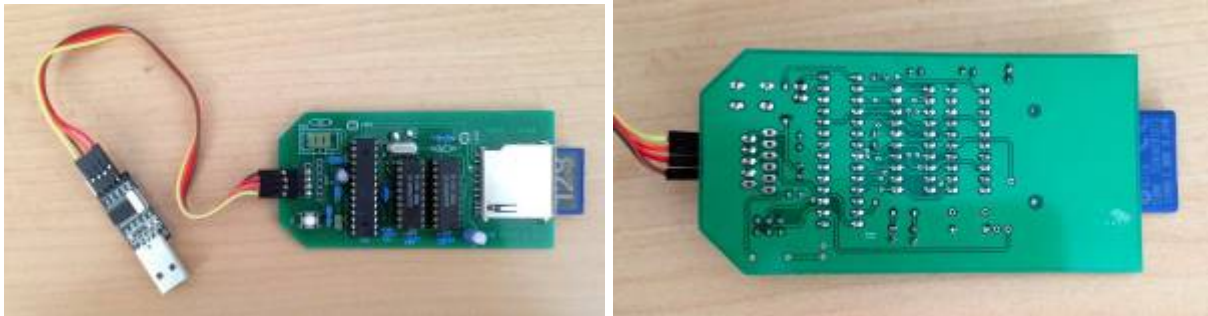


# AVRCPM

[http://www.mikrocontroller.net/articles/AVR\\_CP/M](http://www.mikrocontroller.net/articles/AVR_CP/M)

Ein kleines CP/M-System mit minimaler Hardware (AVR-CPU + Speicher). Das CP/M-System kommuniziert über eine serielle Schnittstelle mit einem Terminal (z.B. einem PC)



s.a. <http://www.retro-compi.de/index.php/avrcpm-projekte/cpm-stick>

Erfinder dieses System ist © 2010 **Sprite\_tm** (Jeroen Domburg) <http://spritesmods.com/?art=avrcpm>  
Von ihm stammen die grundlegende Minimal-Schaltung, die 8080-CPU-Emulation sowie viele tolle Ideen in der Software (z.B. die DRAM-Ansteuerung und die Nutzung von Ports zur Kommunikation mit dem AVR). <sup>1)</sup>

2010-2013 wurden im mikrocontroller.net vor allem die komplette Z80-Emulation, die Verwendung des FAT16-Filesystems anstelle spezieller CP/M-Partitionen und viele Verbesserungen in der Software und auch in der Hardware vorgenommen, die zu einem ca. fünffachen Tempo gegenüber der ursprünglichen Variante führten!

## Download

- [avrcpm.zip](#) Diskettenimages, Hex-File zum Flashen
- [avr-src-vp.zip](#) Quellcode AVR

die ehemals einzelnen Disketten stecken alle im 8MB-Image CPMSDK\_D.IMG als USER:

```
0 ASM, System
1 SPL
2 TURBO
3 Spiele
4 WS
5 MP
6 ALGOL
7 FORTRAN
8 PLI
9 C
10 HITECHC
11 MUMATH
```

## 12 AZTEC C

Den Sourcecode vom mikrocontroller.net findet man hier:

<http://www.mikrocontroller.net/svnbrowser/avr-cp-m/avrcpm/?sortdir=down>

## Bedienung

Die SD-Karte wird mit FAT16 formatiert. Auf der SD-Karte müssen Images als „Disketten“-Abbilder abgelegt sein (Name CPMDISK\_x.IMG).

Der Stick wird mit einem USB-Port des PCs verbunden. Danach sollte der Treiber (siehe unten) für den USB2TTL-Chip installiert werden – anschließend steht ein weiterer COM-Port zur Verfügung. Nun ist ein Terminalprogramm (z.B. Hyperterminal, Putty oder TeraTerm) zu starten, der Port auszuwählen und eine Übertragsrate von **115200,N,8,1** einzutragen. Nach einem Reset (Taster auf Platine) erscheint dann die Startmeldung.

## Aufbau

Hardwarevariante 3, als 8-Bit Version in USB-Stick Form ausgelegt

Version 3.1 vom Marcel A. (dl1ekm), Datum: 27.12.2014

<http://www.retro-compi.de/index.php/avrcpm-projekte/cpm-stick>

<http://www.mikrocontroller.net/topic/177481?page=4#3941030>

Die beiden benötigten 256k x 4bit-DRAMs habe ich von einer alten ISA-VGA-Grafikkarte abgelötet.

Der AVR ATMEGA328P wird mit folgenden Aufruf programmiert:

```
avrdude -c usbasp -p m328p -U flash:w:avrcpm.hex
avrdude -p m328p -c usbasp -U lfuse:w:0xf7:m -U hfuse:w:0xdf:m
```

Die SD-Karte wird mit FAT16 formatiert. Auf der SD-Karte werden die Dateien CPMDISK\_A.IMG, CPMDISK\_B.IMG, ... gespeichert. Diese Dateien („Disketten“-Abbilder) sind die Laufwerke unter CP/M.

### mein Adapter

Es ist zu beachten, dass bei den „China“-USB/Seriell-Adaptern die Beschriftung für TX und RX nicht immer eindeutig ist. I. Allg. müssen Rx/D und Tx/D über Kreuz verbunden werden.

USBTTL		CP/M-Stick			
5V	n.c.			n.c.	5V
3.3V	gelb	-----\	/-----	braun	GND
GND	braun	-----\--/	/---	rot	RX
RxD	orange	-----\-----/-----		orange	TX
TxD	rot	-----\	\---/-----	gelb	3.3V

\-----/

## Disketten-Abbilder

Es werden verschiedene Abbildformate unterstützt:

1. YAZE Format: alle Varianten  
DPB wird aus Disk ausgelesen und übernommen
2. simhd 8MB Format: simh altair 8800 hard disk Format, 8 MByte Disk, s. dpbdat\_simhd
3. MyZ80 8MB Format: 8192KB + 256 Byte Header, s. dpbdat\_myz80
4. reine 256K-Images, s. dpbdat\_avrcpm

für die CPMTOOLS sind folgende diskdefs für die beiden üblichen Formate (256K und 8MB) geeignet.

```
# AVR CP/M
diskdef avr256
  seclen 128
  tracks 77
  sectrk 26
  blocksize 1024
  maxdir 64
  skew 1
  boottrk 2
  os p2dos
end

# SIMH AltairZ80 Harddisk # AVR CP/M
diskdef avr8M
  seclen 128
  tracks 2048
  sectrk 32
  blocksize 4096
  maxdir 1024
  skew 0
  boottrk 6
  os 2.2
end
```

## simulierte Hardware

- ~ 3 MHz Z80 CPU
- ca. 64K RAM, 62K TPA !!
- Kommunikation über serielle Verbindung mit einem Terminal, 115200 Baud
- ggf. 192K-RAM-Disk oder Banking für CP/M 3 möglich
- 16 virtuelle Laufwerke

Es gibt eine Reihe Portadressen, über die die virtuelle CPU mit der AVR-Umgebung kommuniziert. s. config.inc, Z80-seitig virt\_ports.asm

## Einschaltvorgang

Nach dem Reset wird die „Hardware“ initialisiert (RAM-Test, Suche der Laufwerke, ...) Abschließend wird der erste Sektor des ersten Laufwerks gelesen. In diesem Sektor befindet sich ein 8080-Programm namens IPL (initial program loader, Urlader). Dieses Programm lädt CCP+BDOS+BIOS vom Laufwerk und startet CP/M.

## BIOS

## CP/M

bei Zugriff auf falsches Laufwerk - keine Chance, auf anderes LW zu kommen?

Turbo Pascal TINST vt100 Umstellung highlight von invers (ESC [ 7 m) auf fett (ESC [ 1 m)

# GENCPM

System auf Spur 0 schreiben:

```
A>power
A0=load cpm.bin 4000
Last Address:59FFH    52 sectors
A0=write 0 1 4000 52
```

<sup>1)</sup>

Um seine Arbeit zu würdigen, lohnt sich ein Blick in seine originalen Quellcodes!

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

Permanent link:  
<https://hc-ddr.hucki.net/wiki/doku.php/cpm/avrcpm>

Last update: **2016/12/06 14:51**

