

# picoAC1

Nachbauprojekt von Frank Heyder und Jörg Reul

Vor 40 Jahren startete mit der Ausgabe 12/1983 der Zeitschrift Funkamateur die Artikelserie zum Selbstbauprojekt Amateurcomputer AC1. Im Juni 2023 überlegten Frank und Jörg, wie ein kompatibler AC1 heute aussehen könnte, und starteten das Projekt picoAC1 mit dem Ziel, es im Heft 12/2023 der Zeitschrift Funkamateur vorzustellen und alle Unterlagen zum Nachbau im Internet zur Verfügung zu stellen.

## Links

- <https://www.bw38.de/picoac1> Projektseite
- <https://codeberg.org/bw38/picoAC1> Git-Archiv zum Projekt

## Anschluss

Der Einplatinencomputer wird mit einem 5V-USB-Netzteil (z.B. von einem alten Handy, aber kein Schnelllade-Netzteil!) und ein USB-C-Kabel mit Strom versorgt. Der USB-C-Anschluss zur Stromversorgung befindet sich auf der Leiterseite unterhalb der SD-Karte. Außerdem wird eine Standard-USB-Tastatur und ein Monitor via VGA-Kabel angeschlossen.



Programme und Dateien werden einer Mini-SD-Karte gespeichert (s.u.)

Nach dem Anschließen der Stromversorgung sollte der picoAC1 ein Bild ähnlich wie folgt anzeigen:

## TODO

Die mittige LED kennzeichnet den AC1-Betriebsmodus. Im CP/M-Betrieb ist sie dunkel <sup>1)</sup>

## Bedienung

Direkt nach Anschluss der Stromversorgung ist der Computer einsatzbereit - es gibt keinen Boot-Vorgang oder ähnlich. Das Betriebssystem (der sogenannte Monitor) ist im EPROM gespeichert und wird direkt gestartet.

Das Standard-System des picoAC1 ist eine Erweiterung des originalen Monitors 3.1 [Monitor](#). Statt des Minibasics gibt es zusätzliche Kommandos zur Arbeit mit der SD-Karte. Das Minibasic ist trotzdem mit dem Kommando „Z“ verfügbar (s. [externe\\_funktionen](#)).

## Tastatur

Der Original-AC1 kennt nur Großbuchstaben. Das muss bei der Eingabe beachtet werden. In Programmen wie z.B. [minibasic](#) sollte man mit **Capslock** auf Großbuchstaben umschalten, sonst gibt es nur fehlerhafte Eingaben!

Folgende Tasten bzw. Tastenkombinationen lösen spezielle Funktionen aus. Bei Shift und Strg wird zwischen linker und rechter Taste unterschieden.

<b>Reset</b>	
Left-Strg + Alt + Entf	Z80 - Reset
<b>Betriebssystem-Varianten</b>	
Left-Strg + Alt + F1	Monitor 1: Ur - AC1 V3.1 und picoAC1 Erweiterung
Left-Strg + Alt + F2	Monitor 2: Ur - AC1 V3.1 und Minibasic (das Original)
Left-Strg + Alt + F3	Monitor 3: SCCH V1088
Left-Strg + Alt + F4	Monitor 4: SCCH V11
<b>Zeichengeneratoren</b>	
Alt + F5	AC1 - 6 x 8 Pixel (das Original)
Alt + F6	AC1 - 9 x12 Pixel
Alt + F7	AC1 - 10 x14 Pixel
AltGr + F5	SCCH - 6x 8 Pixel
AltGr + F6	SCCH - 9x12 Pixel
AltGr + F7	SCCH - 10x14 Pixel
<b>Z80 - Taktfrequenz</b>	
Alt + F9	2MHz
Alt + F10	4MHz
Alt + F11	6MHz
Alt + F12	8MHz
<b>Print to File</b>	
Left-Strg + Alt + P	Druckdatei initialisieren / abschließen
<b>Lock-Tasten, alternierend</b>	

Capslock	Umschaltung A-Z (Ä, Ö, Ü) ↔ a-z (ä, ö, ü)
Scrolllock	Tastaturausgabe - Dauer bzw Einzelimpuls (aktiv)
Numlock	frei
<b>Hilfsfunktionen</b>	
Left-Strg + Capslock	virtuelle Scrolllock-Funktion f. Tastaturen ohne Scrolllock-Taste
Left-Strg + Left-Win + ?	Speichern der aktuell verwendeten Scancode-Tabelle auf SD-Karte
Left-Strg + Left-Win + ...	Anzeige des von der jeweiligen Taste erzeugten Scancodes in der Statuszeile (2-stellig hex), nur: 01h .. 67h

## SD-Karte

Zur Daten- und Programmspeicherung wird eine normale Mini-SD-Karte genutzt. Die SD-Karte ist am PC mit FAT32 oder exFAT zu formatieren. Das ermöglicht einen einfachen Datei-Transfer mit dem PC.

### Datei-/Pfadnamen

Datei-/Pfadnamen sind Zeichenketten mit einigen Besonderheiten. Namen sind **caseinsensitive**, d.h. es wird nicht zwischen Klein- und Großbuchstaben unterschieden. Dateinamen dürfen max. **16 Zeichen** lang sein.

Der '.' (Punkt) gilt als normales Zeichen und darf an beliebiger Stelle und beliebig oft vorkommen. Ein '.' an erster Stelle kennzeichnet die Datei als „versteckt“. Diese werden in Auflistungen nicht angezeigt, können aber ansonsten normal verwandt werden. Dieses Feature ist für Systemdateien und -verzeichnisse vorgesehen.

Es dürfen Sonderzeichen entsprechend der allgemeinen Regeln für FAT-Dateisysteme verwandt werden. Die Verwendung von Wildcards ('\*', '?') ist unzulässig. Ebenso ist die Verwendung des Slash ('/') in Datei oder Pfadnamen nicht zulässig.

Es wird empfohlen, Datei- und Pfadnamen nicht ausschließlich aus Ziffern zu bilden. Auch am PC vergebene Dateinamen sollten nicht länger als 16 Zeichen sein und keine Leerzeichen enthalten.

Es ist **genau eine Verzeichnisebene** unterhalb von Root nutzbar. Auf weitere Unterverzeichnisse kann über die Monitorerweiterung nicht zugegriffen werden.

## Beispiel

TODO

### Systemdateien

Es gibt mehrere spezielle Dateien, die vom picoAC1 intern genutzt werden:

#### **.config**

In der „.config“ werden Systemeinstellungen automatisch gespeichert und beim Masterboot gelesen.

#### Verzeichnis **.ac1bin**

Im Verzeichnis „ac1bin“ werden überschriebene bzw gelöschte Dateien abgelegt. Der jeweilige Dateiname wird um einen Zeitstempel erweitert. Die Dateien lassen sich am PC durch umbenennen und verschieben wiederherstellen.

### **.scan2ascii**

individuelle Tastaturbelegung. Erläuterungen s. Keyboard

### **.hklist**

Hotkey -Liste. Erläuterungen s. Keyboard

## **Monitor-Kommandos**

im System „picoAC1 Monitor V3.1 plus“ stehen statt des Minibasic im EPROM zusätzliche Monitorkommandos. Diese dienen vor allem der Arbeit mit der SD-Karte.

Die zusätzlichen Monitorkommandos sind **Kleinbuchstaben**, die originalen Kommandos des [Monitors 3.1](#) i.d.R. **Großbuchstaben**.

Monitorkommandos bestehen aus einem Buchstaben/Zeichen und Argumenten.

Dem Kommando-Kennbuchstaben können je nach Befehl 1 bis 3 Argumente folgen. Diese werden wie im AC1 üblich in 1 bis 4-stelligen Hex-Zahlen angegeben. Für die Zeichen 'A' bis 'F' sind Großbuchstaben zu verwenden.

Einige Kommandos verlangen die Eingabe einer Zeichenkette. Diese ist nach den Argumenten bzw. dem Kennbuchstaben einzugeben. Von einem Argument ist die Zeichenkette durch mindestens ein Leerzeichen zu trennen. Nach einem Kennbuchstaben ist kein Leerzeichen erforderlich. Bis zum Anfang der Zeichenkette sind bis zu 9 Leerzeichen zulässig.

Die Zeichenkette wird durch ein Leerzeichen abgeschlossen. Nachfolgende Zeichen werden ignoriert. D.h. Zeichenketten können keine Leerzeichen beinhalten.

### **s aaaa eeee ssss Dateiname/Infotext (save)**

Hiermit kann man eine File ab der Adresse aaaa bis zur Adresse eeee mit der Startadresse ssss unter dem Namen Dateiname (Länge max 16 Zeichen) auf SD-Karte abspeichern. Dem Dateinamen kann ein Infotext angefügt werden. Beide Teile werden durch '/' getrennt. In diesem Fall wird im Header nicht der Dateiname sondern die Info gespeichert, die beim Laden angezeigt wird. Leerzeichen sind in der Gesamtheit nicht möglich.

### **l Dateiname/Nr (list/load)**

l (kleines „L“) dient zum Auflisten bzw. Laden von Daten von der SD-Karte. Wird kein Dateiname angegeben oder enthält Dateiname das Wildcardzeichen „\*“, so wird das aktuelle Verzeichnis aufgelistet. Angezeigt werden alle Dateien, auf die Maske passt. Vorangestellt ist jedem Dateinamen eine Nummer (Index). Anhand dieser Indexnummer kann eine Datei geladen werden. Die Sortierung kann mit + und - gesteuert werden: l + → Sortierung nach Änderungsdatum aufsteigend, l - → Sortierung nach Änderungsdatum absteigend.

Wird ein Dateiname oder eine Nummer angegeben, wird die Datei im aktuell gewählten Verzeichnis gesucht und ab der im Header angegebenen Anfangsadresse in den RAM geladen.

Danach wird die Startadresse im Header nach ARG1 geladen, sodass das Programm mit „J:“ gestartet werden kann. Die restlichen Headerparameter werden nur angezeigt.

### **a AADR Dateiname (absolute)**

Lädt die Datei an Absolut-Adresse AADR. AADR < 1000H werden zurückgewiesen. Weiteres Verhalten wie beim „I“-Kommando.

### **r Dateiname (remove)**

Verschiebt Datei in den Papierkorb-Systemordner „ac1bin“. Am PC können Dateien in diesem Verzeichnis „wiederbelebt“ oder endgültig gelöscht werden.

### **d Verzeichnisname/Nr (directory)**

Ohne Parameter werden alle Verzeichnisse aufgelistet. Enthält Verzeichnisname das Wildcardzeichen „\*“, so werden nur passende Verzeichnisse aufgelistet. Die Sortierung kann mit + und - gesteuert werden (vgl. Kommando „I“).

Wird ein Verzeichnisname oder eine Nummer angegeben, wird in das entsprechende Verzeichnis gewechselt. Mit „d /“ wird ins Root-Verzeichnis gewechselt.

Es wird mit einer Verzeichnisebene unterhalb von Root gearbeitet. Weitere Unterverzeichnisse werden nicht unterstützt. Beim Auflisten wird das aktuell gewählte Verzeichnis mit einem '\*' markiert. Es ist möglich, direkt von einem in ein anderes Verzeichnis zu wechseln. Ein Umweg über das Root-Verzeichnis ist nicht erforderlich.

### **m Verzeichnisname (makedir)**

Es wird ein neues Verzeichnis unterhalb von Root angelegt. Es ist dabei unerheblich, ob man sich aktuell im Root- oder einem Unterverzeichnis befindet. Das Löschen von Verzeichnissen kann nur am PC erfolgen.

### **t YYYY-MM-DD\*hh:mm:ss (time)**

Ohne Parameter wird die aktuelle Uhrzeit angezeigt, mit Parameter wird die Uhr gestellt. Die Systemzeit ist dabei in UTC (MEZ-1h | MESZ-2h) zu setzen. Diese Zeitbasis wird zum Speichern auf SD-Karte benötigt. Andernfalls wird am PC ein falscher Zeitstempel angezeigt!

### **c v:h:r (color)**

Setzt die drei Bildschirmfarben für Vordergrund, Hintergrund und Rahmen. Ohne Parameter wird die Farbliste ausgegeben. Der dritte Parameter ist optional. Bei weglassen wird '0' = Schwarz angenommen. Beispiel: „c 7:0:3“

### **H (help) h (help)**

Anzeige der Bedienung des Ur-Monitors 3.1 mit H, die Bedienung der Monitorerweiterungen mit „h“.

## **Drucken in Datei**

Left-Strg + Alt + p

Dabei handelt es sich um eine Hilfsfunktion, um Zeichen, die von Anwendungsprogrammen an eine Druckerschnittstelle im Monitor übergeben werden, in eine Textdatei umzulenken. Diese Dateien können **NUR** am PC betrachtet, bearbeitet oder gedruckt werden. Gestartet wird die Funktion mit der Tastenkombination Left-Strg + Alt + P. Dabei wird automatisch ein Dateiname PXXXX.txt erzeugt, wobei XXXX eine fortlaufende Nummer ist, die in der .config gespeichert wird. Danach ist das Anwenderprogramm zum Drucken zu veranlassen und nach Abschluss des Vorgangs ist wiederholt Left-Strg + Alt + P zu betätigen. Erst dann wird die Datei auf der SD-Karte gespeichert. In der Statuszeile wird Dateiname und Fortschritt angezeigt.

## Externe Funktionen

Funktionen / Kommandos, die nicht im EPROM Platz gefunden haben oder weichen mussten, lassen sich im Flash des picoIO einbinden und per Kommandozeichen über die ZCOM-Schnittstelle in den RAM laden und starten.

Aktuell eingebundene Kommandos:

```
Z --> MiniBasic-Interpreter nach 0F000h laden und dort starten.
```

Beim Verlassen wird angeboten, das Basic-Programm per Kommandozeile zu speichern. Hierzu ist das Prompt-Symbol '#' zu löschen und ein individueller Name zu vergeben.

## Hardware

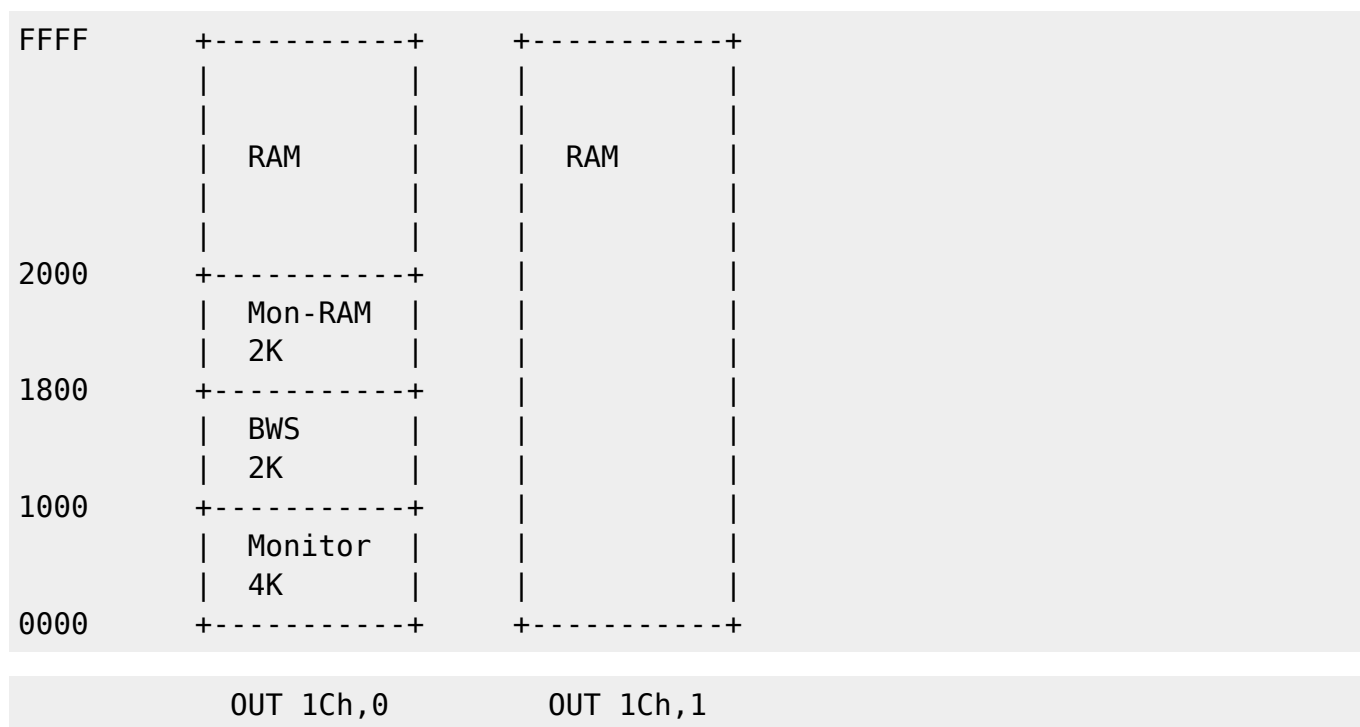
Die Funktionsweise der Hardware ist auf <https://www.bw38.de/picoac1> ausführlich beschrieben. Hier nur ein paar Bemerkungen:

Merkmal	Beschreibung
CPU	U880
ROM	4K 0000-0FFF (Monitor)
RAM	64K 1000-FFFF (ab 0000 im CP/M-Modus)
Takt	1,2,4,8 MHz
Anzeige	64x32 Zeichen, s/w, versch. Zeichensätze möglich
Tastatur	ASCII
Peripherie	2 PIO, 1 CTC
Port	Verwendung
00-03	CTC
04-07	PIO
1C(-1F)	CP/M-Umschaltung

Der picoAC1 entspricht einem AC1 mit 64K RAM und CP/M-Umschaltung. Auf der Platine sitzt ein 128K-SRAM. Dessen Kapazität kann allerdings nicht komplett genutzt werden. Die beiden 64K-Bänke werden parallel genutzt. Die untere Bank (A16=L) wird als Bildwiederholtspeicher und Arbeitszellenspeicher genutzt (1000h-17FFh), auf andere Bereiche der Bank ist kein Zugriff möglich! Die obere Bank (A16=H) wird als normaler RAM ab 2000h, genutzt, im CP/M-Modus durchgehend ab 0000-FFFFh.

Die Umschaltung in den CP/M-Modus erfolgt durch ein Latch auf Port 1Ch(..1Fh). OUT 1Ch, Bit0=0 schaltet den AC1-Modus ein (Monitor-ROM, BWS aktiv), OUT 1Ch, Bit0=1 schaltet den CP/M-Modus ein (durchgehend RAM).

<ditaa noedgesep>



</ditaa>

## CP/M

Für den Betrieb von CP/M sind auf der SD-Karte diverse Dateien abzulegen, am besten in einem eigenen Verzeichnis.

TODO

Der Bildschirmtreiber nutzt die Cursor-Positionierung nach SCP. Steuerzeichen:

```

; Bildschirm-Steuerzeichen, SCP-kompatibel
; 00h      NOP (keine Wirkung)
; 01h      Cursor links oben (home)
; 07h      akustisches Zeichen
; 08h      Cursor zurueck
; 0ah      Linefeed (neue Zeile)
; 0ch      Bildschirm loeschen (verzoegert zum Lesen der
;          zuletzt ausgegebenen Bildschirmzeilen), Cursor
;          links oben
; 0dh      Carriage Return (an Zeilenanfang)
; 14h      Rest des Bildschirms loeschen
; 15h      Cursor nach rechts
; 16h      Rest der Zeile loeschen

```

```
; 18h      Zeile loeschen, Cursor an Zeilenanfang
; 1ah      Cursor eine Zeile hoch
; 1bh      Einleitung Cursorpositionierfolge, die naech-
;          sten beiden Bytes beinhalten Zeile und Spalte
;          Offset 00h oder 80h
; 7fh      Delete (streichen Zeichen links vom Cursor)
; 20H..7EH ASCII-Zeichen
```

Hier die **diskdefs** für die [Cpmttools](#) (ohne Gewähr!)

```
# 800K AC1PICO-Hausformat
diskdef ac1
  seclen 512
  tracks 10
  sectrk 160
  blocksize 2048
  maxdir 256
  skew 0
  boottrk 0
  os 2.2
end
```

```
# 8MB AC1PICO-Hausformat
diskdef ac1_8mb
  seclen 512
  tracks 32
  sectrk 512
  blocksize 4096
  maxdir 512
  skew 0
  boottrk 0
  os 2.2
end
```

1)

nur bei BWS-Zugriffen wird sie kurz aktiviert

From:  
<https://hc-ddr.hucki.net/wiki/> - **Homecomputer DDR**

Permanent link:  
<https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/ac1/picoac1?rev=1713089775>

Last update: **2024/04/14 10:16**

