

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 ¹⁾

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.

| | |
|----------------------------------|---|
| Reset | |
| Left-Strg + Alt + Entf | Z80 - Reset |
| Betriebssystem-Varianten | |
| 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 |
| Zeichengeneratoren | |
| 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 |
| Z80 - Taktfrequenz | |
| Alt + F9 | 2MHz |
| Alt + F10 | 4MHz |
| Alt + F11 | 6MHz |
| Alt + F12 | 8MHz |
| Print to File | |
| Left-Strg + Alt + P | Druckdatei initialisieren / abschließen |
| Lock-Tasten, alternierend | |

| | |
|----------------------------|--|
| Capslock | Umschaltung A-Z (Ä, Ö, Ü) ↔ a-z (ä, ö, ü) |
| Scrolllock | Tastaturausgabe - Dauer bzw Einzelimpuls (aktiv) |
| Numlock | frei |
| Hilfsfunktionen | |
| 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 - save - Daten auf SD-Karte schreiben

```
s AAAA EEEE SSSS Dateiname/Infotext
```

AAAA Anfangsadresse

EEEE Endadresse

SSSS Startadresse

Dateiname - zulässige Zeichen entspr. FAT32, keine Wildcards, nicht casesensitiv

Infotext (optional)

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.

I - list/load - Daten von SD-Karte laden/aufisten

```
l Dateiname (kleines "L")
```

Die Datei wird 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.

Dateiinhalte des aktuellen Verzeichnisses auflisten

```
l *      --> gesamten Inhalt anzeigen. Der "*" kann entfallen.  
l XX*   --> alle Dateien anzeigen, die mit "XX" beginnen  
l *.bas --> alle Dateien anzeigen, die mit ".bas" enden
```

Das „l“-Kommando bezieht sich auf das aktuell ausgewählte Verzeichnis. O.g. Kommandos geben eine alphabetisch sortierte Liste aus.

```
l + --> Sortierung nach Änderungsdatum aufsteigend  
l - --> Sortierung nach Änderungsdatum absteigend
```

Dateilisten sind auf 256 Einträge eines Verzeichnisses begrenzt.

Laden von Dateien an Hand der Indexnummer

Mit dem l-Kommando können Inhalte des aktuell gewählten Verzeichnisses angezeigt werden. Vorangestellt ist jedem Dateinamen eine lfd Nummer (Index). Anhand dieser Indexnummer kann eine Datei geladen werden. Beispiel 1:

```
l 5 --> lädt die Datei mit der lfd Nummer 05.
```

Beispiel 2:

```
l - --> Anzeigen der Dateinamen, aktuellste Datei zuerst  
l 1 --> lädt die zuletzt gespeicherte Datei
```

Die Nummerierung richtet sich immer nach der zuletzt angezeigten Liste.

Nebenwirkung: Es sollten keine Datei- und Verzeichnisnamen vergeben werden, die ausschließlich aus numerischen Zeichen bestehen. Mindestens ein Buchstabe, Sonderzeichen oder Punkt sollte enthalten sein.

a - addr - Laden von Dateien an Absolut-Adresse im RAM

```
a AADR Dateiname
```

AADR < 1000H werden zurückgewiesen. Es wird nur die Anfangsadresse geprüft. Weiteres Verhalten wie beim „l“-Kommando

Beispiel:

```
a C000 5
```

Lädt die 5.Datei der zuletzt ausgegebenen Liste nach 0C000H.

r - remove - Datei in Papierkorb verschieben

```
r Dateiname
```

Der Dateiname wird um einen TimestampSuffix der aktuellen Systemzeit (.YYMMDDhhmmss) ergänzt und die Datei in den Systemordner „ac1bin“ auf der SD-Karte verschoben. Am PC können Dateien in diesem Verzeichnis „wiederbelebt“ oder endgültig gelöscht werden.

d - directory - Verzeichnisse anzeigen / wechseln

```
d Verzeichnisname --> Setzen des aktuellen Arbeitsverzeichnisses
d* --> Auflisten aller Verzeichnisse, alphabetisch sortiert, '*' kann
entfallen
d+ --> Liste nach Dateidatum sortiert, aufsteigend
d- --> Liste nach Dateidatum sortiert, absteigend
d *abc* --> auflisten aller Ordner, die die Zeichenfolge "abc" im Namen
enthalten.
d / --> wechseln ins Root-Verzeichnis
d1 --> wechseln ins Verzeichnis mit ldf Nummer(1), entspr. der zuletzt
ausgegebenen Liste
```

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 - makedir - Verzeichnis anlegen

```
m Verzeichnisname
```

Es wird ein neues Verzeichnis unterhalb von Root angelegt. Es ist dabei unerheblich, ob man sich aktuell im Root- oder einem Unterverzeichnis befindet. Es kann kein Verzeichnis angelegt werden, wenn bereits eine gleichnamige Datei im Wurzelverzeichnis existiert. Das Löschen von Verzeichnissen kann nur am PC erfolgen.

t - time - Zeitfunktionen

```
t YYYY-MM-DD*hh:mm:ss
t 2023-09-03*18:05:45 --> Systemzeit setzen
```

```
t --> Anzeige der aktuellen Systemzeit
```

Falls ein RTC-Modul installiert ist, wird die Zeit beim picoIO-Start übernommen. Die Systemzeit ist 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.

Beim Setzen der Systemzeit ist das Format einzuhalten. Entscheidend ist die Position der Ziffern in der Zeichenkette. Die Trennzeichen sind beliebig. Die Eingabe erfolgt als Zeichenkette und unterliegt o.g. Einschränkungen, d.h. Leerzeichen sind unzulässig.

c - color - Bildschirmfarben

```
c 7:0:3 -> Zeichen-, Hintergrund- und Rahmenfarbe  
c --> Farbliste ausgeben
```

Der dritte Parameter ist optional. Bei weglassen wird '0' = Schwarz angenommen.

h/H - help - Hilfetexte ausgeben

```
H --> Bedienung "Ur-"Monitor"  
h --> Bedienung "Monitorerweiterungen"
```

Drucken in Datei

```
LStrg + 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 LStrg + 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 LStrg + 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 | 64×32 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 Bildwiederholpeicher 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, 0 schaltet den AC1-Modus ein (Monitor-ROM, BWS aktiv), OUT 1Ch, 1 schaltet den CP/M-Modus ein (durchgehend RAM).

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=1711873885>

Last update: **2024/03/31 08:31**

