Eine VAX (und mehr) für die Hosentasche

Links:

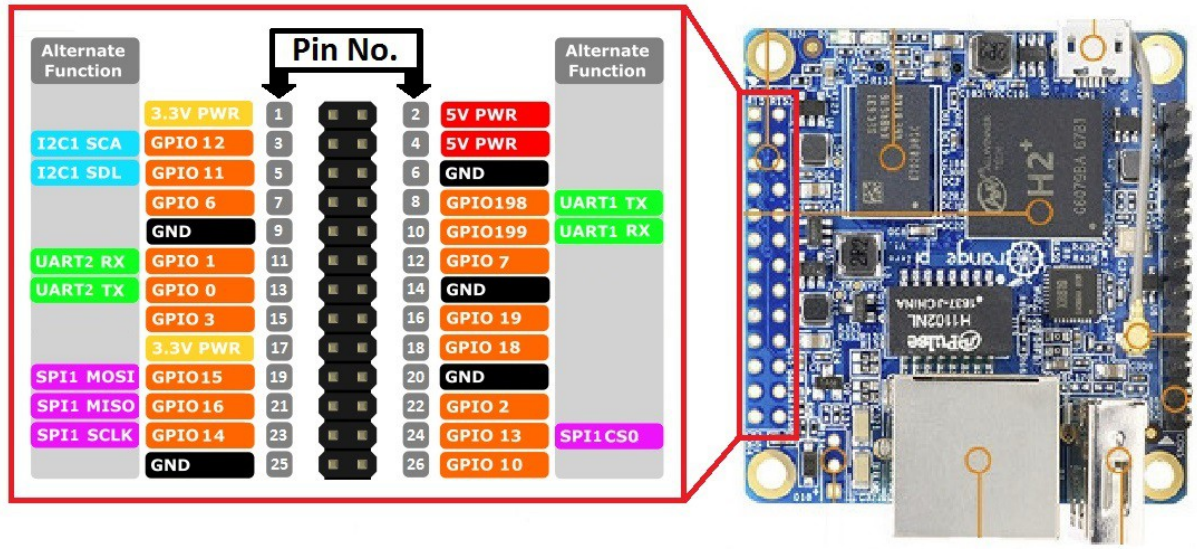
https://www.armbian.com/orange-pi-zero/

http://lucsmall.com/2017/01/19/beginners-guide-to-the-orange-pi-zero/

https://www.wherry.com/gadgets/retrocomputing/vax-simh.html

Nach Beschäftigung mit SIMH, dem Multi-Systememulator (siehe Links) unter Linux und der Verfügbarkeit immer kleinerer Linux Mini-Platinen, kam mir die Idee, daraus eine winzige VAX (Emulation) mit seriellem Anschluß für Terminals oder auch z.B. Atari ST als Terminal zu basteln.

Als Mini-Linux System fiel meine Wahl auf Orange Pi Zero. Mit ca. 5x5cm wirklich winzig. Dieses Board hat eine Allwinner Quad-Core CPU mit 1,2GHz, 256/512MB Ram, 4GB Flash, RJ45 Netzwerk, Wlan und Micro-SD Slot.



Außerdem ist eine Linux Distribution (Armbian) für diese Plattform verfügbar. Man benötigt:

- Orange Pi Zero Board (ca. 12-15€)
- 32GB Micro-SD Karte (16/8? GB sollte auch gehen) (ca. 10-12€)
- 5V/2A Netzteil mit passendem Stecker (z.B. USB Handy Lader) (ca. 5-8€)
- RS232 Wandlermodul mit 3.3V Pegeln (ebay-Arduino) (ca. 1-2€)

Das ganze Vorhaben gliedert sich in drei Teile.

1. Armbian aufspielen
2. SIMH + Emulationsumgebung aufspielen
3. Serielle Schnittstelle konfigurieren mit 9600 8N1

Armbian aufspielen

Man lädt sich die unter www.armbian.com [Links] verfügbare Debian_jessie Server Version herunter.

Ich habe verwendet: Armbian_5.24_Orangepizero_Debian_jessie_3.4.113.7z

Diese muss dann mit z.B. 7zip entpackt werden, um die SD Card Imagedatei (IMG) zu erhalten.

Diese IMG Datei wird danach mit z.B. Win32DiskImager auf die 32GB Micro-SD Karte geschrieben.

Anschließend den Orange Pi Zero mit dieser bespielten Micro-SD Karte booten mit Netzwerkverbingung und sich über Putty/etc. ins System einloggen (root/1234). Dieser gesamte Vorgang (inkl. auch serieller Verbindung) ist hier sehr gut beschrieben:

http://lucsmall.com/2017/01/19/beginners-guide-to-the-orange-pi-zero/

Ab da hat man ein Micro-Linux System, welches über Netzwerk und seriell (115200 8N1) zugänglich ist.

SIMH + Emulationsumgebung aufspielen

Anschließend folgt man dieser Anleitung um die SIMH/VAX-Emulation und VMS zu installieren:

https://www.wherry.com/gadgets/retrocomputing/vax-simh.html

Ich hatte dazu vorher alles unter z.B. Windows heruntergeladen und eingerichtet und musste daher dieses Verzeichnis mit allen Dateien (Emulierte HDD+Config) nur von einem USB Stick kopieren und SIMH/VAX unter Armbian kompilieren.

Wenn alles wie beschrieben eingerichtet ist, hat man ab da eine Micro-VAX ;-) mit ca. 5x5cm und 1,2GHz CPU, die über Netzwerk und/oder seriell 115200 Baud zugänglich ist. Neben der VAX kann man mit SIMH auch weitere Systeme emulieren. Z.B. PDP 1/8/11 etc. Auch CP/M mittels RunCPM habe ich bei mir installiert. Dazu benötigt man noch:

sudo apt-get install ncurses-dev libreadline-common

Serielle Schnittstelle konfigurieren mit 9600 8N1

Im Standard läuft alles mit 115200 baud. Wenn man die Micro-VAX nicht an älteren und langsamen Terminals anschließen will, kann man das Alles einfach so belassen und ist bereits fertig. Ansonsten habe ich folgende 2 Anpassungen gemacht, damit sowohl die Bootmeldungen (Part 1) als auch die Konsole-Login (Part 2) mit 9600 Baud möglich sind:

==== Part 1 - Change console output to 9600 baud

cd /boot

vi boot.cmd

===== Change from 115200 to 9600

if test "${console}" = "serial" || test "${console}" = "both"; then setenv consoleargs "${consoleargs} console=ttyS0,**9600**"; fi

===== Recompile with:

mkimage -C none -A arm -T script -d /boot/boot.cmd /boot/boot.scr

===== Part 2 - change baud rate of serial-getty (/sbin/agetty) to 9600 baud for ttyS0

cd /etc/systemd/system

cd serial-getty@ttyS0*

ls

10-rate.conf

vi *.conf

===== Change from 115200 to 9600

cat *.conf

[Service]

ExecStart=

ExecStart=-/sbin/agetty -L **9600** %I linux

systemctl daemon-reload

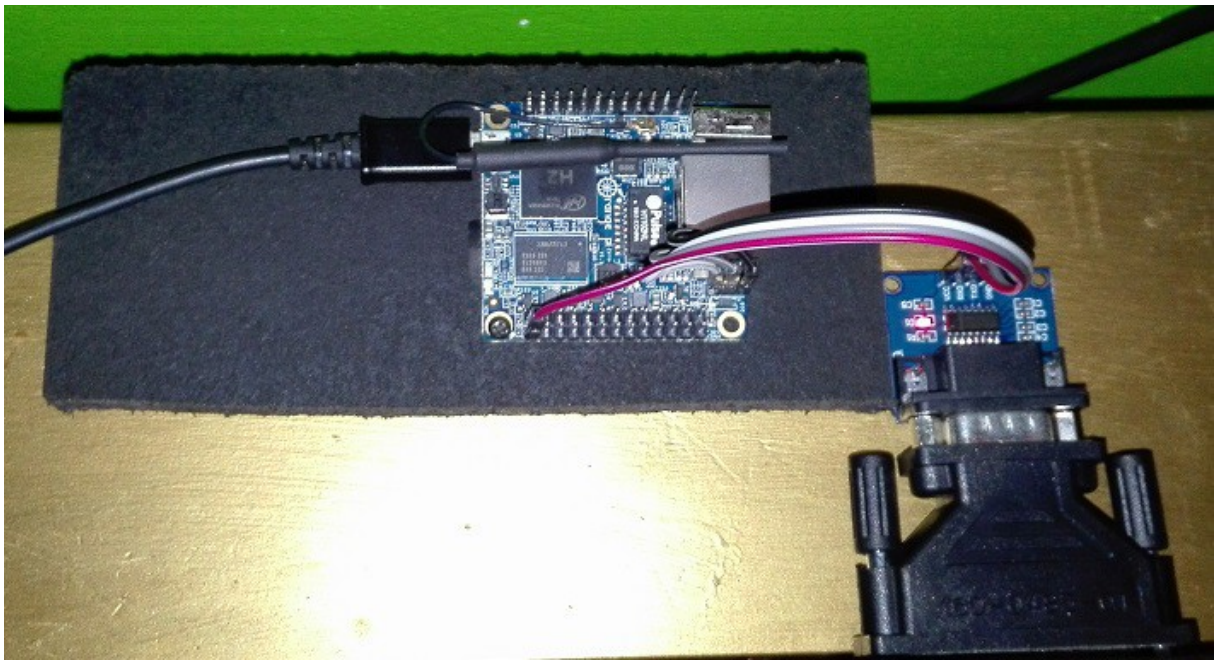
systemctl stop serial-getty@ttyS0.service

systemctl start serial-getty@ttyS0.service

```
# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 22:40 ?        00:00:03 /sbin/init
root         2     0  0 22:40 ?        00:00:00 [kthreadd]
...
root       921     1  0 22:53 ttyS0    00:00:00 /sbin/agetty -L 9600 ttyS0 linux
root       923   885  0 22:53 pts/0    00:00:00 ps -ef
#
```

3.3V sind an Pin 1 der doppelreihigen Stiftleiste des Orange Pi Zero verfügbar, um den Wandler mit Strom zu versorgen.

Und so sieht das Ganze dann fertig aus:



Und Bildschirmmeldungen/VAX:

Evtl. eignet sich dieser Setup noch für weitere 'Spielereien'. Immerhin hat man nun ein Quad-Core CPU Linux System mit Netzwerkverbindung an seinem Terminal/Atari ST..