

Vorwort:

picoAC1-2026 ist eine Überarbeitung von picoAC1, in die alle nachträglichen Erweiterungen durch die Firmware 1.4 wie NMI-Taste, Zeichengeneratorumschaltung im SCCH-Mode und 8 schaltbare Monitorvarianten eingearbeitet sind. Darüber hinaus wurde die Unterstützung des AC1-2010-Bus erweitert (A8-A15, /IOSEL5, /MEMDI2, MEMDI8, CP/M-U). Hierzu ist die Speicher- und IO-Decodierung durch zwei GAL 16V8 realisiert. Mit normalen Logik-IC's hätte dies nicht mehr auf die Leiterplatte gepasst. GAL 16V8 werden aktuell wieder von Atmel/Microchip hergestellt. Außerdem können hier nachträglich einfach Modifikationen vorgenommen werden. Die Firmware von PicoIO und PicoVDU ist identisch mit dem ursprünglichen picoAC1. picoAC1-2026 ist abwärtskompatibel zum ursprünglichen picoAC1.

Durch die Erweiterungen ergeben sich neue Funktionen wie:

- Nutzung der Modul1-Leiterplatte am picoAC1-2026, Paket-X
- Anschluss des AC1-2010-Bus-Analysators mit Schrittbetrieb zur Fehlersuche
- Möglichkeit der Verwendung von zwei Speicherbänken im Bereich 0x2000 bis 0xFFFF
- Implementierung der Funktion „Durchschreiben“ durch den Monitor-EPROM in den darunterliegenden RAM, mit der Möglichkeit, einen Monitor direkt aus dem RAM auszuführen (Y-Befehl). Hierzu wurden die Monitore 10/88 und 11 entsprechend angepasst.

Durch die Abkündigung der Z80-CMOS IC im DIP-Gehäuse sind diese leider mittlerweile zur Mangelware geworden. Noch vorhandene oder leichter zu beschaffende NMOS-Varianten funktionieren im Regelfall auch, benötigen aber deutlich mehr Strom und gestatten die höheren Taktraten (8MHz, 12MHz) meist nicht.

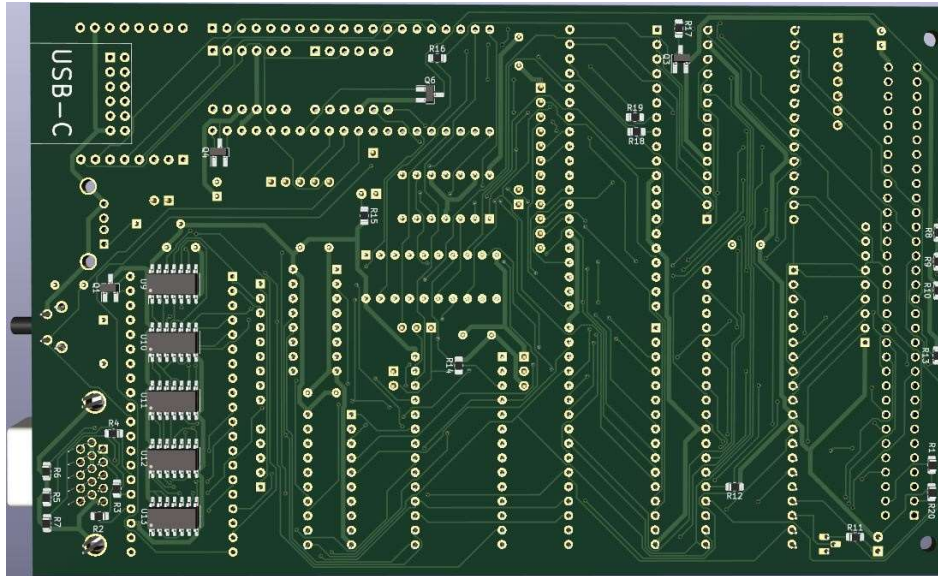
Aufbauanleitung:

Anmerkung vorab: Die VGA-Buchsen haben leider teilweise größere Toleranzen und lassen sich dadurch dann ggf. nicht ohne Vorbehandlung in die Lötäugen einführen. Deswegen sollte die VGA-Buchse vorab auf Passgenauigkeit getestet werden. Falls nicht alle 3 Reihen leicht in ihre Lötäugen wollen, reicht ein leichtes Zurechtbiegen. Dazu die vordere und hintere Kontaktreihe mit Hilfe einer Kante oder Leiste ein wenig zur mittleren Reihe hin biegen, schön parallel, die Mittelreihe bleibt wie sie ist. Wenn alle 3 Reihen die Lötäugen treffen, Buchse bei Seite legen, ... noch nicht verlöten.

Los gehts.

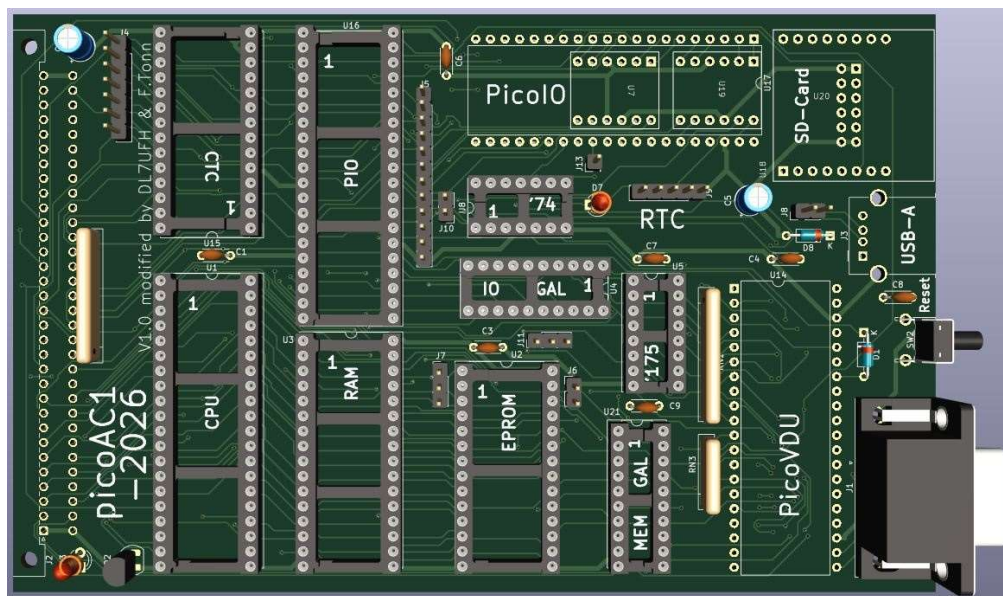
Beim Bestücken immer gut auf die Ausrichtung der Bauteile achten. Bei den Schaltkreisen musste im Interesse der Packungsdichte von allen 4 Richtungen Gebrauch gemacht werden. Pin1 hat bei den Durchsteckbauteilen ein rechteckiges Lötauge. Zusätzlich ist im Bestückungsaufdruck eine "1" als Hilfestellung aufgedruckt. Im Zweifelsfall einfach mit Hilfe von Ub und GND die richtige Ausrichtung prüfen. Es wird empfohlen, für alle IC's und die Widerstandsnetzwerke Sockel zu verwenden. Das erleichtert auch ggf. die Fehlersuche.

Schritt 1: L-Seite bestücken



- 03er auflöten
- restl. SMD auflöten
- USB-C Shield und LP-Lötseite aufrauen und mit Sekundenkleber aufkleben, mit 1 mm Draht zentrieren
- Sichtkontrolle der Lötstellen, am Besten mit Lupe/Kopflupe und viel Licht
- Kontrolle der richtigen Spannungszuführung

Schritt 2: B-Seite bestücken



- Schottky-Dioden
- Fassungen für R-Netzwerke (auf jeden Fall für die Datenbus Pullup-R)
- IC Fassungen
- Pegelwandler unter PicoIO mit Hilfe der zugehörigen Stiftleisten verlöten, LV1 bis LV4 zeigen zur Leiterplattenaußenkante
- Pico's mit Hilfe von Steckbrett oder Shield mit Stiftleisten verlöten

- Buchsenleisten für IO aufstecken, bestücken und verlöten
- VGA Buchse auflöten (ggf. vordere und hintere Reihe leicht zur Mittelreihe biegen)
- Buchsenleisten für VDU aufstecken, bestücken und verlöten
- Mikro-SD-Shield mit Hilfe der Stiftleisten verlöten, es genügt die Pin's zu löten, die gemäß Schaltbild kontaktiert werden (macht es leichter, falls es ausgetauscht werden muß)
- C's und Stiftleisten und restliche BE auflöten
- Sichtkontrolle am Besten mit Lupe/Kopflupe und viel Licht
- Kabel Mikro-USB-Stecker - Buchsenleiste 2 pol. für Anschluß D- D+ des PicoIO zur USB-A Buchse anfertigen

Schritt 3: schrittweise Inbetriebnahme

Wenn vorhanden, empfiehlt es sich eine Stromversorgung mit Strombegrenzung zu nehmen und diese für die Logik IC's auf 30 mA einzustellen. Danach sollten 150 - 200 mA reichen. Man kann den Strom für die Inbetriebnahme auch über Stiftleiste PIO (Pin1 und Pin 12) zuführen und messen. Oder man benutzt einen kleinen USB-Adapter mit Strom und Spannungsanzeige. Diese sind zwar nicht so genau, sind aber immer noch besser als gar keine Kontrolle. "Intelligente" USB-Ladegeräte sollten nicht als Stromversorgung genommen werden. Gute Erfahrungen haben wir z.B. mit der USB-Stromversorgung zum Raspberry Pi gemacht. Nach jedem Schritt die Stromzuführung wieder trennen.

PicoIO und PicoVDU vorher flashen. EPROM und GAL's brennen. Alles beschriften, damit es nicht zu Verwechselungen kommt.

Inbetriebnahme:

Die Bauteile nicht unter Spannung stecken!

1. Die 4 Logik IC's stecken Stromaufnahme ca. 17 mA
 2. PicoIO stecken 43 mA
 3. PicoVDU stecken 52 mA
 4. CPU und Datenbus Pullup's stecken, VGA Monitor anschließen 83 mA,
jetzt sollten Leerzeichen und 9 immer abwechselnd auf dem Schirm zu sehen sein
(CPU führt ständig RST 38 (OFFh) aus und schreibt 00 39 in den Stack)
 5. RAM, EPROM, Jumper entsprechend EPROM setzen, und PIO stecken 95 mA
die LED für den AC1-Mode leuchtet,
jetzt sollte die Ausschrift "picoAC1 Monitor V 3.1 plus"
auf dem Monitor erscheinen, und der Cursor blinkt
 6. CTC, und Tastatur anstecken 85 mA
(drahtlose Tastatur ca. 100 mA)
- nun sollte die Tastatureingabe funktionieren. Wenn der PicoIO eine Tastatur erkannt hat, leuchtet seine grüne LED. Ist das nicht der Fall, entweder D- D+ tauschen oder die Tastatur ist nicht geeignet. Drahtlose Tastaturen, die einen Treiber nachladen wollen, gehen nicht am PicoIO.

7. RTC und Mikro-SD-Karte stecken. Am Besten am PC schon ein mal ein paar Programme im Headersave Format aufspielen. Nun kann man mit d und l die Inhalte anschauen und ein erstes Programm laden.

picoAC1- 2026 lebt

Für einen ersten Test zur ordnungsgemäßen Funktion des Arbeitsspeichers in den AC1 und CP/M Modi wird das Programm CHECKCPM.Z80 empfohlen (laden im Monitor 1 von SD-Karte, Start mit "t" in einem SCCH-Monitor).

```
Check CPM-Umgebung SCCH(14H) & FA-Modus(1EH)      HR 30.03.11
Start mit: t <CR> startet alle Test's nacheinander
          t TestNr. AADR EADR <CR> fuer Einzeltest's
Nach Test 01 sind auf Adr.2600H je 13Byte AC1 CPM-RAM 0H & 1800H

Test01 Umschaltung AC1->CPM Mode je 13Byte RAM ab Adr.#2600H OK
Test02 Check CPM-RAM an Adresse: 0000H =00 OK
Test03 Check CPM-RAM an Adresse: 0800H =00 OK
Test04 Check CPM-RAM an Adresse: 1000H =00 OK
Test05 Check CPM-RAM an Adresse: 1800H =00 OK
Test06 Schreiben/Lesen AC1->CPM 0000H..0000 OK
Test07 Schreiben/Lesen AC1->CPM 0800H..0800 OK
Test08 Schreiben/Lesen AC1->CPM 1000H..1000 OK
Test09 Schreiben/Lesen AC1->CPM 1FFFH..1FFF OK
Test0A Schreiben/Lesen AC1->CPM 1000H..1FFF OK
Test0B Durchschreiben AC1->CPM 0000H..0FFF OK
Test0C Trennung RAM's AC1->CPM 1000H..1FFF OK
Test0D Check AC1-RAM ab Adresse: 1000H..FFFF OK
Test0E Check CPM-RAM ab Adresse: 0000H..FFFF OK

#
```

Monitor:3	Clk: 2 MHz	Kbd: CL	Z8:SCCH	Dir:Test	Prn:	11:25
-----------	------------	---------	---------	----------	------	-------

Anmerkung: Die Stromaufnahmewerte gelten für Z80-CMOS IC. Bei Verwendung von ausschließlich NMOS-Typen ist bereits bei 2 MHz Taktfrequenz mit mehr als 300 mA zu rechnen.

Viel Spaß mit picoAC1-2026.

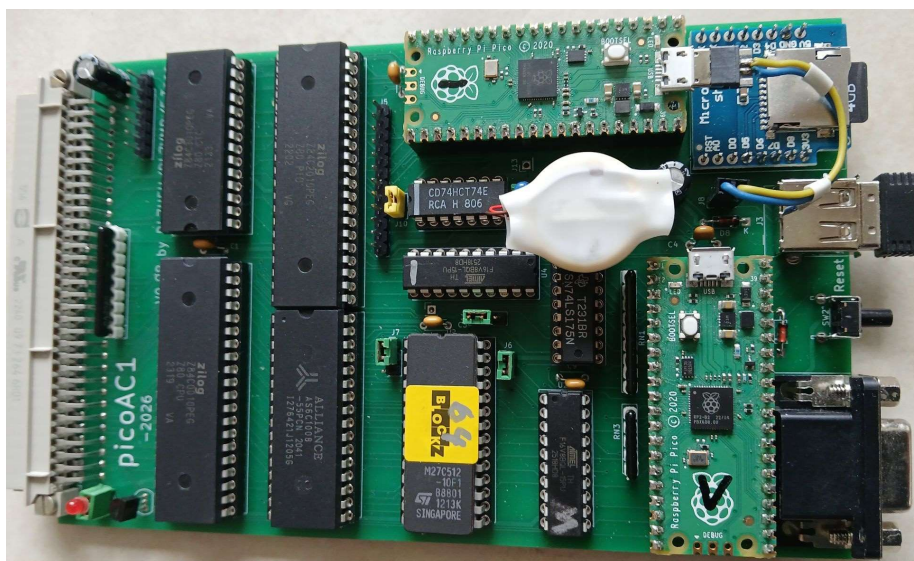


Bild: Prototyp picoAC1-2026 aufgebaut von DL7UFH