

AVR CP/M

It really works!! Unglaublich!

8080 Emulator und CP/M mit einem ATmega88, 256kx4 DRam und SD Karte.

Noch nicht ausgereift, mehr ein Proof-of-concept. Terminal auf 38400 8N1 einstellen. Die diskimage mit dd direkt auf eine SD Karte schreiben.

Zip-Archiv mit allen Dateien: **avr-cpm.zip**

Hier cpmtools fuer win32 und ein erweitertes diskimage. Enthaeelt eine ersteinmal lauffaehige mbasic.com, allerdings sind die Sprungziele 00000? Aber das

othello Spiel laeuft: **cpmtools.zip**

Muss zwingend im Wurzelverzeichnis entpackt werden, sodass \cpmtools entsteht.

Aufrufbeispiele:

`cpmls -f avrcpm diskimage - zeigt Inhaltsverzeichnis an`

`cpmcp -f avrcpm diskimage datei.com 0:datei.com - kopiert datei.com in das image`

`cpmcp -f avrcpm diskimage 0:datei.com datei.com - kopiert datei.com aus dem image`

`cpmrm -f avrcpm datei.com - loescht datei.com aus dem image`

ipl.asm + bios.asm lassen sich hiermit assemblieren: <http://www.tni.nl/products/tniasm.html> - wenn man das end Statement auskommentiert.

Links:

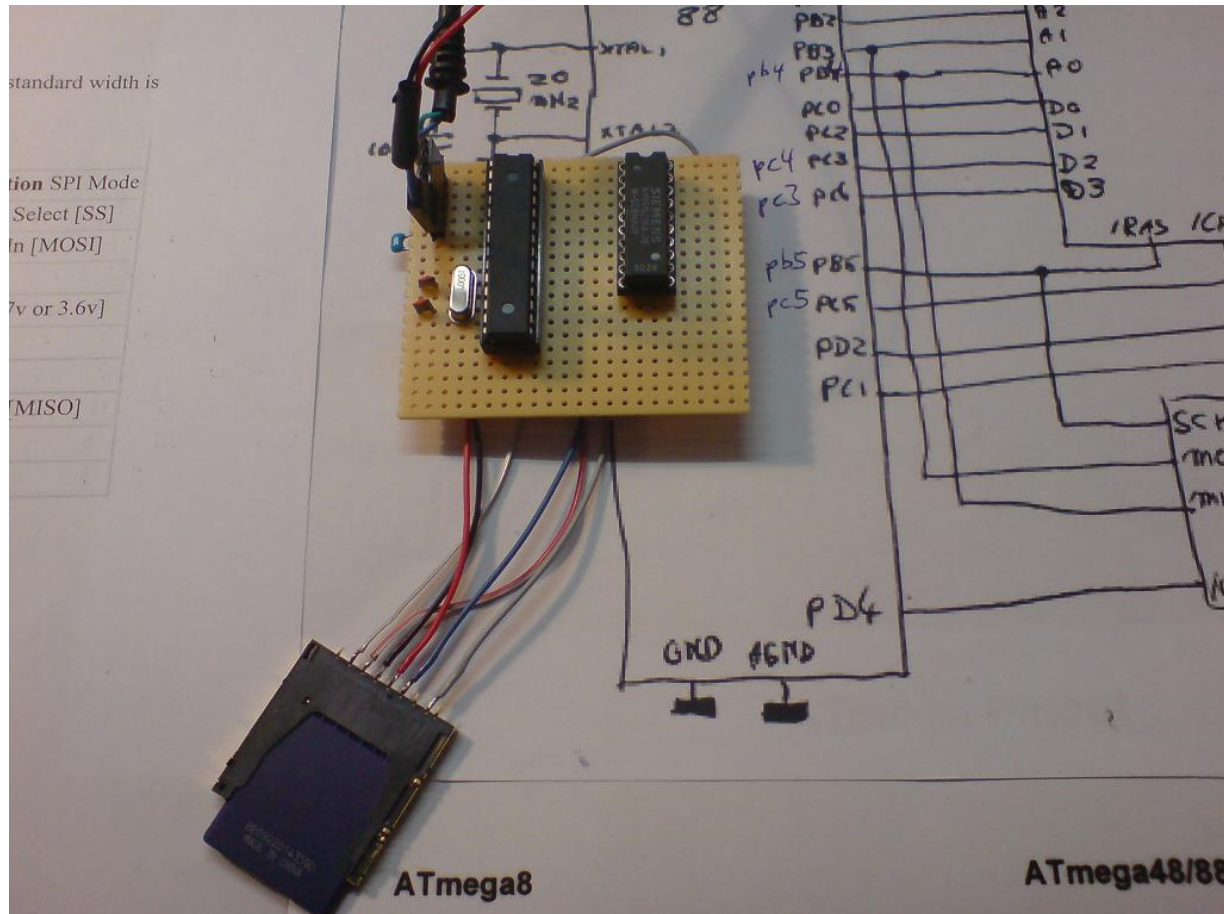
<http://spritesmods.com/?art=avrcpm>

<http://www.mikrocontroller.net/topic/177481>

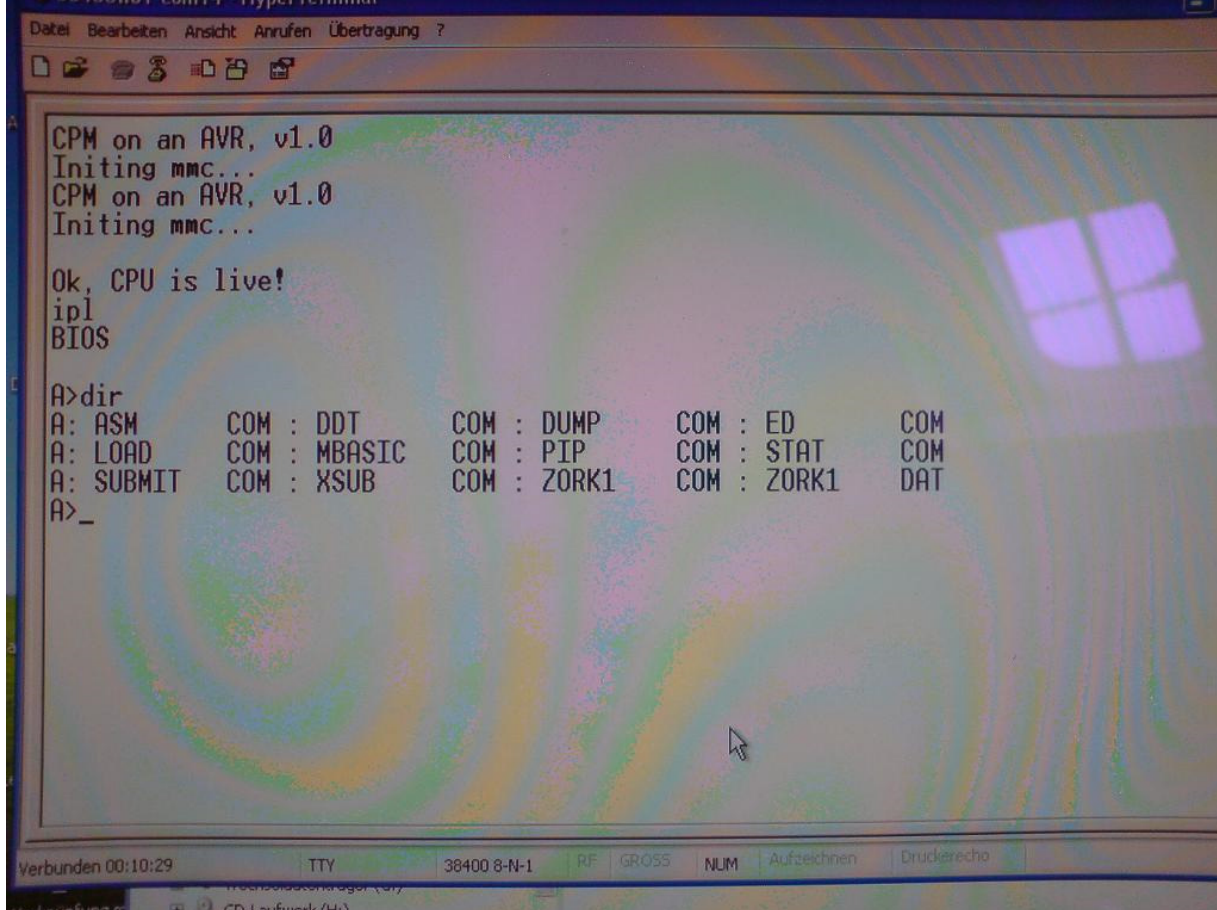
http://www.mikrocontroller.net/articles/AVR_CP/M

<http://cloudbase.homelinux.net/viewvc/avr-cpm/avrcpm/trunk/>

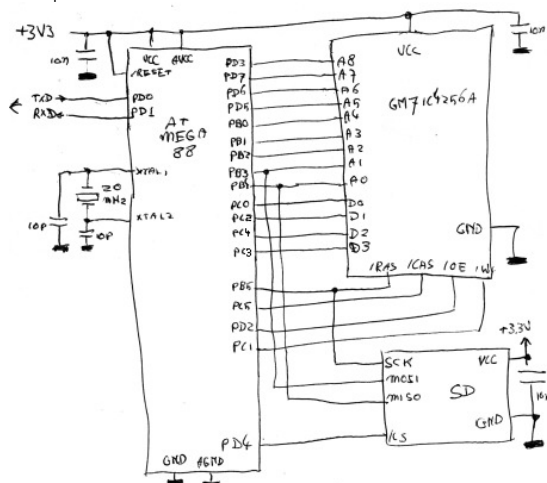
Lochrasteraufbau:



Bildschirmmeldung:



Schaltplanskizze:



Bauteile Pinouts:

ATmega88:

PDIP

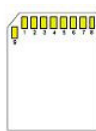
(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

Dram 256x4:

44256, 44258
256kx4 DRAM.

D0	1	20	GND
D1	2	19	D3
/WE	13	18	D2
/RAS	14	17	/CAS
	15	16	/OE
A0	16	15	A8
A1	17	14	A7
A2	18	13	A6
A3	19	12	A5
VCC	10	11	A4

SD Karte:

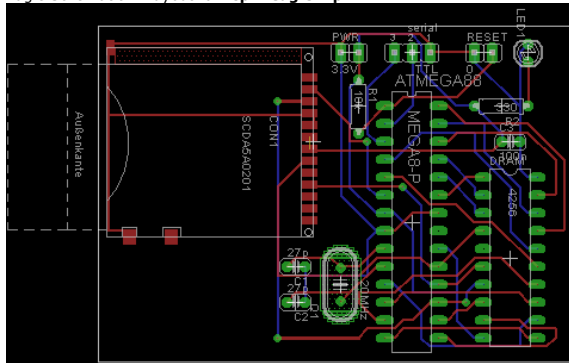


SD Card

The Secure Digital card dimensions are: 24mm wide x 32mm long. The standard width is 2.1mm, while the Thin SD Memory Card has a width 1.4mm.

Secure Digital Pinout				
Pin #	Pin Name	SD Signal Function	SD Mode	SPI Signal Function
1	DAT3/CS	Data Line 3	Chip Select/Slave Select [SS]	
2	CMD/DI	Command Line	Master Out/Slave In [MOSI]	
3	VSS1	Ground	Ground	
4	Vdd	Voltage Supply [2.7v or 3.6v]	Voltage Supply [2.7v or 3.6v]	
5	Clock	Clock	Clock [SCK]	
6	Vss2	Ground	Ground	
7	DAT0/D0	Data Line 0	Master In Slave Out [MISO]	
8	DAT1/IRQ	Data Line 1	Unused or IRQ	
9	DAT2/NC	Data Line 2	Unused	

Eagle Schematic + Layout: **avrcpm_eagle.zip**



Aus Diskussionen im Mikrocontroller Forum wurde von Prof. Dr.-Ing. habil. Joerg Grabow eine erweiterte Platine (USB, FT232RL) entworfen: **avrcpm_m.tar.gz**

Update

Hier der aktuelle Entwicklungsstand der Software: **avrcpm_upd.zip**

- * Warmstart (gehoben im Bios; in den 2 diskimages enthalten)
- * 8080 Opcodes fehlerbereinigt (MBASIC !?uft jetzt!)
- * Unterstuetzung fuer ATmega8, 88, 168
- * 2 Diskimages mit neuem Bios enthalten

Achtung: 2 Leitungen sind zu tauschen (Beschleunigung DRAM Zugriffe):

Pin 24 - PC1 - /WE

Pin 27 - PC4 - IO3

Update 2

Hier der neueste Stand (2010-08-01): **avrcpm_upd2.zip**

- * Es werden jetzt bis zu 4 Partitionen (Typ: CP/M = 52) als Laufwerke A:-D: eingebunden
- * 3 Diskimages enthalten (Basis, Spiele, BDS C-Compiler)

Ich habe bisher noch keinen direkten Weg gefunden, diese Partitionen unter Windows zu erstellen und zu befüllen. Mit einer Linux-Live-CD geht das aber z.B. mit cfdisk und dann mit z.B. dd if=diskimage of=/dev/sdh4 bs=512 etc.

Inhaltsverzeichnis der 3 Diskimages:

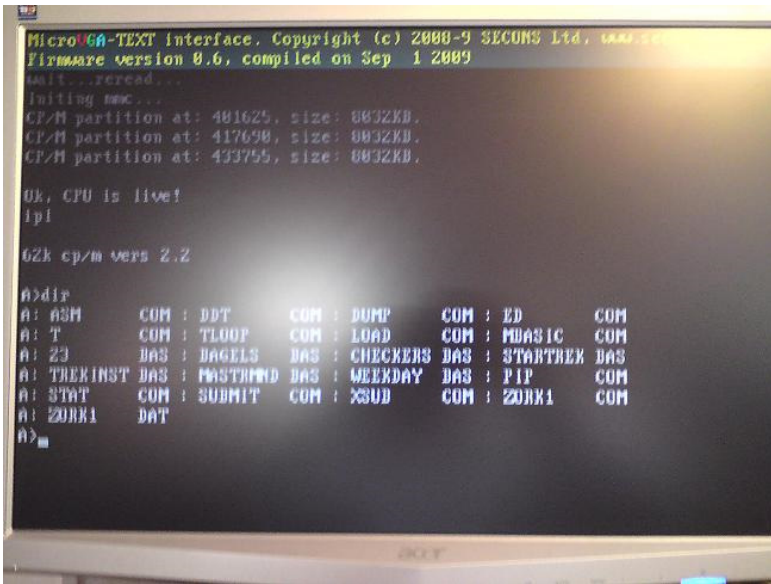
```
A>dir
A: ASM      COM : DDT      COM : DUMP    COM : ED      COM
A: T        COM : TLOOP    COM : LOAD    COM : MBASIC   COM
A: 23       BAS : BAGELS   BAS : CHECKERS BAS : STARTREK BAS
A: TREKINST BAS : MASTRMND BAS : WEEKDAY  BAS : PIP      COM
A: STAT     COM : SUBMIT   COM : XSUB    COM : ZORK1    COM
A: ZORK1    DAT

A>dir b:
B: TTT2     COM : SARGON   COM : OTHELLO  COM : MMIND    COM
B: ED       COM : LOAD     COM : STONE    COM : PIP      COM
B: STAT     COM : SUBMIT   COM : XSUB    COM : LADDER   COM
B: LADDER   DAT : LADCONF  COM : CURON   COM : PACMAN   COM

A>dir c:
C: C        CCC : C        SUB : CC2     COM : CC      COM
C: CLIB     COM : CLINK    COM : DEFF2    CRL : DEFF    CRL
C: BDS      LIB : STDIO    H : TTT2     C : MMIND    C
C: OTHELLO  C : STONE     C : ED       COM

A>_
```

Hier ein paar Bilder der Platine mit MicroVGA als Terminal:



Erreichte Geschwindigkeiten durch Optimierungen und Uebertakten:

		Peters	Leos		
		Schleife	Schleife		
AVR	MHz	Zeit [s]	Zeit [s]	"Speed"	HW SW
88	20	35.999			original hardware
88	20	21.380			4 (unoptimized code)
168	24	17.620			4 (unoptimized code)
88	25	16.871			4 (unoptimized code)
8	20	12.180			4 0
8	20	12.453	3.488	827 KHz	4 1
8	20	11.500	3.488	827 KHz	4 3
88	30	7.573			4 ?
8	20	7.599	2.039	1380 KHz	8 2
8	20	6.680	2.039	1380 KHz	8 3
8	20	6.418	1.958	1470 KHz	8 4
*** Hochrechnung ***					
88	40	5.6		~1600 KHz	4 ?
X8	30	4.3		~1800 KHz	8
X8	40	3.25		~2000 KHz	8

"HW 4" ist die jetzt aktuelle 4-bit Verdrahtung und "HW 8" die bald neu kommende 8-Bit Dram-Access Variante.

Update: 25.03.2011

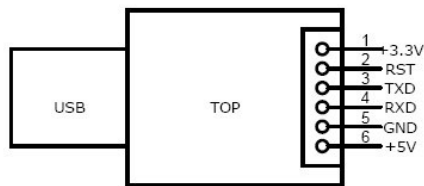
Ich habe bei ebay einen USB-TTL Adapter gesehen, der nicht nur eine serielle TTL/3V Schnittstelle über USB enthält, sondern auch 3,3V und 5V zur Versorgung bereitstellt. Ideal für dieses Projekt.

Bild des Adapters:

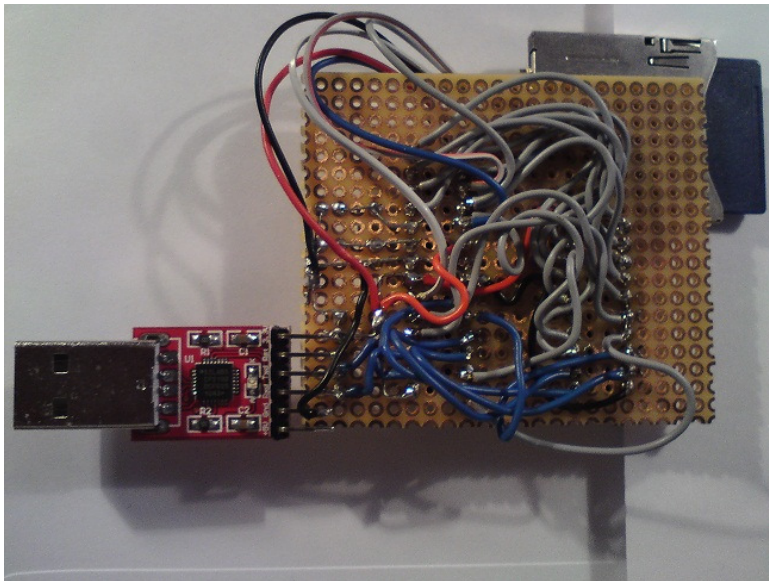
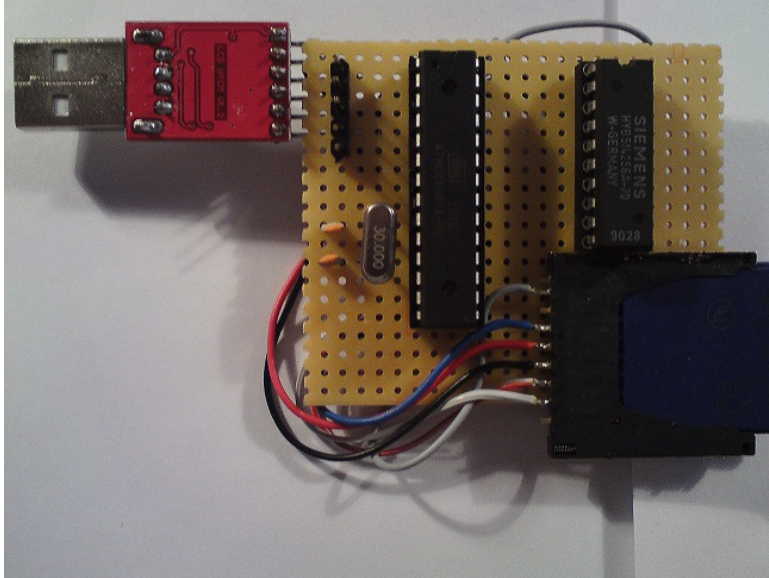


Pinout:

- Built-in USB to RS232 Transfer chip.
- Designed to be used for USB to TTL electronic projects.
- TTL interface output, easy to connect to your MCU.
- Status LED.
- Dual 3.3V and 5V Power output, work with 3.3v and 5v target device.
- Compact design.
- Dimension: 38 mm x 16 mm



Lochrasteraufbau mit Adapter:



Update: 11.04.2011 - Achtung neue Tools; BIOS inkompatibel zu vorherigen Versionen!

FAT16 Unterstützung: **avrcpm_fat16.zip**

Damit entfällt das lästige Partitionieren und füllen des Partitionen mittels dd.
Die Diskimages werden einfach ins Wurzelverzeichnis eines mit FAT16 formatierten SD Karte in der Form:
CPMDSK_A.DSK
kopiert. _A ist der Laufwerksbuchstabe: A,B,C,D...
Im ZIP Archive ist der Quelltext für den AVR. Hex Datei für m168, 30MHz, 4-bit Dram.
Da in den Diskimages das Bios, Bdos, CCP (CP/M System 2.2) schon enthalten ist, reicht es den AVR je nach AVR Typ und Quarz passen zu flashen und mind. ein A Diskimage auf die SD Karte zu kopieren.
Partitionen werden nach wie vor unterstützt.
Die Disk Images müssen mind. so groß wie in diskdefs definiert sein, sonst gibt es beim schreiben Probleme, weil der Treiber die Datei nicht vergrößert!
Das wird in make_disk mittels dd ibs=256k conv=sync sicher gestellt.

Aliados wird genutzt, um das Bios+ipl zu assemblieren (CP/M Emulation die M80+L80 nutzen). Außerdem enthält das Verzeichnis auch die cpmttools um die Diskimages zu bilden (make_disk.bat)

avrcpm_aliados.zip

BIOS assemblieren:

Im aliados Verzeichnis ist bios.asm+ipl.asm, m80+l80 zum assemblieren und make_bios.bat zum übersetzen von ipl+bios:
aliados m80.com =bios.mac
aliados l80.com bios.rel,bios.bin/n/e
aliados m80.com =ipl.mac
aliados l80.com ipl.rel,ipl.bin/n/e

CPM.BIN bilden:


```
make_cpm.bat zum bilden von cpm.bin (nutzt dd.exe, was auch enthalten ist)
dd conv=sync bs=128 count=1 if=iopl.bin > cpm.bin
dd conv=sync bs=128 count=44 if=CPM.SYS >> cpm.bin
dd conv=sync bs=128 count=7 if=bios.bin >> cpm.bin
```

Erstellen von Diskimages:

Das Resultat cpm.bin wird dann zum bilden von diskimages mit den cpmttools verwendet:
mkfs.cpm.exe -f avrcpm -b cpm.bin -L test CPMDISK_A.IMG

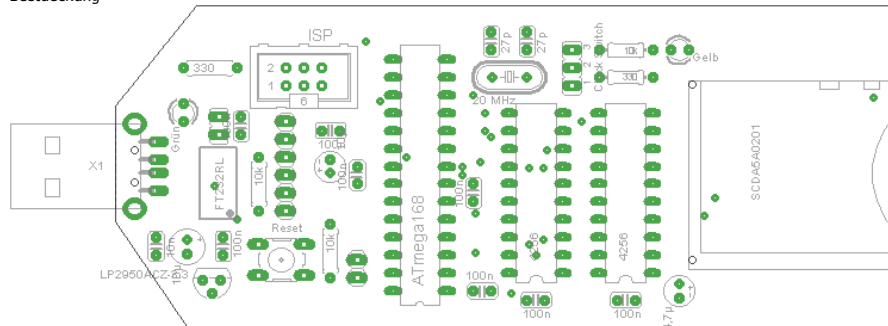
und füllen mit z.B:

```
cpmcp -f avrcpm CPMDISK_A.IMG cpmdsk/MBASIC.COM 0:MBASIC.COM
```

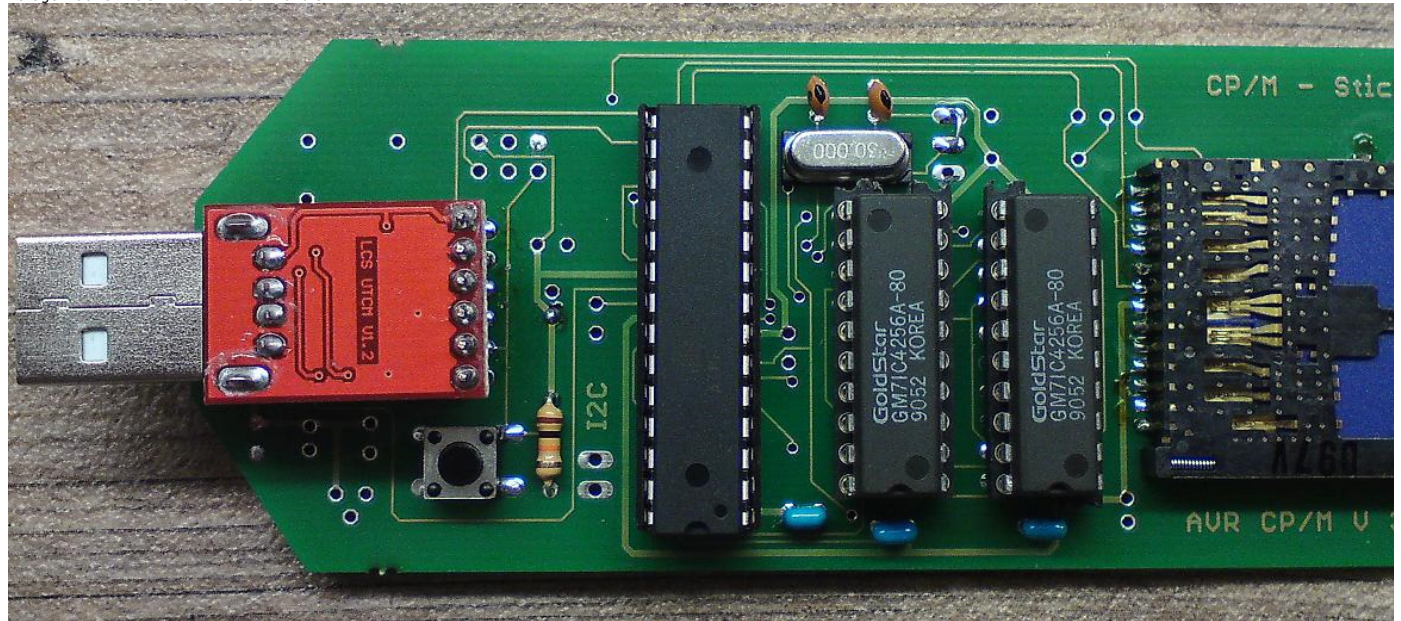
Neue Platinenversion: CP/M Stick

Eagle Dateien: [avr-cpm-stick.zip](#)

Bestueckung



Fertige Platine mit CP2102 TTL-USB Wandler



Einschaltmeldung mit 'Benchmark-Loop' (4-bit = 5,293)

```
Testing RAM: fill...wait...reread...
Initing mmc...
```

```
A:CP/M partition at: 445544, size: 7808KB.
B:CP/M partition at: 461160, size: 7808KB.
C:CP/M partition at: 476776, size: 8296KB.
D:FAT16 File-Image at: 066, size: 256KB.
E:FAT16 File-Image at: 130, size: 8192KB.
Partinit done.
Ok, CPU is live!
```

```
iopl
62k cp/m vers 2.2
```

```
A>dir
A: ASM      COM : DDT      COM : DUMP    COM : ED      COM
A: I        COM : TLOOP   COM : LOAD    COM : MBASIC  COM
A: 23       BAS : BAGELS  BAS : CHECKERS BAS : STARTREK BAS
A: TREKINST BAS : MASTRMND BAS : WEEKDAY  BAS : PIP    COM
A: STAT     COM : SUBMIT  COM : XSUB    COM : ZORK1   COM
A: ZORK1    DAT : SURVEY  COM
A>t b
```

```
Timer running. Elapsed: 002.623s.
A>
```