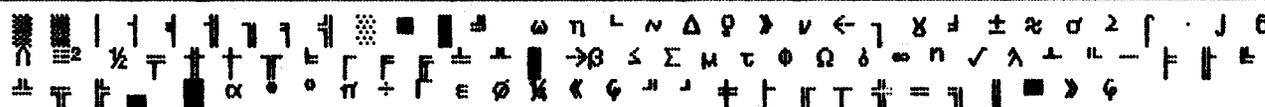
Hartmut Traber

Hohbeulstr. 8, den 27.09.90
5270 Gummersbach

Leserbrief und Besprechung

Tel.: 02261/65399 ab 16.00 Uhr
Tel.: 02261/80-200 ab 07.30 Uhr

Es sind weitere Fortschritte im MTX-User-Club zu vermelden!



DAS IST DER GRAPHISCHE ZEICHENSATZ VON ERIK d'HONDT. ES WIRD DIE GESAMTE ASCII-TASTATUR MIT SHIFT, ABER OHNE SHIFT-O VERWENDET. EINFACH NUR MIT ^PX DA UMSCHALTEN, WO MAN WILL UND NACH DEM EINFÜGEN DER ZEICHEN WIEDER MIT ^PX ZURÜCKSCHALTEN.

Empfehlenswert ist es, sich eine Schablone für die Tastatur zu fertigen. Die Masse der Zeichen kann man nicht im Kopf behalten!

Die Methode von Michael Keßler funktionierte ja auch schon hervorragend, aber die vielen ^ö und ^ä (für jedes von den weniger Zeichen) war mir doch etwas zeitaufwendig. Es ist komisch, auf meiner alten Schulschreibmaschine (mit Hebeln, Amboß, Federn usw.) ging's noch mit 10 Fingern, zwar nicht blind. Auf dem MTX benutze ich eigentlich nur die Zeigefinger. Wenn ich Linefeed mal brauche, wird es schon schwierig.

UMSO TOLLER FINDE ICH ERIKS METHODE !

Mein alter DMX80-Drucker wird hier zwar sehr gefordert. Man wird von ihm durch das erscheinende Schriftbild direkt zum Druckkopf-Reinigen aufgefordert. Wenn nur eine von den Nadeln hakt, sieht man es sofort. Allerdings ist mir etwas aufgefallen: Wenn man innerhalb z. B. eines Rahmens weitere Steuerzeichen unterbringt, versteht sie der DMX80 offenbar nicht richtig. Leserbrief usw. oben im Text ist von mir einmal mit ^PS unterstrichen. Offensichtlich erscheinen aber zwei Unterstriche? Außerdem werden bei mir manchmal die Zeichen nur dann richtig gedruckt, wenn vor dem gesamten Text ein ^B steht. Der Treiber müßte also ggf. noch überarbeitet werden, oder liegt es an meinem Drucker, NW oder dessen Installation? Im übrigen glaube ich auch, daß Erik bald WordStar ebenso aufgemöbelt haben wird wie New-
word! Mein Interesse bekunde ich hiermit. Ich bin gespannt, wie unser Info demnächst aussieht. Die noch mitgelieferten Treiber für Epson-Drucker konnte ich noch nicht ausprobieren, aber man hat sie ja mal schon! Für die prompte Lieferung des Eproms und der NW-Files zum konkurrenzlosen d'Hondt-Preis bedanke ich mich noch einmal hier im Info.

Hartmut Traber

NewWord / WordStar: Leserbrief Erik d'Hondt, B-9320

NewWord

VERSUS

WordStar

Nachdem NewWord Grafikzeichen gelernt hattete, war es die höchste Zeit um WordStar dasselbe anzulernen. Das hat schon Michael Keßler getan mit ^PÄ (Anfang Grafik) und ^PÖ (Ende Grafik). Mit diesen Kontrollcodes schaltet man um zum originalen 'Special Graphics Set' von Memotech. Nach einigen Patches von WordStar hatte ich die Grafikzeichen meiner 2732 Grafiks-Eprom auf Bildschirm. Aber,... in das Eprom war noch 2 Kb frei, ausreichend für einen zweiten Zeichensatz.

Und nun die Bildchirmattribute in WordStar:

Bit	Bedeutung	Monochromschirm
none	Normal text	Dim
0	Strike-Out	Invers
1	Warning & errors	Blinken
2	Marked text	Invers
3	Underlining	Unterstreichen
4	Subscripting	Invers
5	Superscripting	Invers
6	Highlighting	Helle Zeichen
7	Italic	Invers

Das einzige Attribut das in Frage kam, war Italic (^PY). Ich meinte mit ^PY auch Kursivschrift aufs Schirm zu bringen. Und ja, beim setzen das Attribut-Byte 06 80 (Grafiks) und der Escape-Sequenze 1B 'A' (Alternate) kann man den freien Platz in das 2732-Eprom erreichen. Ich habe mich also an die Arbeit gesetzt und mit viele Mühe und durchs Brennen der Eproms einen Kursiven Zeichensatz zusammengebastelt für das Bildschirm!

Nach das Patchen der VIDATT-subroutine in WordStar kommt nun mit ^PY das Kursivschrift auf Schirm UND auf Papier!

Während ich ans Patchen war, habe ich tiefgestelle und hochgestelle Text so angepaßt das mit ^PV oder ^PT dieser Text invers und in alternate Zeichensatz aufs Schirm kommt (nur mit dem originalen alternaten Zeichensatz von Memotech). Alles zusammen sind nun 4 Schriftarten gleichzeitig aufs Schirm: Standard, Alternate, Grafik und Kursiv.

NewWord / WordStar: Leserbrief Erik d'Hondt, B-9320

Leider funktioniert das Kursivschrift noch nicht mit NewWord, da Bit 7 der Bildschrimattribute nicht angewend werd durch NewWord (reserviert für zukünftige Erweiterungen). Hat jemand Lust und viel Zeit um NW.OVR auseinander zu holen und eine Subroutine einzufügen um mit ^PY Bit 7 zu setzen?

Ein kurze Vergleichung zwischen NewWord und WordStar:

NewWord	WordStar
3 Schriftarten aufs Schirm: Standard, Alternate und Grafik	4 Schriftarten aufs Schirm: Standard, Alternate, Grafik und Kursiv
Treiber für LQ-Drucker leicht anzupassen und optimal zu benutzen Idem für DMX80-Drucker (auch für Grafik) Grafikzeichen selbst zu definieren	Treiber nur teils an zu passen (nur mit eingebautem Grafikzeichen) Der DMX80-Drucker hat keine Chance um Grafiks auf Papier zu bringen! Nicht zu definieren.

Vorschlag: für den Grafik-Zeichensatz soll mann die IBM-Grafiks benützen die standard eingebaut sind in die meiste Druckers (ausgenommen der DMX80):

160	SP =	192	§ = L	224	' = α
161	! = í	193	A = ⊥	225	a = β
162	" = ó	194	B = T	226	b = Γ
163	# = ú	195	C = T	227	c = π
164	\$ = ñ	196	D = -	228	d = Σ
165	% = ñ	197	E = †	229	e = σ
166	& = @	198	F = †	230	f = μ
167	' = Q	199	G = †	231	g = τ
168	(= ;	200	H = †	232	h = φ
169) = →	201	I = †	233	i = θ
170	* = ←	202	J = †	234	j = Ω
171	+ = ½	203	K = †	235	k = δ
172	, = ¼	204	L = †	236	l = ∞
173	- = i	205	M = =	237	m = ∅
174	. = «	206	N = †	238	n = €
175	/ = »	207	O = †	239	o = ∩
176	0 = ☐	208	P = †	240	p = ≡
177	1 = ☐	209	Q = †	241	q = ±
178	2 = ☐	210	R = †	242	r = ≥
179	3 = †	211	S = †	243	s = ≤
180	4 = †	212	T = †	244	t = ∫
181	5 = †	213	U = †	245	u = ∫
182	6 = †	214	V = †	246	v = ÷
183	7 = †	215	W = †	247	w = ≈
184	8 = †	216	X = †	248	x = °
185	9 = †	217	Y = †	249	y = ·
186	: = †	218	Z = †	250	z = ·
187	; = †	219	Ä = ■	251	ä = √
188	< = †	220	Ö = ■	252	ö = n
189	= = †	221	Ü = ■	253	ü = z
190	> = †	222	^ = ■	254	ß = ■
191	? = †	223	_ = ■	255	DEL

New Word / WordStar: Einige Anmerkungen zu WordStar

Wenn wir diese Tabel anwenden beim WordStar, dan kommen ^PF und ^PG wieder frei und können für andere Zwecke angewent werden.

Die Patch-Programme sind auf CLUB.0??. Wenn Du zuviel Probleme hast um deine WordStar zu patchen, schicke mir dann eine Diskette mit deiner WS-version und mit Aufgabe welche Drucker du benützt und welche in NWPRINT.OVR aufgenommen werden müssen. Abrechnung der Kosten (nur Port-Kosten) über Club-Konto. Für das 2732-EPROM (mit Grafik und Kursivschrift) siehe Angebotsliste Hardware. Upgrade von 2732-EPROM (Grafik) mit Kursivschrift gegen Portkosten.

Viel Spaß

Erik D'Hondt - Wilgstraat 25A - B 9320 EREMBODEGEM - Belgien

Einige Anmerkungen zu WordStar

(Herbert zur Nedden, 2000)

Auf den vorherigen Seiten wurden ja einige Features von NewWord und WordStar genannt, die diesen beiden Super-Programmen beigebracht wurden. An dieser Stelle möchte ich erläutern, was davon mit meinen Patches (die habe ich zusammengetragen und -gebastelt; die Ideen stammten zum Teil von anderen!) für WordStar auf der CLUB.055 erschlagen werden kann.

1. MTX-Grafikzeichen einzeln in ^PÄ und ^PÖ eingerahmt, also insbesondere die Rahmen so, wie mit unserer ersten Idee für die Einbindung von Grafik-Sonderzeichen von Michael Keßler. Das diese auch weiterhin unterstützt werden müssen, dürfte wohl klar sein - was soll ich sonst mit meinen alten Texten tun? Damit sie gedruckt werden, muß der Druckertreiber entsprechend angepaßt werden.
2. IBM-Grafikzeichen in ^PX ^PX eingerahmt, also Erik d'Hondts Version der Grafikeinbindung (wohlgemerkt in ^PXer und nicht ^PYer!). Hier muß WordStar gepatcht werden und der Drucker den IBM-Grafikzeichensatz drucken können.
3. Kursivschrift auf dem Bildschirm (auch dank Erik d'Hondt). Das ist ein kleiner Patch für WordStar.

Was muß nun für die o.g. drei Feinheiten getan werden?

- a) WS-Patches von CLUB.055 einspielen.
- b) Für 2) und 3) muß natürlich das Block-Grafik-EPROM der 80Zeichen-Karte ausgetauscht werden, damit die IBM-Grafiksonderzeichen und die Kursivschrift auf dem Bildschirm sichtbar werden können.
- c) Wenn Du keinen EPSON LQ-850 oder dazu kompatiblen, wie den EPSON LQ-500 Dein Eigen nennen kannst, mußst Du vermutlich den Druckertreiber selbst anpassen, um in den Genuß des richtigen Ausdrucks von 1) und 2) zu kommen. Auf CLUB.055 findest Du dazu allerdings die eine oder andere Hilfestellung.

S o f t w a r e: KLIX-Moni

K L I X - M o n i Version 1.0

(Herbert zur Nedden, 2071)

KLIX-Moni ist ein ... ääh **der** Debugger für Memotech-Computer, so sie unter **RAM 6.1** (oder besser) laufen. **KLIX-Moni** ist ein **KLIX-Overlay**, und gerade dadrin liegt seine Stärke! Ist **KLIX-Moni** geladen, kannst Du ein laufendes TPA-Programm einfach mit **<SHIFT-ESC>** unterbrechen, Dir die aktuellen Register-Inhalte holen und das Programm im Debugger weiterlaufen lassen.

Läuft das Programm, welches zu untersuchen ist selbst im **KLIX-Heap**, ist das Unterbrechen mittels **<SHIFT-ESC>** natürlich nicht möglich. Außerdem mußt Du beachten, daß **KLIX-Moni** den Pufferbereich an Adresse **0F980h** im Common selbst benutzt - und einige **KLIX-Overlays** tun das auch. Willst Du ein **KLIX-Overlay** austesten, solltest Du es während der Entwicklungs- und Entwanzungs-Phase nicht diesen Puffer verwenden; Du kannst z.B. den Puffer zum Testen nach **0C000h** legen - er überschreibt dort zwar einen Teil der TPA, aber das macht i.a. nichts. Anderenfalls passiert evtl. magisches.

An den Stellen, an denen **KLIX-Moni** zum Zuge kommen soll, mußt Du per **BnkCal** in die TPA springen und dort z.B. folgendes Programm aufrufen

```
CALL 0F016h ; Lies Zeichen von Tastatur
DS   20,0   ; 20 x NOP
RET
```

damit **SHIFT-ESC** überhaupt zum Zuge kommen kann - oder - was eigentlich einfacher ist: Du läßt dein **KLIX-Overlay** gleich unter **KLIX-Monis** Trace-Modus laufen.

Die Installation geht einfach mit **TheInst KM.KLX** unter Zuhilfenahme der Datei **KM.INS**. Dabei bitte **genau** lesen, was auf dem Bildschirm geschrieben steht! Wenn es darum geht, Bits aus einem oder dem anderen Byte zu installieren, darfst Du **AUF KEINEN FALL** die nicht erläuterten Bits ändern!

Da wir der Meinung sind, daß **KLIX-Moni** nichts für 'unbedarfte' ist, da sich damit leicht einiges kaputtmachen läßt, wenn man nicht weiß was man tut und will, haben wir uns viel Arbeit gespart, und die Beschreibung der Kommandos von **KLIX-Moni** kurz gehalten.

Die Steuerung von **KLIX-Moni** erfolgt über Olafs OK-Menü. Das Hauptmenü ist als unterlegter zweizeiliger Balken oben auf der Glotze immer zu sehen, die Untermenüs werden unter dem jeweiligen Auswahlpunkt angezeigt. Du kannst die einzelnen Menüpunkte außer ENDE über den groß angezeigten Buchstaben aus den Texten direkt anwählen. Zusätzlich kannst Du natürlich mit den Cursortasten den hervorgehobenen Menüpunkt bewegen und diesen mittels **<HOME>** anwählen. Um ein Menü ohne Ausführen einer Auswahl abzubrechen dient im Hauptmenü **<ESC>** und in den anderen **<BRK>**.

Beachte bitte übrigens, daß **KLIX-Moni** in der Lage ist, im Assembler-Listing nicht nur vorwärts, sondern auch rückwärts zu scrollen! Auch einige Befehle, die Du im Trace-Modus hast laufen lassen, kann **KLIX-Moni** rückwärts abspulen (allerdings werden dabei nur die Registerwertsänderungen rückgängig gemacht).

Übrigens: **KLIX-Moni** schaltet automatisch auf Funktionstastentabelle 5 um. In dieser Tabelle solltest Du die Pfeiltasten, HOME, BRK und CLS nicht belegen. Der Grund für die Verwendung einer Funktionstastentabelle ist, daß Du Dir so z.B. TE auf F1, TU auf F2 usw. legen kannst, um bequemer zu debuggen.

T u r b o - P a s c a l : Directoryzugriff

Directory-Zugriff

(Herbert zur Nedden, 2000)

Was ist der Unterschied zwischen folgender Routine:

```

1  {-----}
2  {           Art 1, ein direktory einzulesen           }
3  {           (aus dir.inc von club.011)                }
4  {-----}
5
6  Procedure Dir;
7
8
9  Var FCB      : Array[1..32] of Char;
10     Buffer   : Array[1..128] of Char;
11     Offset  : Integer;
12     Loop    : Integer;
13     DatNam  : String[12];
14     NDat    : Integer;
15     Ant     : Char;
16
17 Begin
18     write('Laufwerk (A..P) ? ');
19     read (kbd,Ant);
20     Ant:=upcase (Ant); write (Ant);
21     writeln;
22
23     BDOS(26,Addr(Buffer));
24     FCB[1]:=Char (Ord(Ant)-Ord ('A')+1);
25     For Loop:=2 to 12 do FCB[Loop]:='?';
26     For Loop:=13 to 32 do FCB[Loop]:=Char($00);
27     Offset:=BDOS(17,Addr(FCB));
28
29
30     If Offset <> 255 then
31         Begin
32             NDat:=0;
33             Repeat
34                 DatNam:=Copy(Buffer,Offset*32+2,11);
35                 Insert ('.',DatNam,9);
36                 write (DatNam,' : ');
37                 Ndat:=Ndat+1;
38                 If NDat = 5 then
39                     Begin
40                         NDat:=0;
41                         writeln
42                     end;
43                 Offset:=BDOS(18);
44                 until offset = 255;
45             end
46         else
47             writeln ('Keine Dateien gefunden !');
48     end; {Dir}
49
50     {-----}
51
52 begin {MAIN}
53     dir
54 end.

```

T u r b o - P a s c a l : Directoryzugriff

Und dem folgenden Programm, fragte Peter Würfel

```

1  {-----}
2  {           Art 2, ein direktory einzulesen           }
3  {           (aus irgendeinem Programm von HzN)         }
4  {-----}
5
6  Procedure Dir;
7
8
9  Var FCB      : String[13] absolute $005b;
10     Buffer    : String [128];
11     Offset   : Integer;
12
13     DatNam   : String[12];
14     Ndat     : Integer;
15     Ant      : Char;
16
17 Begin
18     write('Laufwerk (A..P) ? ');
19     read (kbd,Ant);
20     Ant:=upcase (Ant); write (Ant);
21     writeln;
22
23     BDOS (26,Succ(Addr(Buffer)));
24     FCB := concat(Char(Ord(Ant)-Ord ('A')+1),'?????????',#0);
25
26
27     Offset := BDOS(17,Succ(Addr(FCB)));
28     Mem[Addr(Buffer)] :=128;
29
30     If Offset <> 255 then
31         Begin
32             Ndat:=0;
33             Repeat
34                 DatNam:= Concat(Copy(Buffer,Succ(Succ(Offset shl 5)),8),
35                                 '.',Copy(Buffer,10+Offset shl 5,3));
36                 write (DatNam,' : ');
37                 Ndat:=Ndat+1;
38                 If Ndat = 5 then
39                     Begin
40                         Ndat:=0;
41                         writeln
42                     End;
43                 Offset := BDOS (18);
44                 until offset = 255;
45             End
46         else
47             writeln ('Keine Datein gefunden !');
48     End; {Dir}
49
50 {-----}
51
52 begin {MAIN}
53     dir
54 end.
```

S o f t w a r e: DIREX

Hier nun die Unterschiede, wobei ich mich dabei auf die Zeilennummern der Vorseiten beziehe - dazu habe ich sie schließlich eben deshalb im Info.

Zeile	1. Programm	2. Programm
9	FCB = Array im Pgm	FCB = String an Adresse 5Ch
10	Puffer: 128-Byte Array	Puffer: 128-Byte-String
	Hinweis: Ein String hat als erstes das Längenbyte, und dahinter seinen eigent-lichen Inhalt. Daher belegt ein 128-Byte-String 129 Bytes im Speicher.	
23	DMA-Adresse auf Buffer legen	Hier muß das mittels succ erfolgen, da wie im o.g. Hinweis gesagt, der Datenbereich des Strings hinter dessen ersten Byte, dem Längenbyte beginnt.
24	32 Bytes des FCB gefüllt (gemütlich per FOR-Schleife)	Die ersten 14 Bytes des FCB gefüllt (Das reicht auch fürs Dateisuchen)
27	Wie 23	Wie 23
28		Längenfeld des Buffer (ist ein String) auf 128, damit auch die 128 Bytes ansprechbar sind.
34	Offset*32 = Offset shl 5,	succ(x) = x +1
34-35	Namen komplett holen und Punkt dann einfügen	Namen aus seinen Teilen zusammen-setzen und dabei Punkt dazwischentun.

Fazit: Beide Programme leisten das selbe - sind nur etwas unterschiedlich programmiert, wobei beide ihre Vor- und Nachteile haben!

D I R E X . I N C

(Peter Würfel,7262)

Ich gehöre zwar noch nicht zu den Computerbenutzern, die zum ikonenklickenden Nagetierschieber verkommen sind, aber ich finde es immer wieder störend, wenn ich von einem Programm dazu aufgefordert werde, einen Dateinamen einzugeben. Die meisten dieser 8+Punkt+3-Wortkürzel halten sich nur in meinem Kürzestgedächtnis auf und stehen in solchen Momenten nicht zur Verfügung. Es bleibt mir dann nur, mit Dijey nochmal nachzuschauen, wie diese Datei nun eigentlich heißt. Schöner wäre es doch, wenn das Programm, immer dann, wenn es nen Dateinamen wissen will, mir ein Directory anbietet und ich dann den gewünschten Dateinamen nur noch 'anklicken' brauche. Und nachdem ich in Pascal meine ersten Gehversuche unternehme, hab ich mich hingesezt, mit dem Ziel eine möglichst universell zu verwendende Procedure zu schreiben, die genau dies leistet: PROCEDURE DirEx.

Diese Procedure findet Ihr auf Club.058 im Include-file 'direx.inc'. Wie die Procedure funktioniert, kann man sich in 'demodir' bzw. 'krypto2' anschauen, wenn ... und nun die Einschränkung... man den Bildschirmtreiber von Ram 6 hat.

Wenn ich hier auf diese Procedure hinweise, dann nicht deshalb, weil ich sie für besonders toll halte, sondern weil ich auf Nachahmer hoffe, die möglichst universell einsetzbare Routinen für die Club-PD zur Verfügung stellen. Da schlummert doch in verschiedenen Diskettenboxen sicher einiges an Procedures, die auch von anderen beim Programmieren sinnvoll eingesetzt werden können. Außerdem hoffe ich als Pascal-Novize auf Kritik bzw. Verbesserungsvorschläge. Einen Nachteil hat DirEx mit Sicherheit: der Bildschirmhintergrund an der Stelle, an der das Directory-Fenster geöffnet wird, wird nicht gerettet. Diesen Nachteil habe ich durch die Procedure WindowSave (wsave.inc) auszubügeln versucht. Doch aus zwei Gründen habe ich diese Procedure nicht in DirEx eingebaut: 1. Sie könnte schneller sein und 2. sie braucht mir zu viel Speicherplatz. Mit dem ersten Mangel könnte ich leben; wie man den zweiten Mangel beheben könnte, hab ich so ne Idee: Zeiger? Doch wie, das ist mir nicht klar. Wer hilft?

H a r d w a r e: Centronics-Schnittstelle**Ansteuerung der Centronics-Schnittstelle**

(Herbert zur Nedden, 2071)

Gelegentlich kann man ja mal in die Verlegenheit kommen, etwas über die Centronics-Schnittstelle übertragen zu wollen ... und es klappt nicht. Thomas Mäurer wollte z.B. auf diesem Weg Daten vom Memotech auf einen Atari ST übertragen. Seine Versuche unter BASIC mittels LPRINT bescherten ihm lediglich einen hängenden Rechner.

Die Centronics-Schnittstelle u.a. folgende Signale:

DATA 1-8	Die 8 Datenbits
STROBE	Dieses Signal muß vom Sender kurz auf Low gelegt werden, damit der Empfänger die Daten übernimmt.
BUSY	Hier signalisiert der Empfänger mit High, daß er nicht bereit ist, Daten entgegenzunehmen
PE	Hier signalisiert der Empfänger mit High, daß das Papier alle ist.
SLCT	Hier signalisiert der Empfänger mit High, daß er da ist.
ERROR	Hier signalisiert der Empfänger mit Low einen Fehler.
ACK	Hier bestätigt der Empfänger den Eingang von Daten durch einen Low-Impuls.

Die Centronics-Schnittstelle des MTX hat die o.g. Signale bis auf ACK. Da Memotech sehr gründlich war, müssen BUSY, PE, SLCT und ERROR alle den richtigen Pegel aufweisen, damit die Daten via STROBE an den Empfänger gereicht werden.

Dummerweise liefern nicht alle Centronics-Empfänger die o.g. Signale - 0-8-15-Drucker kennen nur BUSY und legen die anderen der o.g. Pins einfach fest auf den richtigen Pegel. Es kann gut sein, daß die Centronics-Schnittstellen von IBM-PCs und Atari-STs, die sowohl Daten senden als auch empfangen können genialerweise im Empfangs-Modus die o.g. Signale nicht alle korrekt erzeugen (wundern tät es mich nicht!). Im Prinzip ist es ja auch berechtigt, wenn diese z.B. PE auf High legen, da schließlich diese Geräte kein Papier drin haben...

Memotechs Centronics-Druckerausgabe des MTX läuft wie folgt ab:
(Siehe auch die Beschreibung von Port 04h im MTX-Handbuch fast ganz hinten.)

```

CENTR:  IN    A,(4)      ; Lies Statusport
        AND    00001111b ; nur die unteren 4 Bit sind relevant
        CP     00001010b ; SLCT=1, PE=0, ERROR=1, BUSY=0 ?
        JR     NZ,CENTR  ; Nein? Dann warten wir halt noch etwas
        LD     A,(Daten) ; Daten zu mir
        OUT    (4),A     ; Daten auf den Port
        IN     A,(0)     ; STROBE=0
        IN     A,(4)     ; STROBE=1

```

Um nun mit 'faulen' Centronics-Empfängern, die nur BUSY liefern zusammenarbeiten zu können, ändere man die 2. und 3. Zeile in die folgenden:

```

        AND    00000001b ; nur BUSY interessiert mich
        CP     00000000b ; BUSY=0 ?

```

(Ja, der CP kann entfallen, da der AND das Zero-Flag schon richtig gesetzt hat!)

Also flugs eine eigene Ausgabe-Routine in Assembler gestrickt oder in FDX-BASIC (oder womit Du auch gerade ausgeben willst) die o.g. Routine suchen, also insbesondere das AND 0Fh gefolgt vom CP 0Ah und diese beiden Befehle etwas vereinfachen, z.B. in AND 01h und CP 00h. Dann könnte es funktionieren.

Übrigens: RAM 4.x/6.x prüft nur das BUSY-Signal und nicht PE, SLCT oder ERROR.

H a r d w a r e: IBM-Tastatur**IBM-Tastatur von ESCOM am MTX**

(Herbert zur Nedden, 2071)

Ich wurde von Martin Reuter gebeten, zu versuchen, eine IBM-kompatible Tastatur von ESCOM am MTX unter RAM 6.x ans Laufen zu bekommen. Das schöne an der Tastatur ist zum einen, daß sie schwarz wie die Nacht ... äh wie unsere Kiste ist und, zum anderen, daß sie einen eingebauten Taschenrechner hat. Außerdem ist es für mich auch eine Art Herausforderung gewesen, das Teil ans Laufen zu bekommen!

Klar, im Info stand kürzlich, wie es gemacht wird: Man nehme ein paar ICs und bastle ein Interface, welches dann über einen PIO angeschlossen wird. Bevor ich in die Vollen gehen wollte, habe ich mir erst mal die Artikel aus der c't geholt, um mir nochmal durchzulesen, was diese Teile an Signalen wie liefern. Hier kurz die Quintessenz:

PC-Modus: 1 Startbit gefolgt von 8 Datenbits

AT-Modus: 1 Startbit, 8 Datenbits, 1 Paritybit ungerade, 1 Stopbit

Dabei unterscheiden sich die von der Tastatur gelieferten Daten, die sog. Scancodes in diesen beiden Modi. Obendrein ist es so, daß es im AT-Modus einige Tastaturen für Tasten mit der gleichen Bezeichnung (z.B. rechte und linke SHIFT-Taste) an sich den selben Scancode liefern, jedoch bei einer der beiden Tasten mit vorangestelltem E0h.

Im c't-Artikel, in dem das Interface für AT-Tastaturen beschrieben wird, wurde über das Datenformat, insbesondere über das Paritybit gemosert - damit hätte der ACIA (oder wie das Teil heißt) des Atari Probleme. Wenn Du jedoch mal das Datenformat genau mit dem einer seriellen Schnittstelle vergleichst, darfst Du Dich freuen! Es ist eben dieses.

Damit war für mich beschlossen, daß ich die Tastatur direkt an die serielle Schnittstelle anschließen wollte. Mittels eines Oszilloskops habe ich herausgefunden, wie schnell die Tastatur ihre Daten auf die Reise schickte und erst mal versucht, diese Baudrate auch zu erwischen. Meine ersten Versuche, Daten zu empfangen, gingen natürlich in die Hose! Das lag daran, daß die Tastatur zwar die Daten im seriellen Format sendet, aber mit TTL-Pegel (0-5 Volt), während die RS232 (auch V24 genannt) mit ± 12 Volt arbeitet. Nachdem ich die V24-Konverter herausgenommen hatte, lief es. Die Daten kamen an! Dazu mußte ich nur die Daten der Tastatur direkt auf RXD des DART zu legen - also alles ohne Interface.

Um die beiden Frequenzen untersuchen zu können, habe ich folgendes getan:

1. Tastatur mit Masse und 5 Volt versorgt und die Leertaste mit einem Gewicht gedrückt gehalten. Dann die Datenleitung auf den Oskar gepackt.
2. Programm geschrieben, welches dauerhaft ein Byte seriell ausgibt; zwischen dem Senden zweier Bytes eine Verzögerung eingebaut, damit ich die einzelnen Bytes gut erkennen kann - auf dem 2. Kanal des Oskar liegt RXD.
3. Mit RSINIT an der Baudrate gedreht.

Die Baudrate wird vom CTC geliefert, und dieser erhält 4 MHz geteilt durch 13 an seinem Zählereingang. Mit diesen 4/13 wurde ich übrigens nicht glücklich, so daß ich dem CTC an seinem Zählereingang mit 4 MHz versorge.

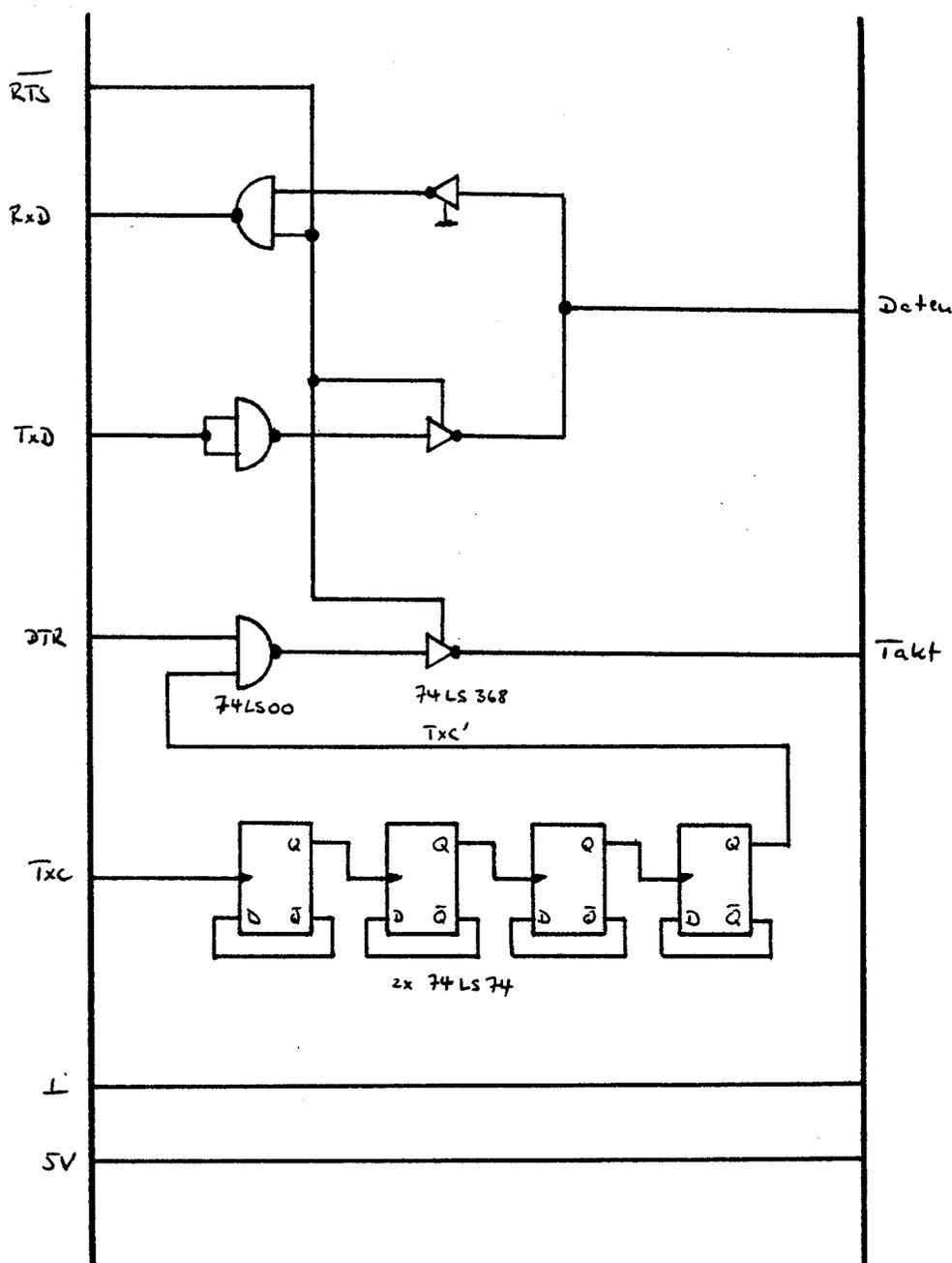
Leider lieferten die linke und rechte SHIFT-Taste genau die selben Scancodes und obendrein beim Loslassen der Taste F0h gefolgt von zweimal dem Scancode. O.K.: , damit kann ich zur Not noch leben, aber die Pfeiltasten des 10-Blocks und des Cursor-Blocks waren nicht zu unterscheiden - und das wollte ich nicht. Folglich griff ich nochmals zur c't, und fand heraus, daß der Rechner der Tastatur auch etwas senden kann. (Genaugenommen hatte ich das schon früher registriert und auch deswegen mit der seriellen Übertragung geliebäugelt.) Kurz vorab: seit ich nun der Tastatur den Reset-Befehl übermitteln kann, tut sie so, wie ich will!

Hardware: IBM-Tastatur

Wie senden? Der Rechner muß die Taktleitung kurz auf Low ziehen und danach die Daten und den Takt senden. Jedenfalls steht das so in der c't. Daher war eine Schaltung erforderlich, die folgendes konnte: Wahlweise die Daten-Leitung an RxD oder TxD an die Daten legen; die Takt-Leitung wahlweise in Ruhe lassen, auf Low legen oder mit dem Sendetakt versorgen. Dazu brauchte ich zwei Signale: eines mit dem die Übertragungsrichtung gesteuert wird und eines, um den Takt auf Low zu ziehen. Erfreulicherweise bietet der DART zwei programmierbare Pins /RTS (lies als RTS-Quer) und DTR. /RTS ist sogar besonders schön: Ich kann es jederzeit auf Low legen; lasse ich es wieder auf High gehen, so passiert das erst, wenn der Sendepuffer des DART leer ist. Damit war klar, daß /RTS die Richtungssteuerung erledigen muß.

Beim Takt mußte ich leider auch etwas tun. Das Signal an TxC ist der Sendetakt mal 16, da der DART durch 16 teilt. Dieses Signal muß ich daher noch durch 16 teilen, was mit vier FlipFlops kein Problem darstellt. Bei dieser Aktion wird das Signal auch in das erforderliche Rechteck-Format umgesetzt.

Hier die Schaltung:



Hardware: IBM-Tastatur

Ist /RTS = High, so werden die Tastatur-Daten an RxD geleitet (ein NAND-Gatter, bei dem ein Eingang fest auf High liegt, wirkt wie ein Inverter; liegt ein Eingang fest auf Low liefert das NAND immer High). Die beiden Inverter hinter TxD und in der Takt-Leitung sperren, so daß nichts vom Rechner raus geht.

Wird /RTS auf Low gelegt, kehrt sich diese Situation um: Bei RxD kommt nichts mehr an; dafür gelangt TxD an Daten und das 'Kombinat' aus DTR und TxC/16 geht an den Takt. Ist nun DTR Low, liegt Takt auch auf Low (das liegt wieder am NAND-Gatter); ist DTR High liefert TxC/16 am Takt an.

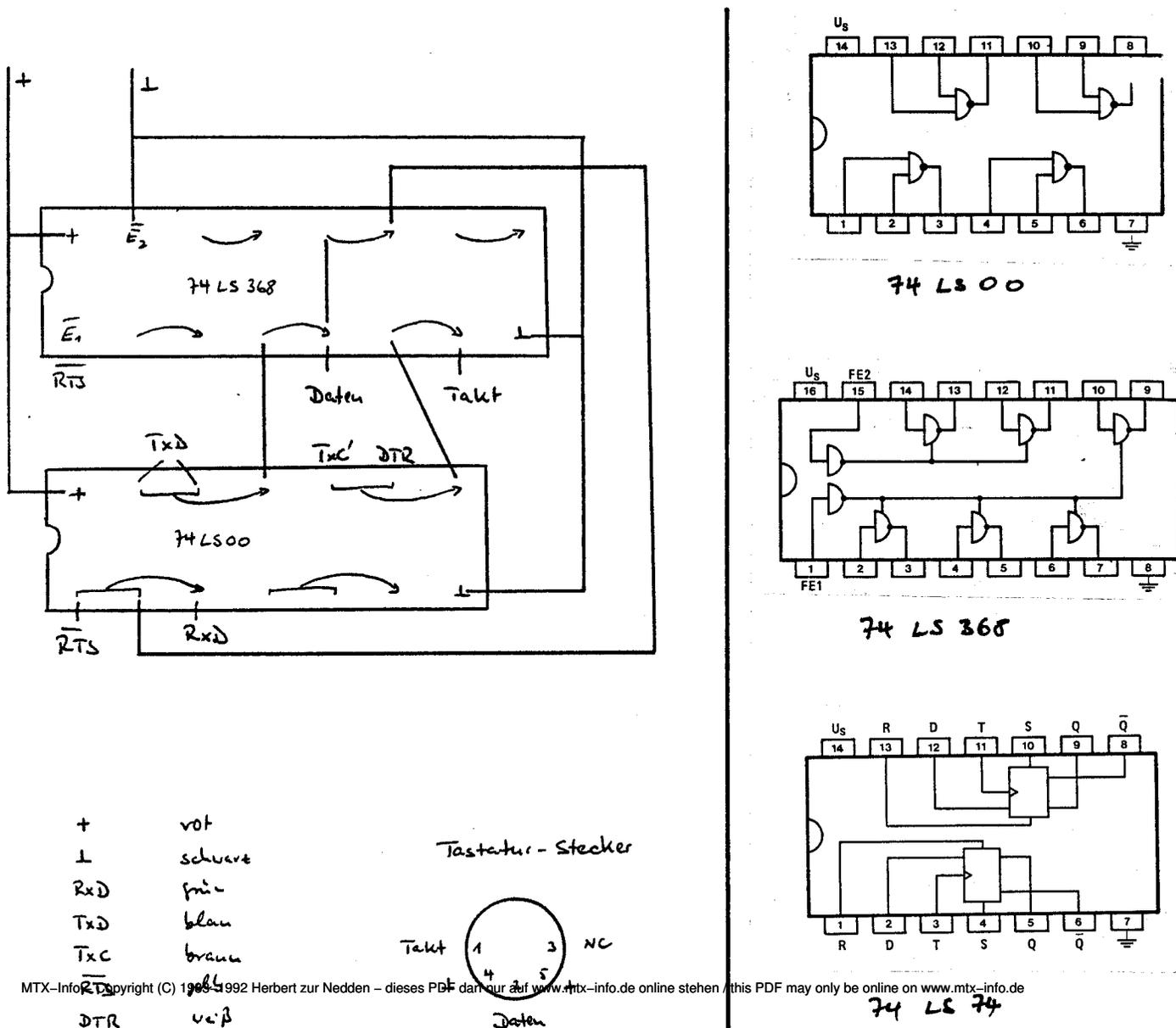
Für mein Interface ergibt sich mit /RTS und DTR folgende Programmierung:

Beim Empfangen sind /RTS und DTR auf High und Senden gesperrt.

Um zu Senden passiert nun:

1. DTR und /RTS auf Low
2. Daten an den DART
3. DTR auf High und Senden freigeben
4. /RTS auf High
5. Nach Erfolgsmeldung von der Tastatur Senden sperren.

Damit es mit dem Löten einfacher ist, hier ein 'Blick von oben' auf meine Schaltung; den Takteiler aus 2x 74LS74 habe ich nicht mit dabei - den kriegst Du sicherlich auch so hin.



H a r d w a r e: IBM-Tastatur

Kommen wir zum Treiber, also dem Programm, welches die Arbeit tut.

Da ich eh schon ein Programm habe, welches serielle Daten per Interrupt einliest und daraufhin etwas in den Tastatortreiber stellt, nämlich den Maustreiber, habe ich diesen erweitert: er liest nun Maus und Tastatur ein.

Dieser neue Treiber heißt **KbdMou 1.0** und belegt zwei serielle Schnittstellen - beide müssen den selben DART verwenden; nur Maus oder nur Tastatur wird nicht unterstützt - wenn Du willst, kannst Du Dir gerne Teile aus dem Treiber heraus holen oder löschen.

So wird der DART für die Tastatur initialisiert:

```

DartReg1Keybd:  db      1
                db      00010100b
                ;          ^----  ext. int. disabled
                ;          ^----  Tx int. disabled
                ;          ^----- status affects vector (nur B!)
                ;          ^^----- int. on all Rx chars (p. affects vector)
                ;          ^^----- wait/ready disabled
DartReg3Keybd:  db      3
                db      11000001b
                ;          ^----  Rx enabled
                ;          ^----- no auto enables
                ;          ^^----- 8 bits per received character
DartReg4Keybd:  db      4
                db      01000101b
                ;          ^^----  odd parity
                ;          ^^----- 1 stop bit
                ;          ^^----- x16 clock mode
DartReg5Keybd:  db      5
                db      01100000B
                ;          ^----  RTS
                ;          ^----- Tx disabled
                ;          ^----- nix send break
                ;          ^^----- 8 bits per transmitted character
                ;          ^----- DTR

```

Wichtig für den normalen Lese-Betrieb ist im Vergleich zur Maus, daß Auto-Enables aus ist, DTR und RTS auf 0 gesetzt werden (RTS auf Low bedeutet /RTS auf High) sowie daß das Senden gesperrt ist (Tx disable).

Nach dem Laden versucht der Treiber KbdMou erst mal, die Tastatur zu initialisieren; und das versucht er ... und versucht ... und versucht ...! Das funktioniert natürlich erst mal nur bei der ESCOM-Tastatur FK3002, wenn beide DIP-Schalter auf OFF stehen. Falls das nicht so recht klappt, kann es durchaus helfen, die Tastatur aus- und wieder einzustöpseln.

Bei KbdMou ist auch ein Pascal-Testprogramm ESCOM dabei, mit dem Du erst mal schauen kannst, ob die Tastatur überhaupt sendet und empfängt. Wenn Du bei dem Programm z.B. ein Reset an die Tastatur zu senden versuchst, werden auch weiterhin die eingelesenen Informationen angezeigt - und davon liefert eine nicht initialisierte Tastatur reichlich. Dieses Programm ermöglicht es auch, die Parameter der seriellen Schnittstelle mal etwas zu verändern, um evtl. auch andere Keyboards zum Arbeiten an unserer Kiste zu überreden.

Tja, daß die Tastatur tut, beweist dieser Artikel - er ist nämlich damit verbrochen. Bleibt eigentlich nur noch die Frage, wie ich die Tastatur belegt habe.

H a r d w a r e: IBM-Tastatur

Die neue Tastatur hat einige Tasten mehr. Daher stellte sich die Frage: "Wohin mit den neuen Tasten? Welche Tastencodes vergeben?".

Unter RAM 6.x sind folgende Tastencodes belegt:

```

00h-7Fh: normale ASCII-Codes
8xh:     F, F'
9xh:     10er
Axh:     10er', 0ADh = SHIFT-ALPHALOCK
Bxh:
Cxh:     ^F, ^F'
Dxh:     ^10er
Exh:     ^10er'
Fhx:     sonder

```

Die Tasten der IBM-Tastatur liefern folgendes:

- Die handelsüblichen wie die Buchstaben usw. des Hauptteils liefern das, was draufsteht. Etwas gewöhnungsbedürftig ist sicherlich, daß nun das ' auf der unteren ungeshifteten Ebene liegt.
Da es Verschwendung ist, das ' auf zwei Tasten zu haben, nämlich auch noch auf SHIFT-#, habe ich SHIFT-# mit / belegt.
SHIFT-0 ist mit = belegt, da es auf der Taste steht. Dafür befindet sich auf SHIFT-^ ein kleiner Kringlel, der nun das Datum liefert.
- F9-F12 sind nicht dem 10er-Block angegliedert worden sondern auf B0h-B3h gelandet. Mit SHIFT +04h, mit CTRL +40h.
- Der 10er-Block war schon problematischer: Beim MTX sind die Pfeile auf den Ziffern 5, 1, 3 und auf dem .; die IBM-Tastatur hat die Pfeile auf 8, 4, 6 und 2. Daher stellte sich die Frage, wie der 10er-Block belegt werden soll. Entweder so, daß die Ziffern passen oder daß alles etwas verschoben wird, damit die Pfeile passen.
Ich habe den 10er-Block so integriert, daß die Pfeiltaste weiterleben! Daher stimmen die Zahlentasten nicht mehr! Ein kurzer Versuch ergab, daß es mehr stört, wenn die Pfeile nicht passen, als bei den Ziffern. Mit <NUMLOCK> kannst Du die Tasten des 10er-Blocks fest auf die Ziffernbelegung einfrieren. Die /, *, - und + des 10er-Blocks liefern eben das, was draufsteht.
- Die Tasten des Cursor-Blocks habe ich fest belegt - Funktionstasten gibt's wirklich genug. Diese Tasten sind immun gegen SHIFT, weil ich das schon immer mal für die Pfeile haben wollte! Allerdings hat CTRL eine Wirkung:

<u>Taste</u>	<u>ohne/mit CTRL</u>
Einf	^V / ^U
Entf	DEL / BS
Pos1	^QS / ^QDEL
Ende	^QD / ^QY
BildU	^R / ^QR
BildD	^C / ^QC
PfeilU	^E / ^Z
PfeilD	^X / ^W
PfeilR	^D / ^F
PfeilL	^S / ^A
- Bleiben noch ein paar Sondertasten:
Pause liefert das selbe, wie SHIFT-Esc
Rollen liefert FCh, Druck FBh und SHIFT-Tab FAh, sind also Funktionstasten.

Der Fortschritt

(ein Kurzbericht von und mit D.Richter)
für die Sparte "Moderne Aufklärung für MSX-User"

Endlich war es soweit. Mein Freund Otto (der Name wurde von mir geändert) hat sich einen Computer des Fortschritts gekauft. Der gute, alte MSX - "...ich werde meinen 8235 bis ans Ende meines Lebens behalten" - mußte gehen. Der neue Computer mit einem noch kürzeren Kürzel PC machte sich auf dem Schreibtisch breit. Tja, umso kürzer das Kürzel, desto mehr steckt dahinter. Betrachten wir doch einmal Otto's Schreibtisch etwas genauer: Anfangs kam mir der Computer sehr klein vor - "Ach so, das ist nur die Tastatur!". Außerdem fand ich gar kein Laufwerk. Ah, da ist es schon, wie konnte ich diesen "Panzerschrank" nur übersehen? (Die Technik wird ja mit zunehmendem Fortschritt immer kleiner; das hatte ich ja ganz vergessen...)

Mein Freund Otto erklärte mir, daß in diesem Schrank, der jedes Wettrennen aufgrund seiner vorzüglichen Aerodynamik gewinnen würde, nicht nur ein, nein, zwei (2 !) Laufwerke enthalten sind; außerdem noch eine Festplatte, ein recht exotischer Begriff im Wortschatz eines MSX-Users. Das ist eine sog. Harddisk - "Warum denn dieses, die 3.5" Disks waren doch stabil genug..." -, die eine immens große Speicherkapazität besitzt. Man kann also alle wichtigen Programme auf diese kopieren; das ist eine erleichternde Sache, allerdings nicht nur für den Besitzer des Terminals, sondern auch für den Virenprogrammierer, der das gesamte Ding, samt allen wichtigen Programmen ins Jenseits befördert.

Das zweite "normale" Laufwerk ist gar kein normales Laufwerk. Der Schlitz ist nämlich viel länger und schmaler. Ach ja! Ich erinnere mich: Neunzehnhundert... längst vergessen (!?) Klomeldore C64. Fünf'n'viertelzoll, erklärt mir mein Freund Otto, ist heutzutage immer noch ein mehr oder weniger bekanntes Diskettenformat. (Wie man bereits jetzt schon sieht, ist der Fortschritt unaufhaltsam in Sachen PC...)

Der PC ist übrigens ein sog. Industriestandard-Rechner, d.h. verschiedene Firmen bauen dieses Hardwaremonster, und sie sind trotzdem immer kompatibel. (Das kennt man doch als MSX-User irgendwo her...)

Allerdings ist das Ganze bei PCs nur "fast immer" kompatibel..., aber dazu später mehr.

Werfen wir mal einen Blick auf den Schreibtisch meines Freundes Otto: Für die supergenialen Grafiken reicht natürlich kein normaler Fernseher, wie zu alten MSX Zeiten, aus. Ein spezieller Supermonitor ohne oder mit Biepstonausgabe - Klasse Sound !! - muß her.

Die meisten Programme für PCs sind mausgesteuert, damit sich der schreibfaule Chef, in seinem Ledersessel zurücklehnd und Zigarre paffend, nicht so "hektisch" bewegen muß. Selbstverständlich fehlt also auch die Maus nicht auf dem Schreibtisch meines Freundes Otto.

(Wenn man sich den gesamten Terminal mit allem, was dazugehört, anguckt, könnte man denken, es ist der Nachfolger des ENIAC (dieser brauchte allerdings ein ganzes Zimmer, der PC, fortschrittlich, wie er ist, nur ein halbes!))

Alle Theorie ist grau, also schalten wir das Ding mal ein: die extrem anwenderfreundliche Bedienung kam damit schon gleich beim Einschalten zutage... den Knopf habe ich natürlich nicht gefunden..., mein Freund Otto kam zu Hilfe.

Nach ellenlangem Initialisierungsvorspann (Konami macht das irgendwie interessanter), sehe ich meinen Cursor hinter einem Prompt blinken. (Nein, nicht MSX-DOS, sondern MS-DOS).

Erster Tippversuch: DIR "cr"... und es funktioniert..., aber was ist denn dieses?!!?... ich seh' ja gar nichts, so schnell geht das. Tja, 16 MHz, 16 Bit !!!

Als konkretes Beispiel: stellt euch vor, ihr fahrt in einem Auto auf der Autobahn und schaut aus einem Seitenfenster heraus...

Da wir gerade bei DIR, wie DIRECTORY (Inhaltsverzeichnis) sind: Um einem die Übersicht der ultravielen Files auf einer Diskette zu verschaffen, gibt es hier sog. Unterverzeichnisse. Im Durchschnitt gibt es davon so viele, daß ich hier diese Zahl nicht ausschreiben will (Das würde den Rahmen dieses Aufsatzes über diesen Superrechner PC sprengen...).

Zusammengefaßt kann man aber sagen, daß sich die Einrichtung tierisch gelohnt hat, da die Übersicht jetzt für jedermann garantiert ist !!!, für jedermann?...

Hier ein Auszug aus der Zeitschrift DOS aus der Reihe "PC-oder wie kann man es komplizierter machen ?"

"... Interessant sind die Inhalte der beiden Bits 3 und 2. Sie definieren, nach welchem Taktgenerator sich der Sequence-Controller der VGA-Karte richtet. Mit Bitkombination 00bin wählen Sie die Frequenz 25,175 MHz. Bei Verwendung der Bitkombination 01bin erfolgt die Ausgabe der Punkte dagegen mit 28,322 MHz. Für unsere Zwecke ist die höhere Frequenz notwendig. Das unterste Bit des Unterschiedliche-Ausgaben-Registers steuert die Basisadresse der Ports der VGA-Karte..." (Anm. von Otto: Dies gilt nur für die VGA-Karte, jedoch nicht für die MDA, Herkules, EGA, CGA, MCGA, TIGA, CGE, 8114/A, ... Der Fortschritt ist unaufhaltsam). Anmerkung von mir: Die Kompatibilität in Sachen Grafik ist also schon einmal gesichert!

Und wie einfach es ist, eine Picture-Datei auf dem Bildschirm auszugeben, zeigt folgendes Beispiel:

Um ein Pixel mit allem drum und dran auf dem Bildschirm auszugeben, braucht man beim PC etwa nur eine (!) Bildschirmseite Programm. Beim MSX sind es schon ganze drei Zeilen. (Programmiert in Assembler mit direkten Portausgaben an den Grafikchip).

Folglich brauchte mein Freund Otto, der nicht ganz unbeholfen in Sachen Computer und Programmieren ist, nur drei Tage "Studium", um eine Picture-Datei sichtbar zu machen. Er brauchte sage und schreibe nur(!) 6 1/2 Bildschirmseiten Programm! Dabei ist anzumerken, daß es meinem Freund Otto noch nicht gelungen ist, auch noch die entsprechenden Farben mitauszugeben... das wird vielleicht in den nächsten Wochen der Fall sein, da er jetzt einen Ausflug in die Uni nach Marburg geplant hat...(Anm. von Otto: Jetzt brauche ich nur noch Lader für die anderen Bildschirmformate zu schreiben, Macpaint, MacPaint (o.H.), IMG, GIF, TIF, EPS, WPG, MSP, BMP, CUT, PIC, SCX, TGA, IFF/LBM/CE, BAS, ART, CBM, BSG, RAW, RLE... zum Glück sind es nicht so viele, ich lebe ja noch ein paar Jahre, habe also noch genügend Zeit!).

Als Vergleich brauchte ich mit meinem niederen MSX-2+ (oder besser MSX-2?) mit nur 19268 Farben geschlagene 4 (soviell!) Zeilen (Puh, war das vielleicht eine Arbeit; da kann man mal wieder sehen, wie vorteilhaft und bedienungsfreundlich es ist, einen PC zu haben: man spart unendlich viel Zeit...).

Resümee von Otto: "Hilfe.....(Verzeih' mir, oh großer MSX)."

Resümee von mir: "High Tech in Perfektion..."