

Vorwort

Der AC1 benötigt wie viele andere Kleinstrechner eine parallele Tastatur, die direkt den ASCII-Code der betätigten Taste an einen PIO - Baustein anlegt. Der Aufbau einer solchen Tastatur ist meist recht kostspielig, zeitintensiv und sieht oft auch nicht besonders aus...

„Ordentliche“ und preiswerte industrielle PC-Tastaturen dagegen liefern keine seriellen Codes und haben noch einige andere Eigenarten (z. B. MAKE- und BREAK - Codes, d.h. es können sowohl beim Niederdrücken als auch beim Loslassen unterschiedliche Codes an den Rechner geliefert werden). Will man eine solche Tastatur anschließen, so muss man mehrere Aufgaben erledigen. Die Wesentlichen sind:

- Wandlung seriell nach parallel und
- Umcodierung.

Was liegt nun näher, als einen Mikrocontroller zur Lösung dieser Aufgaben zu verwenden? Die von mir vorgestellte Lösung mit einem PIC ist nur eine von vielen, aber einfach, flexibel, zuverlässig und preiswert zudem.

Wird das Interface extern als „Zusatzkästchen“ zwischen Tastatur und AC1 aufgebaut, so ist entweder eine zusätzliche Stromversorgung (5V) nötig oder man führt die Betriebsspannung an der Buchse der 8 Datenleitungen mit heraus. Am einfachsten ist es jedoch, eine Mini-Din-Buchse (6 polig) direkt in den AC1 einzubauen und das Interface ebenfalls im AC1 unterzubringen.

Bedienungshinweise

Die Tastatur erzeugt bis auf einige Ausnahmen den Code, der der jeweiligen Taste entspricht. Es müssen also keine Tasten neu beschriftet werden.

Nur einige wenige Tasten reagieren an der PS/2-Tastatur anders als beschriftet oder gar nicht:

Taste	Zeichenausgabe
`	>
~	<
~, , \, {, }, [,] @ ,€	keine Reaktion auf diese mit AltGr zu erreichenen Tasten
„WIN - links“, „WIN - rechts“, „Menü“	keine Reaktion auf die Windows-spezifischen Tasten
Bild ↑ und Bild ↓	reagieren wie Cursor hoch/runter

10er-Tastaturblock

Die Steuerfunktionen für Cursor und Bild auf dem 10er-Tastaturblock (bei NumLock an) sind nicht wirksam. Die Tasten haben **im Monitor** immer die gleiche Funktion (0...9, + - * / Enter).

Eine Ausnahme besteht unter **BASIC**: Ist Numlock ausgeschaltet, so werden anstelle dieser Zeichen die entsprechenden Grafikzeichen ausgegeben.

Beispiel: „*“ (Code 2AH) → „┘“ (Code: 2AH+80AH=AAH)

Steuerblock

Die Tasten des Steuerblocks liefern folgende Codes:

Druck 18H	Rollen *	Pause 13H	DRUCK schaltet die V24-Ausgabe an einen Drucker ein. Das Ausschalten erfolgt mit Strg+Y.
Eingf 05H	Pos1 01H	Bild ^ 0BH	
Entf 04H	Ende 03H	Bild v 0AH	„Ende“ = Eintasten-Ersatz für „Strg+C“

Spezialtasten

Die Taste AltGr dient nicht zum Umschalten auf eine andere Zeichenebene, sondern damit werden Steuerbefehle nur an die Tastatur gesendet

AltGr +Druck	F-Tasten programmieren	siehe dort
Strg +ESC	NMI - Taste	Mit drei weiteren Leitungen (open collector, low-aktiv,) zum Rechner lässt sich per PS/2-Tastatur die AC1-Hardware schalten. Die Eingriffsstellen für NMI, RESET und Grafikmodus sind an die jeweilige AC1-Hardware anzupassen. (Schaltplan zeigt eine Lösung für den „Standard-AC1“)
Alt +ESC	Tastatur+AC1 resetten	
Numlock (out)	Grafik-Modus ein (SCCH)	
Strg+Alt+Entf	RESET	„Affengriff“ bewirkt nur AC1-Hardware-Reset

Bei Einsatz eines NPN – Transistor (wie in der Schaltung angegeben) muss nach dem Einschalten die Num – Taste betätigt werden, damit der Grafikmodus ausgeschaltet ist, also muss immer die NUM-LED leuchten, wenn man im normalen Tastaturmodus ist.

Funktions-Tasten

Auch die Verwendung der 12 Funktionstasten ist möglich. Es gibt sogar vier Ebenen!

Vorbelegung:

Taste	Allein	SHIFT+	STRG+	ALT+
F1	# ? ↵ = Moni-Hilfe			
F2				
F3				
F4	(0C) ↵ = CLS			
F5	(16H) = CLOAD“““			
F6	(17H) = CSAVE			
F7				
F8				
F9	(12H) = STOP			DIR
F10	(15H) = RUN			Exit
F11	(14H) = LIST			LOAD „,“
F12	CONT ↵			CSAV „,“

Bislang unprogrammierte F-Tasten geben den Namen der Taste als Inhalt, z. B. „Strg+F6“ aus. Die F-Tasten lassen sich (einzeln) nach eigenen Wünschen programmieren. Auf den Tasten lassen sich bis zu 16 Zeichen lange Kommandofolgen unterbringen. Die Belegung wird dauerhaft im PIC gespeichert.

Ablauf der Programmierung

Beispiel: F-Taste Shift+F4 soll mit folgendem Kommando belegt werden:

- D 4000 401F (Anzeige 4000... 401F)
- ↵ (Enter = Kommando ausführen)

Schritt	Tasten	Bedeutung	LED-Reaktion
1.	AltGr + Druck	Umschalten in den Programmiermodus	Alle 3 LEDs leuchten
2.	Shift+F4	die zu programmierende F-Taste	Num-LED verlischt
3.	D 4000 401F	Folge der zu programmierenden Zeichen	CAPS-LED verlischt bei jedem Tastendruck kurz
4.	↵		
5.	F1	(oder andere beliebige andere F-Taste) = Programmierung abschließen	Alle 3 LEDs verlöschen

Besonderheiten:

- Auch im Programmiermodus werden die Zeichen (zur Kontrolle) an den Monitor geschickt und sind dort sichtbar. Bereits während der Programmierung ist das Ergebnis sichtbar!
- Werden als zu programmierendes Zeichen **Steuercodes** (01...1FH) verwendet, so werden deren Funktionen (sofern vorhanden) sofort nach dem Erkennen bei der Ausgabe an den Bildschirm ausgeführt, ohne dass erst <ENTER> folgen muss (im obigen Beispiel Strg+L=0CH=Bildschirm löschen).
- Wird als zu programmierendes Zeichen die Taste „**Pause**“ gedrückt, so wird in der Ausführung der Kommandobytes eben eine Pause (ca. ??? ms) eingeschoben, ehe das nächste angegebene Kommandobyte ausgeführt wird. **ACHTUNG!** Diese Funktion ist noch nicht ausgereift – AC1 hängt fest und kann mit BREAK (Strg + ESC) wieder aktiviert werden. – Also am besten nicht verwenden!
- Im **SCCH – Grafik - Basic** können als 1-Byte-Kommandos verwendet werden:
 - 12H (Strg +R) STOP (zurück in Direktmodus, mit CONT ggf. weiter)
 - 13H (Strg +S) BREAK (Halt, weiter mit beliebiger Taste)
 - 14H (Strg +T) LIST
 - 15H (Strg +U) RUN
 - 16H (Strg +V) CLOAD „“ <CR>
 - 17H (Strg +W) CSAVEMit Ausnahme von STOP und BREAK schreiben diese Steuerzeichen allerdings auch nur vermittels der im RAM ab 1F80H definierten „F-Tastenbelegung“ die entsprechenden Befehle als Text auf den Schirm und führen sie dann aus.
- Die Belegung einer Funktionstaste kann wie folgt **gelöscht** werden:
 - AltGr +Druck
 - die zu löschende F-Taste
 - noch mal die zu löschende Taste

Anmerkung:

Damit keine falsche Meinung auf kommt, möchte ich ausdrücklich darauf verweisen, dass das Programm nicht von mir erstellt wurde, sondern nur einige Tastenänderungen vorgenommen wurde, da man der Meinung war, das #7F nicht auf die übliche Taste gelegt werden kann (BS).

Je nach Zeichengeneratur kann bei #60 entweder das Zeichen ° oder @ auf dem BS erscheinen.

Schon seit erscheinen des Monitor V.8 und 10/88 wurde im Programm das Byte #5F nicht mehr als BS genutzt, sondern als Unterstrich, so das als BS nur noch, wie auch im Monitor V.3.1, mit #7F erfolgt.