

Thread-ID: 177481

Die Entwicklung eines Community Projektes

Zusammengestellt: 2012 Peter.Sieg1@gmx.de

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

## Inhalt

<b>CP/M auf ATmega88</b> .....	4
Der Anfang.....	4
Ein erster Platinenentwurf .....	13
Erste Optimierungen .....	13
Erste Software .....	15
Erste Sammelbestellung.....	17
Layout 1 von Joe G. ....	19
Erste Schritte zum Parallelprojekt: AX81 .....	24
Gamesdsk+cpmtools .....	30
Eagle3D Bild der ersten Platine .....	30
Der Entwickler meldet sich zu Wort.....	31
Erste Analysen des Quellcodes.....	31
Leo C. steigt in die Entwicklung ein.....	33
Der Artikel zum Projekt wird aufgesetzt .....	35
SVN Server .....	35
Weiter arbeiten an der Software .....	36
Bilder von Leo C. Testaufbau.....	38
Die erste Platine aus der Sammelbestellung ist fertig .....	40
Weiter arbeiten an der Software .....	40
Erste Benchmarks.....	46
SVN von Leo C. und Wunschliste.....	50
Benchmarking die 2te.....	56
Größere Diskformate/Partitionen.....	70
Wordstar läuft .....	75
Der Entwickler möchte Leo C. und Joe G. im Source erwähnen .....	75
Weiter mit Partitionen .....	76
Update Wunschliste .....	80
VGA-Term .....	82
Wir gehen auf 8-bit Dram.....	85
BDS-C Compiler .....	91
Bilder von VGA-Term/uVGA .....	93
Layout 2 .....	94

Benchmark Übersicht .....	117
Erste Z80 Erweiterungen .....	117
Ladder Spiel .....	131
Anfänge der FAT16 Unterstützung.....	135
Platine Version 2.....	143
Weiter mit FAT16 .....	144
FAT16 fertig – VGA Teilprojekt .....	150
Simh hdsk Format für avrcpm .....	160
Weiter mit VGA-Teil .....	160
SVN Umstrukturierung .....	161
Weiter mit VGA-Teil - Tastatur .....	162
Bilder Lochraster 4-bit + USB Dongle .....	175
Das Projekt schläft etwas ein .....	177
Bios Assemblieren .....	181
AVR CP/M USB Stick .....	203
Weißer Platinen .....	210
Multiplan und Algol .....	212
VGA-Term mit P8X32A .....	213
Neue Platine V4.....	214
Endlich: Z80 Emulation – es geht weiter.....	217
Der Multiplan Fehler .....	224
Z80 – weiter geht's.....	232
Turbo Pascal 3 Image – und läuft!.....	243
Wir haben ZSDOS .....	258
Blaue Platinen.....	269
Fortran und PL/I .....	269
AltairSimH zum übersetzen des BIOS.....	270
MuMath.....	272

# CP/M auf ATmega88

## Der Anfang

Autor: Peter Sieg (Gast)

Datum: 12.05.2010 10:28

---

Hi.

Wollte mal auf diese Projekt aufmerksam machen:

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

Dort wurde ein [ATmega88](#), ein 256kx4 DRam-chip und eine SD Karte zusammen geschaltet, um über einen 8080 Emulator letztlich CP/M 2.2 von der SD Karte zu booten!

Ich dachte erst.. das kann doch nicht sein.. und hielt das für einen Aprilscherz.. aber er funktioniert wirklich! Siehe:

[http://avr.cwsurf.de/?AVR\\_CP%2FM](http://avr.cwsurf.de/?AVR_CP%2FM)

Es wäre wunderbar, wenn sich hierfür noch mehr Leute begeistern könnten und

1. Schauen, welche 8080 CP/M Programme jetzt auch schon laufen.. also ggf.

erweiterte diskimage's erstellen und testen.. BTW: MBASIC steigt mit einem

fehlenden Opcode aus.. ZORK läuft! Prima wären ein paar Spiele z.B Sargon

Schach etc...

2. Versuchen den [AVR](#) Assemblersource zu erweitern (mehr opcodes/Z80).

Peter

Autor: Reinhard Kern (Firma: RK elektronik GmbH) ([rk-elektronik](#))

Datum: 12.05.2010 16:00

---

Peter Sieg schrieb:

> Dort wurde ein [ATmega88](#), ein 256kx4 DRam-chip und eine SD Karte zusammen  
> geschaltet, um über einen 8080 Emulator letztlich CP/M 2.2 von der SD  
> Karte zu booten!

Hallo Peter,

du meinst doch sicher 2 Chips mit 4bit-RAM - einen Z80 auf einer 4bit-Architektur zu emulieren wäre schon Tierquälerei, obwohl natürlich im Prinzip alles geht, auch eine Turingmaschine als Z80.

Ich habe zwar tatsächlich noch Quasi-CP/M-Systeme im Einsatz (der Prozessor glaubt zumindest an CP/M), aber trotzdem die Frage, was soll's - es gibt genügend Z80 und CP/M Systeme auf dem PC, und schliesslich gibts ja den Z80 auch noch.

Gruss Reinhard

Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)

Datum: 12.05.2010 16:14

---

Reinhard Kern schrieb:

> Peter Sieg schrieb:

>> Dort wurde ein [ATmega88](#), ein 256kx4 DRam-chip und eine SD Karte zusammen  
>> geschaltet, um über einen 8080 Emulator letztlich CP/M 2.2 von der SD  
>> Karte zu booten!  
>  
> Hallo Peter,  
>  
> du meinst doch sicher 2 Chips mit 4bit-RAM - einen Z80 auf einer  
> 4bit-Architektur zu emulieren

Perverserweise hat er tatsächlich nur einen RAM Chip benutzt. Was soll ein Z80 auch mit 256kB anfangen :-)

(Kann den Link nicht direkt posten, Das (dot) gegen einen . austauschen.  
Ja der Link sieht ansonsten wirklich so seltsam aus  
<http://avr.cwsurf> (dot) de/?AVR\_CP%2FM

Alle Speicherzugriffe ins REM werden über Nibbles abgehandelt

Wie die Befehlsdekodierung und Ausführung gemacht wurde: Finde ich clever gelöst

Autor: Peter Sieg (Gast)

Datum: 12.05.2010 21:44

---

Jup. Nur 1 DRam a 256kx4. Ein Byte wird als 2xNibble geschrieben und wieder gelesen..

Selbst ich als Assembler Noob finde den Quelltext gut lesbar und strukturiert..

Warum? Weil es geht! Und weil es Spaß macht zu zeigen, das es geht..  
Und weil ein [ATmega88](#) ca. 3€ kostet, eine solches DRam gibts in der Schrottkiste (sind z.B beim Amiga 500 verwendet worden..) also alles für ca. 6-8€..

Ich persönlich finde die aktuelle Geschwindigkeit gar nicht soo schlecht!  
Läuft aktuell mit 20MHz. Für SwinSID sind die 88er aktuell sogar mit 32 und 40MHz übertaktet.. also noch eine Option.. ohne Hardware und (fast) Softwareanpassungen..

Für mich wäre die erste Priorität die Kompatibilität zu erhöhen, indem mehr Opcodes unterstützt werden.. Wäre doch was, wenn MBASIC und Sargon Chess darauf laufen würden (Sargon habe ich noch nicht getestet)..

BTW: Mit der seriellen Anbindung bin ich prima zufrieden.. ich möchte gar keine TV/BAS und PS/2 Anschluß.. (trotzdem ist [AVR](#) Chipbasic ein prima Projekt!!)

Nur CP/M ist hat schon was.. Turbo Pascal, Wordstar, dBase II.. ;-)

Peter

Autor: Reinhard Kern (Firma: RK elektronik GmbH) ([rk-elektronik](#))

Datum: 13.05.2010 14:45

---

[Karl heinz Buchegger schrieb:](#)

> Perverserweise hat er tatsächlich nur einen RAM Chip benutzt. Was soll  
> ein Z80 auch mit 256kB anfangen :-)

viel: program overlays habe ich ausgiebig benutzt und die kann man auch mit page switching einblenden statt nachzuladen, geht viel schneller. Und wenn man 4bit-Nibbles laden kann, dann kann man auch Daten aus einem MB-grossen RAM laden.

Eine Steuerungssoftware von mir hatte 30 Overlays und hätte daher auf 30 RAM-Pages a 32 kB laufen können.

Unterschätzt den Z80 nicht, da konnte man auch schon richtige Computer damit bauen.

Gruss Reinhard

**Autor: karadur (Gast)**

Datum: 13.05.2010 14:53

---

Hallo

es gab auch CP/M 3.0. Damit konnte man dank MMU mit einem Z80 sehr viel mehr Speicher verwalten. Ganz zu schweigen von CP/M86 und CP/M68k.

Und: die eZ80 sind auch nicht ohne.

PS: Auf meinem 8MHz Z80 CP/M war Turbopascal deutlich schneller als auf einem IBM XT.

Und: etwas was ich sehr vermisse. Dank Epromfloppy war der Rechner schneller gestartet als das die Bildröhre warm war.

**Autor: Rene B. ([themason](#))** 

Datum: 13.05.2010 17:00

---

Also nen Linux auf nem [AVR](#) emulieren dürfte denke ich scheitern. Aber das CP/M mit emulierter Z80 ... Schon echt ne geniale Sache.

@reinhard

>Unterschätzt den Z80 nicht, da konnte man auch schon richtige Computer  
>damit bauen.

Das ist wohl auch der Grund warum es so viele Computer auf Z80-Basis gab :-)

Mal im Ernst. Immerhin hat die ganze Computerei ja mit dem 4004 angefangen (wenn man man den Zuse, IBM, Leibnitz und Babbage außen vor lässt :-)

Und nen Z80 auf nem [AVR](#) emulieren, und "nebenbei" noch das DRAM handlen ...

Respekt.

Wenn ich damals mal was mit CP/M gemacht hätte würd ich dieses Projekt auf jedenfall nachbauen. Retro mit neuen mitteln find ich immer witzig. Genauso wie den SwinSID. Schon echt klasse so Projekte. Weiter so :-)

**Autor: Frank K. ([fchk](#))**

Datum: 13.05.2010 19:41

---

Peter Sieg schrieb:

> Es wäre wunderbar, wenn sich hierfür noch mehr Leute begeistern könnten

Ihr habt doch ein Rad ab! Sorry, aber das musste ich einfach loswerden.

Ein EZ80F91AZA mit 50 MHz kostet bei Digikey etwa 10 Euro, bei RS etwa 7 Euro, hat 256k Flash eingebaut, hat Ethernet und noch einiges mehr an Peripherie (2 UART, i2c, SPI, RTC, 32 GPIOs, ...) und ist zu 100% software-kompatibel zum alten Z80. Du kannst sogar CP/M 3.0 Bankswitching machen, das Teil hat selber die entsprechende Logik dafür. Du brauchst nur noch RAM anzuschließen. Ein 512k SRAM kostet bei RS 5 Euro. Also auch nicht die Welt. /CS macht der Controller selber, Daten- und Adressbus und /RD und /WR werdet ihr auch wohl richtig anschließen können.

Was soll das also? Modernes Flagellantentum?

PS:

[http://www.zilog.com/index.php?option=com\\_product&...](http://www.zilog.com/index.php?option=com_product&...)

fchk

Autor: A. K. ([prx](#))

Datum: 13.05.2010 19:48

---

Frank K. schrieb:

> Ihr habt doch ein Rad ab!

Eindeutig ja. Und?

> Ein EZ80F91AZA mit 50 MHz kostet bei Digikey etwa 10 Euro, bei RS etwa 7  
> Euro, hat 256k Flash eingebaut, hat Ethernet und noch einiges mehr

Meinst du wirklich, es ginge hier darum, auf möglichst effiziente Weise Z80-Programme laufen zu lassen?

> Was soll das also? Modernes Flagellantentum?

Noch nie jemanden an einem 50 Jahre alten Motorrad basteln gesehen? Das macht der bestimmt nicht um von A nach B zu kommen.

Autor: Stefan (Gast)

Datum: 13.05.2010 20:13

---

>> Es wäre wunderbar, wenn sich hierfür noch mehr Leute begeistern könnten  
>Ihr habt doch ein Rad ab! Sorry, aber das musste ich einfach loswerden.  
>Ein EZ80F91AZA mit 50 MHz kostet bei Digikey etwa 10 Euro, bei RS etwa 7  
>.....

Du bist zu eingefahren: Du gehst von falschen Anforderungen aus. Du gehst davon aus, dass jedes Problem haufenweise Rechenpower, Speicher und viel Peripherie benötigt. Wer sagt, denn, dass es so sein muss?

Wenn das System nicht schnell sein muss, kommst du preislich mit den von Dir aufgezählten Bauteilen nicht hin. Außerdem muss Du für Deine Bauteile eine Platine fertigen lassen. Das geht mit dem [Atmega88](#) auch

ohne. In Ingenieursstunden gerechnet, wäre Deine Lösung mit Sicherheit überteuert.

**Autor:** karadur (Gast)

**Datum:** 13.05.2010 20:50

---

Hallo

nu übertreibt es nicht! Es gibt Dinge die man einfach aus Interesse oder Spass macht. Die Idee mit einem [AVR](#) einen 8080 zu emulieren ist ja toll.

**Autor:** Frank K. ([fchk](#))

**Datum:** 13.05.2010 21:06

---

[Stefan schrieb:](#)

> Wenn das System nicht schnell sein muss, kommst du preislich mit den von  
> Dir aufgezählten Bauteilen nicht hin. Außerdem muss Du für Deine  
> Bauteile eine Platine fertigen lassen. Das geht mit dem [Atmega88](#) auch  
> ohne. In Ingenieursstunden gerechnet, wäre Deine Lösung mit Sicherheit  
> überteuert.

Eben nicht. Der wesentliche Aufwand, nämlich die CPU-Emulation entfällt. Heißt also: Ich muss nur das CP/M BIOS anpassen, aber das müsst Ihr auch. Das BDOS ist ja überall gleich und wird ja als Objektfile dazugelinkt. Und selbst ein 3.0'er BIOS hat nur verhältnismäßig wenige Funktionen. Hab ich anno 87 schon mal gemacht, als ich noch jung und hybsch war.

Und zum Bauteileaufwand: Gut, der Chip ist ein 144'er LQFP. Also Adapterplatine, damits aufn Lochraster paßt. Das ist auch die einzige Hürde. RAM im DIL-Gehäuse, 3.3V Vcc, kann man direkt anklemmen, ist kein Hexenwerk. Ein Quarz oder Oszillator (braucht Ihr auch), Spannungsregler (braucht Ihr auch), RS232-Pegelwandler (braucht Ihr auch), und eine Handvoll R's und C's zum abblocken und als Lastkapazität für den Quarz und als Pullups.

Mehr braucht der EZ80F91 nicht, um hochzukommen.

RTC gewünscht? OK, noch ein 32 kHz Uhrenquarz plus 2 C's und 1 R dazu. SD-Karte? OK, ab an den SPI-Port. Kennt Ihr ja, ist hier auch nicht anders.

Wenn man kein Ethernet braucht, dann wars das. Wirklich! Ich habe Eval-Board, Prozessormodule und ZDI(Zilog 2-Draht [JTAG](#) Variante)-Debugger und die ganze Doku und Software da, ich weiss, wovon ich rede.

fchk

**Autor:** Rene B. ([themason](#)) 

**Datum:** 13.05.2010 21:53

---

@Frank

>Ihr habt doch ein Rad ab!

Fällt dir aber früh auf ...

>Ein EZ80F91AZA mit 50 MHz kostet bei Digikey etwa 10 Euro, ...

Natürlich gibts Chips/Boards die mehr können weniger kosten, aber es besteht doch gerade der Reiz darin mit den heutigen Standardchips solche Sachen zu machen wie z.b. eine Z80, einen SID, eine [6502](#) zu emulieren, oder eben ein BAS-Signal für nen Fernseher zu erzeugen.

Ich bin ja auch total Plemplem wenn ich mit einem [FPGA](#) Audio mache obwohl jeder DSP für die Hälfte an Euros das doppelte an MIPS/MFLOPS hat.

Trotzdem macht es Spaß, man lernt was dabei und man freut sich nachher. (Selbst wenns im Falle des Audio-FPGAs die Freude hauptsächlich bei mit selbst liegt, aber das ist es mir wert)

Ich glaube man nennt sowas : Meet the Challenge ... Oder anders gesagt : Wenns Spaß macht ...

@A.K.

Autor: 900ss D. ([900ss](#))

Datum: 13.05.2010 22:59

---

[Frank K. schrieb:](#)

> Ihr habt doch ein Rad ab!

Ich freue mich über diese losen Räder. Ehrlich! Schade dass du sie nicht hast.

[Frank K. schrieb:](#)

> ich weiss, wovon ich rede.

Mag sein, aber leider weisst nicht du nicht, wovon hier geredet wird. Nicht von einer aktuellen Technik, sondern dem Schwelgen in Vergangenheit und deren Technik. Fertig kaufen kann man alles. Dann wäre jedes Hobby, wo selbst gebaut wird dummes Zeug. Alles kann man billiger und fertig kaufen. #gähn#

Das Z80-Zeugs was du erwähnst liegt hier auch rum. Weiss also was du meinst. Ist aber nicht das, wovon hier gesprochen wird.

Autor: Christian Berger ([casandro](#)) 

Datum: 14.05.2010 07:17

---

[Frank K. schrieb:](#)

> Eben nicht. Der wesentliche Aufwand, nämlich die CPU-Emulation entfällt.

Die CPU-Emulation ist reine Software, das muss somit nur einmal gemacht werden.

Natürlich stimme ich Dir aber zu, dass wir eine Schraube locker haben. Für uns ist Elektrotechnik ein wichtiger Teil in unserem Leben, und willkommener Ausgleich zum langweiligen Beruf.

Autor: 900ss D. ([900ss](#))

Datum: 14.05.2010 09:37

---

[karadur schrieb:](#)

> Habe selbst

> unter CP/M meine Diplomarbeit

Was ich vergass, hab ich übrigens auch ;-)  
Es war allerdings ein geklontes CP/M und hiess ZDOS und lief auf einem Z-80-System auf ECB-Bus Basis von Fa. Janich & Klass. Die Firma gibt es sogar noch, wie ich gerade festgestellt habe. Und das System hatte zwei 8"-Lauferke. :-)

**Autor: Hauke Sattler ([hauke](#))**

**Datum: 14.05.2010 10:41**

---

@Peter Sieg

Sorry, aber es geht hier nicht um das Einfachste oder Schnellste.  
Es geht um die Herausforderung. Sicher es wäre einfacher auf dem PC einen fertigen Z80 Emulator anzuwerfen und gut. Aber was hat man davon? Es ist für mich halt interessanter eine Rechnerarchitektur auf einer anderen abzubilden. In diesem Falle lernt man nämlich über beide etwas.

Genauso ist es vermutlich bei den Cracks, die heutzutage noch Demos für den C64 machen (teilweise mit Effekten die man vor 15 Jahren nicht einmal einem PC zugetraut hätte) Wenn man dort sagen würde: "wieso nehmt ihr nicht einen PC mit 3d Karte, der ist doch viel schneller"  
Dann würde man vermutlich aus dem Saal geworfen werden.

P.S.

Zur Zeit code ich an einer [6502](#) Emu (mit RAM, ROM, Dezimalmodus und illegalen Opcodes) und kämpfe mit teilweise widersprüchlichen Dokus.

**Autor: Yalu X. ([yalu](#)) (Moderator)**

**Datum: 14.05.2010 11:45**

---

Frank K. schrieb:

> Ihr habt doch ein Rad ab!

Auch wenn ich das alles nicht so eng sehe, kann ich Franks Worte doch ein Bisschen verstehen. Warum?

Grundsätzlich bin auch ich ein Freund von solchen "Radabprojekten" (im Folgenden RAPs genannt). Der entscheidende Unterschied zwischen einer industriellen Produktentwicklung und einem RAP liegt in meinen Augen im Folgenden:

### **Industriell**

Bei einer industriellen Produktentwicklung wird versucht, sehr viele verschiedene Kriterien wie

- Entwicklungsaufwand
- Materialkosten
- Fertigungskosten
- benötigte Fertigungseinrichtungen
- Baugröße
- Energieverbrauch
- seltener auch "Restro"-Aspekt (Beschränkung auf Materialien und/oder Aussehen und/oder Fertigungsverfahren aus der Vergangenheit)
- ...

sinnvoll zu gewichten und das Produkt so zu optimieren, dass die Kriterien in Summe möglichst gut erfüllt sind und somit mit dem Produkt ein möglichst hoher Gewinn erzielt wird.

## RAP

Bei einem RAP wird typischerweise eine radikal andere Gewichtung dieser Kriterien vorgenommen und versucht, ein "Produkt" zu entwickeln, das nach *dieser* Gewichtung optimal ist. Im häufig auftretenden Extremfall wird nur ein einzelnes dieser Kriterien mit einem Gewicht >1 versehen, während alle anderen mit 0 gewichtet sind, also völlig unberücksichtigt bleiben.

Gerade dieser selbstdefinierte Bewertungsmaßstab frei von den Vorgaben irgendwelcher Marketingabteilungen und anderen externen Zwängen macht den Reiz von RAPs aus.

Eine Gewinnerzielungsabsicht besteht bei einem RAP üblicherweise nicht. Stattdessen steht der Coolness-Faktor im Vordergrund, der meist darin besteht, sagen zu können: "Meine Entwicklung stellt hinsichtlich des Kriteriums x alles, was es zu kaufen gibt, in den Schatten."

Beispiele:

"Produkt"	Gewichtung > 0	Gewichtung = 0
Web-Server im SO-6-Gehäuse	Baugröße Materialkosten	Speicherkapazität Geschwindigkeit
Soft-USB	Baugröße Materialkosten	Geschwindigkeit Entwicklungsaufwand
Oszi aus Schrottteilen (alter Fernseher usw.)	Materialkosten	Bandbreite Messgenauigkeit Entwicklungsaufwand
Relaisrechner	Restro-Aspekt	Geschwindigkeit Speicherkapazität Entwicklungsaufwand

Bei dem CP/M auf dem ATmega fehlt nach meinem Geschmack ein wenig dieser Optimierungsgedanke, da der Einsatz des ATmega im Vergleich zu einem Z80-kompatiblen Mikrocontroller bzgl. der o.g. Kriterien kaum Vorteile bringt. Vielleicht habe ich aber auch nur ein wichtiges Kriterium vergessen ;-)

Ok, es gibt noch einen weiteren Aspekt, weswegen man ein Hobbyprojekt beginnt, nämlich der Lerneffekt bzw. Erkenntnisgewinn. Nur hätte ich Hemmungen, dafür so viel Zeit in ein Projekt zu stecken, wo ich vielleicht ähnlich viel in einem anderen Projekt lernen könnte, das weniger zeitaufwendig ist und/oder ein nützlicheres Ergebnis liefert.

Aber die persönlichen Zielsetzungen sind eben verschieden. Deswegen läge mir nichts ferner, als jemandem dieses Projekt auszureden. Auf jeden Fall wünsche ich allen, die sich damit beschäftigen, viel Erfolg!

**Autor: Peter Sieg (Gast)**

**Datum: 14.05.2010 17:30**

---

Na schön, das hier doch einiges an Interesse ist.. ;-)

(Auch - oder gerade weil wir ein Rad abhaben)

Oder wie schon im deutschen Liedgut steht, wer nicht verrückt ist, der ist

nicht normal..

---

Nun jeder ist ja relativ frei, das Projekt zu erweitern.. ist ja GPL3..  
Für mich persönlich, ist die serielle+VT100 Anbindung völlig ok..  
PS/2+BAS  
sind mir nicht so wichtig.. wenn es aber doch jemand macht. prima!

Wichtiger wäre zuerst einmal ein Ausbau der unterstützen Opcodes..

Wenn da mal auch mehr/Z80 Programme laufen etc.. mache ich auch gerne  
ein  
Platinen-Layout mit Eagle ;-)) Oder ganz abgefahren in SMD mit Micro-SD  
in  
Streichholzschachtelgröße. ;-))

Peter

**Autor: Sigint 112 ([sigint](#))**  
**Datum: 21.05.2010 18:10**

---

Ich wollte knur kurz den Thread wieder nach oben bringen. Nebenbei finde ich das ein wirklich schönes Projekt, was ich bei nächster Gelegenheit testen und erweitern möchte. Wenn man einen zweiten [AVR](#) für Tastatur und Video-Output nimmt, dann hat man ein schönes System.

Gruß,  
SIGINT

**Autor: Peter Sieg (Gast)**  
**Datum: 25.05.2010 11:09**

---

@SIGNINT: Prima! Inzwischen kann ich das diskimage unter win32 ändern..  
Habe auch ein othello gefunden, das läuft. Und ein mbasic für 8080.  
Läuft zwar, aber Sprungziele werden beim load zu 00000??

Wird der z80.asm unter [AVR](#) Studio assembliert?  
Wollte mal versuchen, von 20 auf 32MHz umzustellen.. dazu muß ich sicher die  
Geschwindigkeit der RS232 Schnittstelle umstellen..?

Peter

**Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)**  
**Datum: 25.05.2010 11:13**

---

[Peter Sieg schrieb:](#)

> @SIGNINT: Prima! Inzwischen kann ich das diskimage unter win32 ändern..  
> Habe auch ein othello gefunden, das läuft. Und ein mbasic für 8080.  
> Läuft zwar, aber Sprungziele werden beim load zu 00000??

Wenn du diese Emulation benutzt:  
Es sind noch nicht alle Opcodes des Z80 unterstützt.

Speziell die CB und ED Schiene ist noch gar nicht vorhanden.  
Und ob bei den implementierten Opcodes die Status Flags überall richtig nachgezogen werden, hab ich nicht näher untersucht.

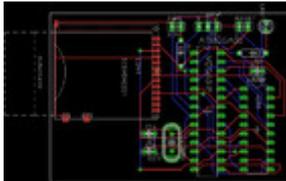
# Ein erster Platinentwurf

Autor: Peter Sieg (Gast)

Datum: 26.05.2010 15:52

Angehängte Dateien:

- [avrcpm\\_eagle.zip](#) (41 KB, 59 Downloads)



- [avrcpm\\_eagle.png](#)  
8,2 KB, 614 Downloads

---

Jup. Bin leider auch kein 8080/Z80 Kenner.. da müßte ich mich erst einarbeiten.. nun, zumindest kann ich den Quelltext mit [AVR Studio 4](#) assemblieren (dabei auch gleich die Baudrate kalkulieren lassen..) und das Ergebnis läuft..

Werde das mal mit [ATmega88PA](#) und 32MHz anstatt 20MHz probieren.. dabei sollte auch der DRam Refresh weiter runter können.. mal schauen, wie viel 'Geschwindigkeit' das bringt..

Dann wäre da auch noch die Frage, warum das Nibble an D0-D3 nicht auch ein Nibble an einem Port ist..

Nun Ja.. zu allem 'Überfluß' hier auch noch ein Platinenlayout ;-)

Peter

## Erste Optimierungen

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 28.05.2010 14:18

---

Ich habe mir den Code mal angesehen und denke, dass die DRAM-Anbindung zwar recht minimalistisch, dafür aber auch arg langsam ist. Grob geschätzt komme ich z.B. auf über 160 Takte bei der dram\_read Routine. Wenn man einen 40-poligen Controller (z.B. Meag16) und zwei DRAMs nimmt, D0-D7 an Port A, A0-A7 an Port C und die Steuersignale an Port D, sollte das wesentlich schneller gehen. Letztendlich müssen nur die entsprechenden Routinen ausgetauscht werden. Ein Beispiel wäre:

```
dram_read:      cli                ;1 disable interrupt
                 ldi         ZL, 0x7F          ;1 RAS
                 ldi         ZH, 0x1F          ;1 RAS + CAS + OE
                 out         PORTC, adrh       ;1
                 out         PORTD, ZL        ;1 RAS active
```

```

out    PORTC,adr1    ;1
out    PORTD,ZH     ;1 RAS + CAS + OE active
ldi    ZL,0x9F      ;1 RAS inactive
out    PORTD,ZL     ;1 CAS + OE (hidden refresh)
in     temp,PINA    ;1 get data
out    PORTD,ZH     ;1 RAS + CAS + OE
ldi    ZH,0xFF      ;1
out    PORTD,ZL     ;1 only RAS
out    PORTD,ZH     ;1 inhibit
sei    ;1 enable interrupt
ret    ;4 end

```

Bringt Beschleunigung um etwa Faktor 7 beim Lesen aus dem Speicher.

Jörg

**Autor: Peter Sieg (Gast)**

**Datum: 29.05.2010 09:55**

Hi Jörg.

Das wäre super!

Da meine Kenntnisse in [AVR](#) Assembler eigentlich nicht vorhanden sind ;-)  
 bleibe ich vorerst einmal beim aktuellen Aufbau. Hier möchte ich noch  
 mal

die Takterhöhung auf 32MHz und die Reduzierung des DRam-Refreshs  
 probieren.

Und der Warmstart muß auch überarbeitet werden (hängt einfach).

Inzwischen kann ich diskimage unter win32 mit den cpmtools ändern und  
 mit BDS-C übersetzte Programme laufen auch unter avr-cpm ;-)

Zu einem anderen Aufbau sehe ich das aber letztlich genauso.. was  
 ein 'fauler' Kompromiss ist und viel Rechenpower kostet muß durch  
 etwas besseres ersetzt werden.. Da kann man beim Ram auch z.B über SRAM  
 nachdenken.. dann braucht man keinen Refresh mehr..

Wäre super, wenn hier genügend zusammen kommen, um daraus einen kleinen,  
 schnuckeligen und brauchbaren CP/M Rechner zu basteln..

Peter

**Autor: Ronny Minow ([hobby-coder](#))**

**Datum: 29.05.2010 10:28**

@ Peter Sieg: Ich habe mir Dein Projekt auf Deiner Homepage angesehen.  
 Als erstes: Das ist ein sehr interessantes Projekt. Nicht zuletzt  
 deshalb, weil mir CP/M eigentlich unbekannt ist. Das soll mich aber  
 nicht daran Hindern, Dein Projekt aufzubauen und mit zu Entwickeln.

Was mir für den Aufbau fehlt, ist ein Z80 Assembler. Wenn ich das Paket  
 zum erstellen der Image Disk richtig verstehe, brauche ich diesem um die  
 Software entsprechend übersetzen zu können. Wo bekomme ich diesem  
 Assembler her...?

@Joerg Wolfram: Du hattest einmal einen einfachen Basic Computer mit  
 einem [Atmega8/88](#) aufgebaut. Könnte man die Umgebung für den TV Out/  
 FBAS, sowie den PS2 Controller an den CP/M Rechner anbringen? Falls  
 notwendig, über einen 2. [Atmega8](#)...

**Autor: kruemeltee (Gast)**

**Datum: 29.05.2010 11:07**

Sowas gibt es auch mit dem Propeller:

<http://forums.parallax.com/forums/default.aspx?f=2...>

TV, VGA, PS/2, SD-Karte, ... mit CPM2.2 oder MPM

Der Minimal-Aufbau ist für unter 30 Euro zu machen, wenn man **alle** Teile kaufen muss.

Das einfachste Board, wäre ein abgespecktes Dracbalde (nur RAM, PS/2 und VGA/TV, SD, keine weiteren IOs usw.)

Autor: Ronny Minow ([hobby-coder](#))

Datum: 29.05.2010 13:11

---

[kruemeltee schrieb:](#)

> Sowas gibt es auch mit dem Propeller:

>

> <http://forums.parallax.com/forums/default.aspx?f=2...>

>

> TV, VGA, PS/2, SD-Karte, ... mit CPM2.2 oder MPM

>

> Der Minimal-Aufbau ist für unter 30 Euro zu machen, wenn man \*alle\*

> Teile kaufen muss.

>

> Das einfachste Board, wäre ein abgespecktes Dracbalde (nur RAM, PS/2 und

> VGA/TV, SD, keine weiteren IOs usw.)

Auch, wenn der Propeller µC schon klasse ist, so denke ich sollten wir den Atmega µC verwenden.

Ich selbst habe ein Hive -Retrosytle Computer, der 3 Propeller µC's verwendet. Der Hive ist ein Klasse Computer. Man kann sicher auch CP/M drauf laufen lassen, was auch in Entwicklung ist. Hier geht es aber um CP/M für den Atmega, nicht für den Propeller µC.

---

@Peter Sieg: Welche Software hast Du schon auf dem Projekt zum leben erwecken können?

## Erste Software

Autor: Peter Sieg (Gast)

Datum: 29.05.2010 17:21

---

@Ronny Minor: Habe ein paar Spiele, die laufen: Mastermind, TicTacToe, Othello, Stone.. kommt ggf. noch Packman mit VT52 Steuerung dazu.. evtl. noch 4-gewinnt..

Peter

Autor: Ronny Minow ([hobby-coder](#))

Datum: 29.05.2010 17:46

---

[Peter Sieg schrieb:](#)

> @Ronny Minor: Habe ein paar Spiele, die laufen: Mastermind, TicTacToe,

> Othello, Stone.. kommt ggf. noch Packman mit VT52 Steuerung dazu.. evtl.  
> noch 4-gewinnt..  
>  
> Peter

Einige Spiele sind schon einmal ein Anfang. Wie sieht es mit typischer CP/M Software aus? Läuft z.B. DBase (gab es das für CP/M überhaupt...?)  
?

---

Wer hat diesem Computer schon aufgebaut? Ach ja, gibt es schon Ideen für ein Video Ausgang? Oder anders: könnte man an Herrn Wolframs Basic Computer mit [Atmega8](#)/88 den DRAM drann hängen und somit eine Symbiose aus beiden Computern machen?

**Autor: Peter Sieg (Gast)**

**Datum: 29.05.2010 21:47**

---

@Ronny Minow: dbase gibt es für CP/M..  
Generell gilt vorerst: NUR 8080 CPU - d.h Programme, die unter CP/M-80 laufen haben gute Chancen..

Aber: Die Emulation hat SICHER noch Fehler! Z.B MBASIC4.5 für CPM-80 läuft ersteinmal.. Interaktiv, also z.B print 2+4 (=6) geht.. aber Programmlauf geht nicht, weil alle Variablen zu 00000 evaluieren.. keine Ahnung warum!  
Im AltairSIMH Emulator mit Z80 CPU ist alles ok..

Die vielen anderen Basic Versionen habe ich noch nicht getestet..

Es gibt generell sehr viel alte Software für CP/M-80..  
<http://www.z80.eu/>  
<http://www.retroarchive.org/cpm/>

Deswegen wäre/ist meine Prämisse:

1. Noch weitere Basic Interpreter testen
2. Noch eine paar weitere Spiele compilieren mit BDS-C
3. Warmstart implementieren (hängt da aktuell einfach)
4. Das ganze mal mit 32MHz laufen lassen (88PA)
5. 8080 Code prüfen und ggf. Fehler beseitigen

---

6. Evtl. andere Hardware mit 8-bit SRam etc. etc
7. Z80 Opcodes implementieren

Generell denke ich reicht es mir, wenn ich 1-5 erreicht habe..  
bin aber allen Neuentwicklungen/Verbesserungen offen gegenüber..  
kann man/das Projekt ja nur von profitieren..

---

Bezüglich des benutzten z80asm.. keine Ahnung, welcher das ist.  
Da sollte man den Entwickler: Sprite\_TM fragen.. ich würde den gerne auch unter win32 haben.. bisher ändere ich das vorhandenen diskimage mit cpmrm und cpmcp..

Peter

**Autor: Ronny Minow ([hobby-coder](#))**

**Datum: 30.05.2010 07:05**

---

@Peter Sieg: Danke für die Links. Ich habe mich da mal umgesehen. Wenn ich dazu komme, werde ich mal "Tiny C", "Pascal" und einen Assembler auf dem Computer laufen lassen. Aber, ersteinmal muss ich den zusammen löten. Die Bauteile hierfür dürften in einigen Tagen bei mir sein.

Ach ja, thx für das Pin Layout des SRam's, das Du auf Deiner Homepage veröffentlicht hast.

## Erste Sammelbestellung

Autor: Peter Sieg (Gast)

Datum: 30.05.2010 12:39

---

Ich weiß, das ist verrückt ;-)  
Ich denke ernsthaft darüber nach 10 Platinen anfertigen zu lassen ;-)  
Aus dem forum64 gibts zumindest schon einmal noch 2 Interessenten außer mir..  
da ich auch mind. 2 für mich nehme wären es schon 4.. wenn wir noch ein paar  
finden könnte das wirklich klappen.. unter 10 macht es keinen Sinn..  
ich rechne mit ca. 9-10€/Platine. SD Slot kostet 3€. Versand ca. 1,50€ (Post). Natürlich kann man das auch auf LR aufbauen.. aber mir fertiger Platine wird es doch ein wenig einfacher (ist durchkontaktiert mit Lötstopp).  
Ach ja, Platine siehe weiter oben.. RS232 Wandler will ich nicht drauf haben.  
Und 3,3V Netzteile gibts bei Pollin..

Peter

Autor: Uhu Uhuhu ([uhu](#))

Datum: 30.05.2010 12:45

---

Da ist doch Platz für einen Regler drauf - dann könnte man auf vorhandene Wandwarzen zurückgreifen...

Autor: Sigint 112 ([sigint](#))

Datum: 30.05.2010 13:13

Angehängte Dateien:

- [z80\\_emu.zip](#) (183,6 KB, 93 Downloads)

---

@Peter: Lohnt es sich für so eine kleine Schaltung eine Platine fertigen zu lassen?!? Ich fände es besser erst ein vollständiges System, mit IO-Hardware (TV-Out, Keyboard, Sound?!, etc.) zu entwickeln und dann die Platinen fertigen zu lassen. Ich glaube, die meisten würden gerne einen kompletten CP/M Rechner haben, den man ohne externe Hardware betreiben kann.

Ich hab schon ein paar Ideen, was man mit dem System machen könnte:  
Portieren eines [AVR](#)-Assemblers auf CP/M-80  
CP/M-80 Software zum flashen von [AVRs](#)  
Grafik-LCD am System anbinden  
mehrer Diskettenimages auf SD-Karte

einfache Soundausgabe per PWM (DMA,IRQ?!)

Mir ist klar, daß dies noch viel Arbeit ist, aber ich Glaube es könnte sich lohnen.

Ich hab schonmal angefangen auf dem PC einen kleinen Z80-Emulator zu basteln, um mich in CP/M ein zu arbeiten. Leider funktioniert das aber noch nicht so, wie ich es möchte.

Gruß,  
SIGINT

**Autor: Peter Sieg (Gast)**  
**Datum: 30.05.2010 14:46**  
**Angehängte Dateien:**

- [AltairSIMH.zip](#) (1,7 MB, 121 Downloads)

---

@SIGINT: Nun, eine 10ner Auflage der aktuellen Hardware steht einer späteren Auflage eines erweiterten Systems ja nicht im Wege..  
(Ich bin nun mal Minimalist ;-)

Schau doch mal, ob Jörg Wolfram auch noch Interesse hat an einer Erweiterung zum vollwertigen CP/M Rechner mitzuarbeiten.. Evtl. finden sich hier weitere Mitstreiter.. kann man die Entwicklung ggf. aufteilen.. Wäre wirklich super, wenn das klappt!

Als Emulator schau mal bitte nach AltairSIMH. Läuft bei mir prima.. Hänge ich hier mal an.. einfach entpacken und Altairall.bat starten.  
Laufwerk A: ist CP/M 2.2  
Laufwerk B: ist MBasic  
Laufwerk C: ist BDS-C compiler  
Mit R/W lassen sich Dateien vom Hostsystem nach CP/M transferieren.

Peter

**Autor: Joerg Wolfram ([joergwolfram](#))**  
**Datum: 31.05.2010 13:00**

---

Es ist wahrscheinlich sinnvoller, eines der in der Codesammlung vorhandenen [AVR](#)-Terminals zu benutzen. Die haben eigentlich alle eine höhere Auflösung als das ChipBasic und die serielle Ansteuerung ist auch schon fertig.

Ich halte das Ganze für ein recht interessantes Projekt, allerdings geht für mich der praktische Nutzen gegen Null.

Wobei ich heutzutage auch kein Projekt mehr anfangen würde, welches nur TV als Ausgabemedium hat. Schon beim ChipBasic mit seiner doch recht groben Auflösung sind bei neueren TVs durch die ganze Samplinggeschichte die Pixel unterschiedlich breit und auch teilweise unterschiedlich hoch wahrscheinlich werden die Bilddaten 4:2:2 gespeichert).

Eine probeweise Darstellung von 80 Zeichen (480 Pixel bei 10MHz Pixeltakt) ist auf einem guten Analoggerät noch lesbar, aber schon auf einem 100Hz Fernseher einfach eine Zumutung (zumindest meinem Eindruck nach).

Jörg

# Layout 1 von Joe G.

Autor: Joe G. ([feinmechaniker](#))

Datum: 31.05.2010 13:07

Angehängte Dateien:

- [AVRCPM.zip](#) (80,7 KB, 74 Downloads)



- [AVR\\_CPM.png](#)  
11,9 KB, 675 Downloads

---

Hallo,

nettes Projekt, Danke Peter!

Ich habe mal Peters Idee aufgegriffen, seine Schaltung leicht erweitert und ein Layout dazu erstellt. Vielleicht findest es ja Gefallen.

1. Die Jumper TTL, Reset und 3.3 V behalten vollständig ihre Pinbelegung und Funktion
2. Die Kommunikation (RS232) ist auf eine USB-Schnittstelle gelegt.
3. Die Stromversorgung (5V und 3.3V) erfolgt über die USB-Schnittstelle
4. Steckverbinder zur ISP-Programmierung des [AVR](#) hinzugefügt.
5. Mit dem Jumper POWER Switch kann wahlweise zwischen 5V und 3.3V umgesteckt werden (Vorsicht CF-Card ziehen!)
6. Mit dem Jumper CLOCK Switch kann der [AVR](#) Takt über den USB Controller bezogen werden. Hier sind 6, 12, 24 oder 48 MHz möglich.

Weitere Ideen oder Vorschläge nehme ich gerne auf.

Joe

Autor: Peter Sieg (Gast)

Datum: 31.05.2010 19:59

---

Hallo Joe.

Deine Ideen finde ich prima! Für mich ist wichtig, das die Jumper Header (TTL Seriell) so bleiben, damit meine Kabel direkt passen..

Aber der [FT232RL](#) ist ja optional..und eine gute Idee.

Stromversorgung über USB Buchse ist auch eine gute Idee.. aber woher kommen die 3,3V.. oder habe ich da einen Regler übersehen..?

[ATmega88](#)+Dram+SD Card sollte über 3,3V versorgt werden.. ich vermute [FT232RL](#) benötigt 5V (und bekommt die ja auch direkt über die USB Buchse) und kann 3,3V TTL Pegel an Rx+Tx ab.. ich hoffe der [AVR](#) kann allerdings

auch +5V ab, obwohl er mit 3,3V versorgt wird..?

Programmierung auf der Platine ist ebenfalls eine gute Idee (wenn ich auch immer 6-Pol Header verwende ;-)

Wozu ist D2 [1N4148](#) da?

Wir funktioniert die optionale Taktgenerierung über den [FT232RL](#)?

Wie gesagt, wenn das Ganze auch ohne [FT232RL](#) Bstückung läuft und man die 3,3V aus dem USB Anschluß erzeugen kann, können wir gerne deine Platine bestellen, da sie nur 10mm breiter ist..

Im Forum64 gibt zumindest auch schon mal Interesse an 3 Platinen.. hier gab es auch Interesse an 1-2 Platinen und ich nehme 2.. langsam kommen wir auf die benötigten 10 ;-)

Peter

**Autor: Uhu Uhuhu ([uhu](#))**

**Datum: 31.05.2010 20:28**

---

[Peter Sieg schrieb:](#)

> aber woher kommen die 3,3V.

Im FTDI steckt ein 3,3V-Regler.

Mich würde die Platine auch interessieren.

**Autor: Joe G. ([feinmechaniker](#))**

**Datum: 31.05.2010 20:39**

---

Hallo Peter,

der [FT232RL](#) hat einen eigenen 3.3V Regler integriert. Er benötigt also selbst 5V aus der USB Schnittstelle oder muß über ein eigenes Netzteil versorgt werden.

Der Rx und Tx Pegel wird, je nach Versorgung des [FT232](#) entsprechend auf 5V oder 3.3V angepaßt. Die Brücke für die 5V hatte ich dafür vorgesehen um für mögliche 5V Programmieradapter kompatibel zu sein.

Für die Programmierung kann ich auch ein 6-Pol Header einsetzen, ist sogar etwas kleiner :-). Auch kam mir noch die Idee zu einem kleinen Resettaster.

Die Diode ist nur zum Schutz, siehe AVR042: [AVR](#) Hardware Design Considerations. Sie kann auch entfallen.

Der [FT232](#) ist nur dann optional, wenn die 3.3V extern zugeführt werden.

Die Takterzeugung läuft über die Programmierung des [FT232](#). Dazu gibt es von FTDI ein kleines Tool über welches das interne EEPROM geschoben wird.

Prima wenn es 10 Platinen werden....

Joe

Autor: Peter Sieg (Gast)  
Datum: 01.06.2010 09:44

---

@Joe: Ich verwende den [AVR](#) ISP Stick. Siehe hier:  
<http://www.mikrocontroller.net/articles/AVR-ISP-Stick>  
Wäre super, wenn du diesen 6-pol Programmierheader einbauen könntest.  
Resettaster parallel zu dem 2-pol Header ist prima, solange wir dadurch  
dir Platine nicht vergrößern.

Also noch mal für mich zum verstehen:  
- Man kann 3,3V extern einspeisen und direkt TTL seriell am 3-pol Header  
kommunizieren.. [FT232RL](#) bleibt dann unbestückt. Takt aus Quarz.  
- Zum Flashen kann +5V über USB Buchse eingespeist werden (Ohne SD  
Karte!)  
- Bei bestücktem [FT232RL](#) kann versorgt dieser die anderen Teile mit  
+3,3V und stellt natürlich auch direkt die Kommunikation über USB her  
(virtueller Com-Port). Alternativ kann man ihm auch noch beibringen den  
Takt für den [AVR](#) zu generieren, spart sich also den Quarz und die beiden  
C's dazu..

So richtig verstanden..?

Dann bitte noch mal an alle, die Platine von Joe und Schaltplan  
vergleichen..  
denn wir fertigen ja keinen Prototypen mehr an.. und Fehler rächen sich  
hier dann sofort an allen Platinen..

Peter

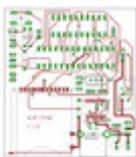
Autor: Uhu Uhuhu ([uhu](#))  
Datum: 01.06.2010 10:52

---

Ich würde vorschlagen, noch eine rote [LED](#) draufzusetzen, die leuchtet,  
wenn das Teil auf 5V gejumpert ist - dann werden nicht so viele SDs  
gebraten.

Autor: Joe G. ([feinmechaniker](#))   
Datum: 01.06.2010 13:50  
Angehängte Dateien:

- [AVRCPM.zip](#) (80,7 KB, 62 Downloads)



- [AVR\\_CPM.png](#)  
8 KB, 474 Downloads

---

@Peter:  
6-polige Programmierbuchse eingesetzt  
Resettaster eingefügt  
Ja, Du hast es richtig verstanden mit den 3.3V und dem TTL Header.

@Uhu:

Mir fällt keine so richtig einfache Schaltung ein die 5V auf der 3.3V Leitung zu erkennen. Möglich ist die Differenz zwischen 5V und 3.3V zu nutzen (1.7V) Eine rote [LED](#) sollte dann leuchten. Die Funktion wäre aber invers, sie leuchtet wenn 3.3V anliegen und nicht wenn 5V anliegen.

Prinzipiell:

Der [AVR](#) verträgt laut Datenblatt 1.8 - 5.5V also kein Problem.

Die SD-Cad nur 3.3V also beim Flashen ziehen!

Der DRAM 3.3V oder 5V Volt ja nach Type, ein 5V Type wird also außerhalb seiner Spezifikation betrieben.

Anbei das geänderte Layout bzw. die Schaltung nochmal zur Durchsicht.

Joe

**Autor: Peter Sieg (Gast)**

**Datum: 02.06.2010 10:51**

---

@alle:

Also: Joe G. hier aus dem Forum wird die Platinenbestellung übernehmen. Wir haben uns diesbezüglich schon abgestimmt.

Redaktionsschluß für Änderungen an der Platinen ist Sonntag, der 6.6 18Uhr MEZ. Verbindliche Bestellungen für

Platinen (und ggf. SD Slots) sind bis zum Sonntag, den 13.6 18Uhr MEZ möglich. Danach erfolgt die Bestellung.

Generell gilt für DIESE Platinenbestellung, das ursprüngliche Hardwaredesign ([ATmega88](#), 256kx4 Dram, SD Card) bleibt

unverändert! An Kosten ist mit etwa 10€/Platine zu rechnen (2-lagig, durchkontaktiert, Lötstoplack). SD Slot etwa 3€/Stück.

Wenn mehrere DRam's benötigt werden.. könnte ich diese Joe zusenden, der sie

dann mit der Bestellung verteilen kann..(Kosten 1€/Stück - Verrechnung mit meiner Platinenbestellung bei Joe)..?

Weitere Details zur Abwicklung von Joe G. (feinmechaniker)...

Peter

**Autor: Markus (Gast)**

**Datum: 02.06.2010 12:14**

---

Ein Frage: wie bekomme ich unter XP das disk image auf die SD-Card?

**Autor: Peter Sieg (Gast)**

**Datum: 02.06.2010 13:00**

---

@Markus: Mit dd.exe für win32 ;-)

Wenn du dir von hier:

<http://avr.cwsurf> (dot) de/ -> [AVR](#) CP/M

die cpmttools.zip ziehst, hast du nicht nur dd.exe, sondern auch die cpmttools um das diskimage unter Windows zu verändern (genauer: Programme löschen, hinzufügen, listen).

Peter

**Autor: Joe G. ([feinmechaniker](#))** ☰

**Datum: 03.06.2010 13:33**

---

@alle:

Wie Peter schon in beiden Foren geschrieben hat, nehme ich die Bestellungen für die [AVR](#) CP/M Platinen entgegen. Dazu bitte Eure benötigte Stückzahl, Eure Mailadresse sowie die Versandadresse über „Feinmechaniker“ an mich senden. Weiterhin würde ich die notwendigen SD-Card Sockel mitbestellen. Wer DRAM's benötigt ebenfalls kurze Info an mich. Es sind noch welche in geringer Stückzahl zu haben. Hier gilt - Verteilung nach Bestelleingang. Jede Bestellung erhält von mir eine kurze Bestätigung über den Eingang. (Wer schon bestellt hat, bitte nicht noch mal wiederholen.) Wenn die Bestätigung nicht sofort erfolgt, etwas Geduld, ich vergesse schon niemanden. Um die angekündigten Preise zu halten müssen mindestens 10 Platinen zusammenkommen. In diesem Fall liegt der Preis einer Platine (zweiseitig, durchkontaktiert, Lötstoplack) bei ca. 10 € netto. Ein SD-Card Sockel kostet 2.95€ und ein DRAM 1€. Dazu kommen noch 1,65€ für die Warensendung bzw. die Kosten für die Versandtasche. Eine genaue Kalkulation gibt es nach dem Bestelleingangsende, da erst dann der Leiterplattenpreis genau kalkuliert werden kann. Bis Sonntag 18 Uhr nehme ich noch Fehlerkorrekturen, Änderungsvorschläge, Verbesserungen der Schaltung auf. Wie Peter schon geschrieben hat gilt jedoch das Hardwaregrunddesign ([ATmega88](#), 256kx4 DRAM, SD Card). Ziel ist zunächst die Softwareentwicklung. Wer gleich mit einem SRAM arbeiten möchte, kann ihn ja über eine Lochrasterplatine auf die DRAM Fassung stecken. Adress-, Daten- und Steuerepins liegen an, der Rest ist NUR Softwareanpassung :-). Bestellungen nehme ich bis Sonntag den 13.6 entgegen. Am 14.6 löse ich die Bestellungen dann aus. Die Lieferzeit für die Platinen beträgt im Normalfall 8 AT, also bitte nicht schon früher drängeln.

Joe

**Autor: Uhu Uhuhu ([uhu](#))**

**Datum: 04.06.2010 09:38**

---

Zu den fehlenden Z80-Opcode-Routinen:  
CP/M ist ein auf dem 8080 lauffähiges OS. Der Befehlssatz des Z80 ist ein Superset des 8080-Befehlssatzes.

Was spricht eigentlich dafür, den [FT232RL](#) nur optional vorzusehen?

Wenn man ihn unbedingt vorsieht, kann man sich den ISP-Stecker sparen, wenn man dem Controller einen Bootlader spendiert.

Das hätte den Vorteil, daß jemand, der keinen ISP verfügbar hat, sich nur von irgend jemandem ein mal den Bootlader flashen lassen muß und das Teil dann selbst über den USB programmieren kann.

Das mit dem eZ80 ist eine Schnapsidee, das funktioniert im Leben nicht mit CP/M.

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 04.06.2010 14:14**

---

@Joe: Finde ich gut noch ein paar Ideen zur Platine zu diskutieren.. aber KISS (Keep It Simple & Stupid). Diese Platine soll nahe am Original mit kleinen Verbesserungen bleiben.. und das Original läuft auch bei mir auf Lochraster, einem (1) Abblockkondensator und 3,3V und gut ist.. Das Ganze soll ja eigentlich lediglich das leidige von Hand verdrahten auf Lochraster ersparen..

@Martin: Danke, aber das wußten wir schon.. wenn du Ideen hast wie man an der orig. Hardware+Firmware noch etwas besser machen kann.. nur zu.. ansonsten gab/gibt es hier auch eine 'Fraktion' die über ein Redesign ggf. mit 8-bit Sram nachdenken.. auch hier sind sicher weite Betätigungsfelder für Mitwirkende..

Just my 2 cents  
Peter

**Autor: Joerg Felgentreu (Gast)**  
**Datum: 05.06.2010 13:52**

---

Ich finde, daß wir bei dem vorgesehenen Muster mit CPM2.2 und den beiden Chips beliben sollten.  
Falls jemand was anderes bauen will, so sollte er ein neues Projekt aufziehen und nicht seine Fähigkeiten im Zerreden dieses Projektes einsetzen.

Für mich ist die ursprüngliche Richtung maßgebend.

Alle anderen Sachen wie

- CPM 3.0
- RTOS mit eigener HW
- SRAM Interface mit riesigen Erweiterungen

sollten in einen anderen Projekt erörtert werden. Mir gefällt das Projekt hier gerade, weil es so einfach ist.

**Autor: bix (Gast)**  
**Datum: 05.06.2010 17:09**

---

Habe gerade etwas entdeckt.

In den Kommentaren zum Original-Projekt auf der Seite <http://spritesmods.com/?art=avrcpm&page=4> wird auf ein Testprogramm für den 8080 hingewiesen: (instruction set test program EX8080.COM) Ist sicherlich hilfreich, wenn man fehlende Opcodes ergänzen möchte.

Hier der Link:  
<http://www.schorn.ch/cpm/intro.php>

Auf der Seite gibt es auch unterschiedliche Programmiersprachen, unter anderem Turbo Pascal.

## **Erste Schritte zum Parallelprojekt: AX81**

**Autor: Joerg Wolfram ([joergwolfram](#))**  
**Datum: 06.06.2010 01:28**

---

Letztendlich finde ich das Ganze doch so interessant, dass ich ein "Parallelprojekt" zumindest anfangen werde. Basis wird der aktuelle ChipBasic2 Computer ohne Modifikationen sein. Dazu ein 64K RAM Modul am Parallelport, welches zwar schon seit über einem Jahr bei mir existiert aber erst jetzt durch die ladbaren Treiber wirklich benutzbar wird.

In Punkto Videoauflösung ist es mir gelungen, Treiber für 24 Zeilen a 50 Zeichen und 24 Zeilen a 60 zeichen zu entwickeln. Der Zeichensatz umfasst 128 zeichen, sowohl normal als auch invers.

In der ersten Stufe (soweit Interesse daran ist), könnte ich ein einfaches Terminalprogramm integrieren (das RAM-Modul wird dazu nicht gebraucht). Dazu würde ich mich an z.B VT52 oder VT100 orientieren und implementieren, was halt möglich ist. Die Frage ist natürlich, ob man mit 60 zeichen/Zeile einigermaßen arbeiten kann.

Die nächste Stufe wäre dann, meinen (weitestgehend vorhandenen) Z80 Emulator an das System anzupassen. Eigentlich wollte ich damit mal einen ZX81 mit 2K RAM auf dem [AVR](#) simulieren, hab aber irgendwann aufgegeben...

Jörg

Autor: Rene B. ([themason](#)) 

Datum: 06.06.2010 03:08

---

Ich finde das Projekt einerseits interessant als CP/M Emulation, andererseits als kleine Plattform für weitere Mini-Computer. Es ist im Prinzip ja alles dafür da. 128KByte RAM, externer Speicher (in Form einer SD-Karte) und einer einfachen Schnittstelle nach außen hin. Weiterhin, alles schön einfach zu löten und per USB eben die Möglichkeit alles schön versorgen zu können. Als Experimentier-Plattform schön geeignet, und die IOs lassen sich ja auch noch einfach abgreifen.

Kurzum :

- Schönes CP/M Emulation-Nostalige-Projekt
- Einfache Plattform zum Aufbau und Experimentieren eines Mini-Computers

Daher will ich auch son Dingen :-)

Autor: Peter Sieg (Gast)

Datum: 06.06.2010 11:55

---

@Jörg: Prima! Mit deiner Erfahrung und Vorkenntnissen bin ich gespannt! Und der Herbst kommt ja auch nach dem Sommer ;-) Da kann wieder mehr gebastelt werden..

Ansonsten freue ich mich aber ersteinmal auf Platine Version 1 zum etwas stabileren Aufbau des Originals mit Komm. und Stromversorgung über USB..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 06.06.2010 17:53

Angehängte Dateien:

- [Eagle AVRCPM.zip](#) (83,7 KB, 81 Downloads)

---

@alle

Die Überarbeitung der Schaltung und dem Layout der Version 1.0 sind nun abgeschlossen. Wie zu sehen, sind nicht alle Änderungen berücksichtigt.

Speziell die Variante mit dem zusätzlichen Busstecker hätte größere Änderungen bedeutet.

Zur Bestellung der Platinen:

Alle die bei mir Platinen sowie Bauelemente bestellt haben müssten nun eine Mail in Ihrem Postfach haben. Wer keine Mail bekommen hat, der hat noch nicht bestellt oder ich habe ihn übersehen, dann bitte wie schon oben beschrieben nochmal wiederholen. Auf gleichem Wege bekommt ihr das EEPROM File für den [FT232](#) und eine kleine Hardware-doku.

Joe

**Autor: Höhlenmaler (Gast)**

**Datum: 08.06.2010 00:09**

---

Mein erster Computer mit CP/M 2.2 war voll Eigenbau und hatte 4 Karten im Einschubgehäuse.

Auf der ersten Karte waren nur der Z80 mit Quarz und die PIO zum Anschluss der selbst gebauten Voll-Tastatur.

Die zweite Karte hatte 32 DRAM Chips mit je 16Kx1, eine Refreshlogik in TTL, 4 NiCd AA Zellen als Notstromversorgung und einen [NE555](#) mit 10uH Drossel als Stepupwandler. Weil mir hier zwei Gatter fehlten, aber kein Platz vorhanden war, hab ich mit etwas Diodenlogik nachgeholfen.

Die dritte Karte enthielt eine CTC und ein diskret aufgebautes Kassettenrekorder Interface.

Die vierte war die Grafikkarte mit 2kByte RAM für 24 Zeilen mal 64 Zeichen, viel TTL und einem Kanal 3 Modulator. Noch jetzt erinnere ich mich, wie ich in der Badewanne lag und auf Kästchenpapier die 256 Zeichen aufgemalt und daneben die Hexcodes geschrieben habe.

Das BIOS war auch selbst geschrieben, nach einer kaum lesbaren Interfacebeschreibung auf braunem Thermokopiepapier aus irgendeiner Zeitschrift.

Der Computer war ein Spitzenteil womit ich anfangs in Pascal und später in C programmiert habe. Die 64kByte DRAM teilten sich Betriebssystem, Arbeitsspeicher und RAM-Disk. Der C-Compiler bestand aus zwei COMs die man nacheinander vom Kassettenrekorder in die RAM-Disk laden musste um so den Quelltext zu kompilieren.

Übrigens wurden davon bestimmt 10 Stück nachgebaut und ich habe zumindest den Aufwand daran verdient. Vom Spaß nicht zu reden.

Die Leiterplatten-Layouts wurden mit Tusche per Hand auf karierten Karton gemalt und dann im Kontaktverfahren auf Film übertragen.

Und ja, wir haben schon mit Messer und Gabel gegessen.

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 08.06.2010 13:26**

---

Hmm.. mit Recht..?

Nun, ich denke.. was die graphisch und spielerischen Aspekte und vorallen Dingen die audiovisuellen, erstklassischen Demos angeht.. Amiga=klares JA.

(Inzw. ja auch auf der PC Plattform ein beliebtes Genre)

Aber ansonsten.. ist doch inzw. das 'Netz' der Computer.. soll heißen ohne

E-Mail, Web, Streaming etc. wäre doch der Frontend=PC 'nur' ein

graphisch, verspielter CP/M Rechner ;-)

Die 'Killerapplikationen'=Textverarbeitung, Tabellenkalkulation, Datenbank gab es 'damals' auch schon (Wordstar, Multiplan, dbase). Und mit einem ASCII basierten Mini-Linux kann man auch ohne schnick-schnack Mails lesen und schreiben und Webseiten lesen etc.

Also Inhaltlich/Funktional sind die Urgestalten der CP/M Ära gar nicht soo weit weg.. und manchmal sicher noch einfacher zu verstehen und zu kontrollieren.. nur Kiddies kann man damit schwer hinter dem Ofen vorlocken ;-). Dazu muß es ja heute klickibund und animiert sein.. - Inhalt: Nebensache - ;-)

Sorry,  
Peter

**Autor: Pieter (Gast)**

**Datum: 09.06.2010 13:14**

---

moin moin,

vorsicht, 8MB war doch die maximale Größe je LW die CP/M verwalten konnte.  
WordStar konnte nur Dateien bis 512KB verarbeiten.  
Meine RAM-Disk hatte zuletzt 2BM mit 64Stück U61000.

MfG  
Pieter

**Autor: A. K. ([prx](#))**

**Datum: 09.06.2010 13:32**

---

Verstehe ich nicht. Ein normales CP/M benötigte einen Floppy-Treiber im BIOS. Dieses CP/M bräuchte einen Flash-Treiber ohne Filesystem, im BIOS oder (vorzugweise) im Emulator. Das Filesystem bringt CP/M selber mit - muss ja kein FAT sein.

**Autor: Uhu Uhuhu ([uhu](#))**

**Datum: 09.06.2010 13:49**

---

OK, mit dem Filesystem hast du recht.

Die SD-Karte kann man sich trotzdem nicht sparen, wenn man die Daten mit den CP/M-Tools ins System bringen will.

Aber ich denke, weitere Debatten hier über derlei Verbesserungen erübrigen sich trotzdem. Hier geht es darum, für das von Peter Sieg vorgeschlagene System ein Platinchen zu basteln.

Für mich ist das hier wirklich ein reines Retro-Projekt. Ich habe nicht vor, mit CP/M ernsthaft zu arbeiten.

**Autor: bix (Gast)**

**Datum: 12.06.2010 13:47**

Angehängte Dateien:

- [avrcp\\_m.sch](#) (308 KB, 136 Downloads)

- [AVRCPM\\_HW\\_Rev2.txt](#) (2,5 KB, 189 Downloads)
- [avrcpm\\_100612\\_V2.zip](#) (169,9 KB, 67 Downloads)



- [avrcpm\\_brd.png](#)  
31,1 KB, 286 Downloads

---

[Joe G. schrieb:](#)

> @alle:  
> Wie Peter schon in beiden Foren geschrieben hat, nehme ich die  
> Bestellungen für die [AVR](#) CP/M Platinen entgegen. Dazu bitte Eure  
> benötigte Stückzahl, Eure Mailadresse sowie die Versandadresse über  
> „Feinmechaniker“ an mich senden. Weiterhin würde ich die notwendigen  
> SD-Card Sockel mitbestellen.

Kurz von Torschluss nochmal nachgefragt...

Wurden die letzten Änderungen in das Layout übernommen (Isolationsabstand, Beschriftung?). Da ich nichts mehr gehört habe, habe ich die Änderungen mal eingezeichnet (in die Version vom 6.6.2010), ein paar schiefe Leitungen gerade gezogen und die Beschriftung für Top und Bottom-Layer hinzugefügt. Der DRC läuft nun sauber durch.

Beim Durchsehen der Software fiel mir auf, dass man einige Signale günstiger verbinden kann, um den Zugriff auf das RAM zu beschleunigen. Man muss dann nicht so viele Bits per Software einzeln zusammen setzen.

Um jedoch am Original-Layout nichts zu ändern, habe ich noch zwei 3\*2 Stiftleisten hinzugefügt, die nirgends angeschlossen sind. Es ist aber möglich, später durch Auftrennen von vier Leitungen und Anschluss an die Stiftleisten zwischen original Layout (Hardware Version 1) und optimiertem Layout umzuschalten (Hardware Version 2), falls man das will.

Wie gesagt, es ist optional und freiwillig. Ändert man nichts, läuft alles wie bisher, so hat man keinen Nachteil.

Die optionalen Änderungen sind im beiliegenden Text "AVRCPM\_HW\_Rev2" beschrieben sowie ein kurzes Beispiel, wie die Zugriffsroutinen auf das RAM sich vereinfachen (dram\_getnibble und dram\_sendnibble).

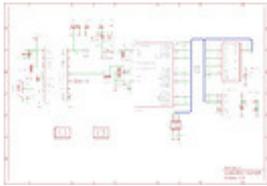
Noch eine Frage:

Wie lange dauert die Freischaltung zum Forum normalerweise? Ich kriege einfach keine Bestätigungsmail und kann mich nicht anmelden, möchte aber auch bestellen.

**Autor: bix (Gast)**

**Datum: 12.06.2010 13:52**

**Angehängte Dateien:**



[avr\\_cpm\\_schem.png](#)

21,2 KB, 537 Downloads

---

Huch

Das Schaltbild wollte ich als PNG anhängen. Hab mich verklickt, Anhänge zeigt die Vorschau leider nicht an - sorry.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 12.06.2010 14:25

---

@bix

keine Torschlusspanik, ist doch noch ein Tag Zeit. Die letzten Änderungen (Beschriftung, Isolationsabstand) sind schon eingearbeitet. Weiterhin noch ein 100n C zwischen /DTR und Reset. Somit könnte der [AVR](#) mit einem Bootloader und Reset über /DTR versehen werden und die Programmierung über ISP entfällt. Damit sollte die Gefahr, die SD-Karte versehentlich mit 5V zu versorgen, genommen sein. Die Idee mit dem kleinen Steckfeld ist gut, ich habe sie übernommen.

Joe

Autor: bix (Gast)

Datum: 12.06.2010 14:44

---

[Joe G. schrieb:](#)

> Die Idee mit dem  
> kleinen Steckfeld ist gut, ich habe sie übernommen.

Hallo Joe,

danke für die schnelle Antwort.

Damit das mit den Stiftleisten später funktioniert (auf Hardware Version 2 Jumpfern), habe ich die im Text mit (\*) gekennzeichneten Pins des ATmega so auf die Lötseite der Platine verlegt, dass man die Leitungen später einfach durchtrennen kann (erkennbar an den quardatischen "Vias").

Ist erst die IC-Fassung aufgelötet, kommt man auf der Bauteileseite ja später nicht mehr an die Leitungen heran.

Es sind beim [AVR](#) die Pins 5, 19, 24 und 27.

Pin 19 musste ich vom [AVR](#) zum RAM verlegen, damit der Pin zum [AVR](#) einzeln durchtrennt werden kann, ohne die restliche Leitung zu unterbrechen (s. Bild [avr\\_cpm\\_brd.png](#)).

```
; ATmega8 (HW-Rev.2)
;
;
;
;          +-----v-----+
;   Reset\ | 01 (RESET)      PC5 28 | CAS\
;   TXD    | 02 (RXD)       PC4 27 | RAS\ (*)
;   RXD    | 03 (TXD)       PC3 26 | D3
;   OE\    | 04 PD2         PC2 25 | D1
```

```

;  (*) WE\      | 05 PD3          PC1 24 | D2  (*)
;  SD_CS\     | 06 PD4          PC0 23 | D0
;  VCC        | 07 VCC          GND 22 | GND
;  GND        | 08 GND          AREF 21 | n.c.
;  Q20MHz     | 09 (XTAL1)      AVCC 20 | VCC
;  Q20MHz     | 10 (XTAL2)      (SCK) PB5 19 | A8 / SD_SCK (*)
;  A5         | 11 PD5          (MISO) PB4 18 | A0 / SD_MISO
;  A6         | 12 PD6          (MOSI) PB3 17 | A1 / SD_MOSI
;  A7         | 13 PD7          PB2 16 | A2
;  A4         | 14 PB0          PB1 15 | A3
;  +-----+

```

E-Mail mache ich gleich fertig.

Danke und Grüße

## Gamesdsk+cpmtools

Autor: Peter Sieg (Gast)

Datum: 14.06.2010 08:31

Angehängte Dateien:

- [gamesdsk.zip](#) (138,1 KB, 72 Downloads)

---

Hier noch mal eine andere 'Floppy' = diskimage für [AVR](#) CP/M, mit Spielen:

- \* Mastermind (mmind) (BDS-C Source vorhanden)
- \* Tic-Tac-Toe (ttt2) (BDS-C Source vorhanden)
- \* othello (22k Fortran Source vorhanden)
- \* othello (12k BDS-C Source vorhanden)
- \* stone (BDS-C Source vorhanden)

Solche images werden ja laut orig. Anweisung direkt mit:

```
dd if=<image> of=<device> bs=512
```

übertragen. VORSICHT! Absolut sicher sein, das <device> auch die richtige

SD Karte ist!! Besser vorher alle anderen Wechseldatenträger abziehen!

(Ein evtl. vorhandenenes Dateisystem wird gnadenlos überschrieben und man hat ein ideales Objekt für forensische Analysen und Neuformattierung ;-)

Die cpmtools im Original gibts übrigens hier:

<http://www.moria.de/~michael/cpmtools/>

Peter

## Eagle3D Bild der ersten Platine

Autor: Peter Sieg (Gast)

Datum: 16.06.2010 11:29

Angehängte Dateien:



- [avrcp\\_m.jpg](#)  
108,4 KB, 578 Downloads

---

Soo ähnlich sollte die voll bestückte Platine dann aussehen ;-)  
Leider kennt Eagle3D die USB Buchse (noch) nicht und auch einige C's  
sehen  
etwas komisch aus.. na ja.. ;-)

Peter

## Der Entwickler meldet sich zu Wort

Autor: Peter Sieg (Gast)  
Datum: 17.06.2010 10:14

---

Hier noch eine Rückmeldung vom Entwickler:

---

Hiya,

Damn, I never expected such a thread to grow out of something this silly  
:) Even with pre-made PCB's, wow.

I use something which calls itself 'Z80 assembler version 1.8', which is  
in the Debian-repositories as 'z80asm'. I think you can use just about  
any  
sane Z80 assembler, though.

Jeroen

---

Peter

## Erste Analysen des Quellcodes

Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)  
Datum: 17.06.2010 10:29  
Angehängte Dateien:

- [z80.asm](#) (67,6 KB, 232 Downloads) | [Codeansicht](#)

---

Ich hab mal angefangen das z80.asm auseinanderzunehmen um zu sehen, wo es noch hapert.

Der entscheidende Teil kommt ab Zeile 1766, dort sind die ganzen Ausführungs-'einheiten' zusammengefasst, die die eigentliche Arbeit machen.

Ich hab da jetzt mal eine OP-Code Tabelle mit reingenommen, aus der hervorgeht, welcher Befehl welche Flags beeinflusst. Im Moment bin ich dran, die vorhandenen Funktionseinheiten durchzugehen und zu kontrollieren, ob die Flagbehandlung mit dieser Tabelle übereinstimmt.

Das Parity-Flag wird leider tatsächlich ziemlich stiefmütterlich behandelt. So wie das gemacht ist, geht das gar nicht. Es reicht einfach nicht aus, die Parity nur bei einer direkten Parity-Abfrage auszuwerten. Da muss auf jeden Fall noch nachgebessert werden.

Wenn sich wer an der Arbeit beteiligen will ....

Bei jeder Funktionseinheit steht dabei, von welchen Opcodes sie benutzt wird. Eine abschliessende Kommentarzeile

```
; OK
```

weist darauf hin, dass diese Funktion schon überprüft ist. Im Gegensatz zu

```
; Not yet checked
```

Wenn sich wer die Mühe macht, und die noch fehlenden OpCodes zu implementieren, bitte auf die Flags achten. Die Flagbehandlung kann bei ähnlichen Befehlen unterschiedlich sein! Das sieht man in der OpCode-Übersichtstabelle:

```
RLC  A
RLCA
```

haben zwar auf den ersten Blick dieselbe Funktion, die Behandlung des Parity-Flags ist aber anders.

Im Zuge der Kontrolle tausche ich auch die magischen Bitkonstanten für die Flags gegen besser lesbare 1<<irgendwas Konstrukte aus.

**Autor: Peter Sieg (Gast)**

**Datum: 17.06.2010 13:29**

---

@Karl Heinz Buchegger: Super!! Leider habe ich von Z80 keine Ahnung.. Was ich aber schon geändert hatte, war die dynamische Berechnung der Baudrate anhand der Quarzfrequenz:

```
.equ F_CPU = 20000000          ; system clock in Hz
.equ BAUD  = 38400             ; baud rate
```

```
; Berechnungen
```

```
.equ UBRR_VAL = ((F_CPU+BAUD*8)/(BAUD*16)-1) ; clever rounding
```

```
...
```

```
; - Init serial port
```

```
ldi temp,$18
sts ucsr0b,temp
ldi temp,$6
sts ucsr0c,temp
```

```
ldi temp,0
sts ubrr0h,temp
ldi temp,UBRR_VAL
sts ubrr0l,temp
```

Der so geänderte Code assembliert und läuft (selbst getestet).  
Ich empfehle allen Platinenbauer den Quarz zu sockeln.. dann man man  
später  
einfacher auf einen 32MHz Quarz wechseln..

Den Lochrasteraufbau werde ich so original belassen (20MHz).. da kann  
ich  
dann aber sehr gerne auch verbesserte Firmware testen..

Bei den Platinen werde ich das dann zuerst mit 20MHz aufbauen, danach  
mal mit 32MHz probieren, wie viel das bringt.. und danach die weiter  
oben beschriebene Pin-Änderung und read\_nibble/write\_nibble probieren..  
Muß mal sehen, ob es da irgendein fertiges Benchmark-Programm gibt..

Peter

**Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)**

**Datum: 17.06.2010 13:49**

---

[Peter Sieg schrieb:](#)

> Was ich aber schon geändert hatte, war die dynamische Berechnung der  
> Baudrate anhand der Quarzfrequenz:

Machs bitte gleich richtig und rechne das High Register auch aus.  
An solchen Stellen bitte auch immer gleich die magischen Konstanten  
austauschen, damit man auch ohne Datenblatt zumindest eine Vorstellung  
davon bekommt, was da ein/aus geschaltet wird.

```
; - Init serial port
ldi temp, (1<<TXEN0) | (1<<RXEN0)
sts ucsr0b,temp
ldi temp, (1<<UCSZ01) | (1<<UCSZ00)
sts ucsr0c,temp
ldi temp, HIGH(UBRR_VAL)
sts ubrr0h,temp
ldi temp, LOW(UBRR_VAL)
sts ubrr0l,temp
```

**Autor: Uhu Uhuhu ([uhu](#))**

**Datum: 17.06.2010 14:06**

---

[Peter Sieg schrieb:](#)

> Leider habe ich von Z80 keine Ahnung..

Wenn du den 8080 kannst, ist das kein Problem. Der Z80 hat ein paar  
Befehle mehr und einen zweiten Registersatz, auf den mit einem der  
Spezialbefehle umgeschaltet werden kann.

Echte 8080-Programme laufen auf dem Z80 ohne Änderung.

## Leo C. steigt in die Entwicklung ein

Autor: Leo C. ([rapid](#))

Datum: 18.06.2010 15:13

---

Ich habe den Thread hier von Anfang an mitverfolgt, hatte aber eigentlich nicht vor, mich zu beteiligen. Leider habe ich gestern in den von kbuchegg geposteten Source geschaut...

Gestern Abend habe ich von einem alten Speicherriegel ein DRAM (4Mx4, 3,3V, SOJ) runtergeholt, und auf einen Lochrasteradapter fürs Steckbrett gelötet. Leider habe ich keinen Mega88 mehr gefunden. Deshalb bin ich gerade dabei, den Source auf Mega8 anzupassen.

Als Assembler benutze ich den avra unter Linux. Die aktuelle Release hat ein paar Macken, die in der Git-Version behoben sind.

[Karl heinz Buchegger schrieb:](#)

```
> Ich hab da jetzt mal eine OP-Code Tabelle mit reingenommen, aus der
> hervorgeht, welcher Befehl welche Flags beeinflusst. Im Moment bin ich
> dran, die vorhandenen Funktionseinheiten durchzugehen und zu
> kontrollieren, ob die Flagbehandlung mit dieser Tabelle übereinstimmt.
```

Danke dafür. Das hat mir den Einstieg deutlich erleichtert.

```
> Das Parity-Flag wird leider tatsächlich ziemlich stiefmütterlich
> behandelt. So wie das gemacht ist, geht das gar nicht. Es reicht einfach
> nicht aus, die Parity nur bei einer direkten Parity-Abfrage auszuwerten.
> Da muss auf jeden Fall noch nachgebessert werden.
```

Entweder verstehe ich da noch etwas nicht, oder die Sache wird so, wie Sprite\_tm sich das vorgestellt hat, nicht funktionieren.

```
do_fetch_af:
    mov opl,z_flags
    mov oph,z_a
    rcall do_op_calcparity
    andi opl, ~(1<<ZFL_P)
    sbrs temp2,0
    ori opl, (1<<ZFL_P)
    ret
```

Der Code wird bei "PUSH AF" verwendet. Hier wird das Parityflag berechnet und im Statusregister im kombinierten P/V-Flag gespeichert. Anschließend wird das Register auf dem Stack gespeichert.

Wenn die letzte OP vor dem PUSH eine arithmetische war, ist das V-Flag nach dem POP verloren (durch P-Flag vom zuletzt gespeicherten parityb überschrieben).

Wenn ich richtig liege, fallen mir 3 Lösungsmöglichkeiten ein.

1. Den Parity-Status (parityb) beim PUSH in einem der ungenutzten Statusregisterbits (Bit 3 oder 5) speichern.
2. Bei jedem Befehl merken, ob Parity gültig ist, und den Merker in do\_fetch\_af auswerten.
3. Parity bei den entsprechenden Befehlen immer sofort berechnen und das P/V-Flag setzen.

Persönlich tendiere ich zu 3. Im 8080-Befehlssatz sind das ja nicht so viele Befehle, wenn auch häufig vorkommende.

Autor: Peter Sieg (Gast)

Datum: 19.06.2010 17:35

---

Leo C. schrieb:

> Ich habe den Thread hier von Anfang an mitverfolgt, hatte aber  
> eigentlich nicht vor, mich zu beteiligen. Leider habe ich gestern in den  
> von kbuchegg geposteten Source geschaut...  
>  
> Gestern Abend habe ich von einem alten Speicherriegel ein DRAM (4Mx4,  
> 3,3V, SOJ) runtergeholt, und auf einen Lochrasteradapter fürs Steckbrett  
> gelötet. Leider habe ich keinen Mega88 mehr gefunden. Deshalb bin ich  
> gerade dabei, den Source auf Mega8 anzupassen.  
>  
> Als Assembler benutze ich den avra unter Linux. Die aktuelle Release hat  
> ein paar Macken, die in der Git-Version behoben sind.

Hallo und herzlich willkommen im Mini [AVR](#) + CP/M Retro Thread ;-) )

Peter

Autor: Leo C. ([rapid](#))

Datum: 19.06.2010 22:14

---

Danke für die Begrüßung. Ich habe übers Wochenende Besuch und bin deshalb noch nicht weiter gekommen. Nur ganz kurz:

Der AVRA funktioniert leider nicht richtig. Letzte Nacht habe ich dann in meiner Verzweiflung den Atmel Windows Assembler unter Wine ausprobiert. Funktioniert. Prost.  
Aber mein Mega8 geht noch nicht. :-(

## Der Artikel zum Projekt wird aufgesetzt

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 20.06.2010 13:15

---

Hallo,

um eine Struktur in die nun beginnenden Soft- und Hardwareänderungen zu bringen hier mein Vorschlag zur Projektseite:  
[http://www.mikrocontroller.net/articles/AVR\\_CP/M](http://www.mikrocontroller.net/articles/AVR_CP/M)

Gruß Joe

## SVN Server

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 21.06.2010 10:20

---

@Martin  
@alle

Alle aktuellen Dateien liegen hier auf dem [SVN](http://www.mikrocontroller.net/svn/list) Server  
<http://www.mikrocontroller.net/svn/list>

Joe

## Weiter arbeiten an der Software

Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)

Datum: 21.06.2010 13:30

---

[Leo C. schrieb:](#)

```
> 3. Parity bei den entsprechenden Befehlen immer sofort berechnen und das
>   P/V-Flag setzen.
>
> Persönlich tendiere ich zu 3. Im 8080-Befehlssatz sind das ja nicht so
> viele Befehle, wenn auch häufig vorkommende.
```

Das ist auch meine Präferenz.  
Bei den 'OK' markierten Sequenzen ist das schon so.

Autor: Leo C. ([rapid](#))

Datum: 21.06.2010 14:01

---

[Karl heinz Buchegger schrieb:](#)

```
>> 3. Parity bei den entsprechenden Befehlen immer sofort berechnen und das
>>   P/V-Flag setzen.
```

```
> Bei den 'OK' markierten Sequenzen ist das schon so.
```

Hmm, ich meine, daß man hier:

```
do_op_inc:
    andi  z_flags, (1<<ZFL_C)      ; bis auf Carry alles auf 0
    ldi   temp, 1
    add  opl, temp
    in   temp, sreg
    mov  parityb, opl
    bst  temp, AVR_Z                ; Zero
    bld  z_flags, ZFL_Z
    sbrc opl, 7                    ; Sign
    ori  z_flags, (1<<ZFL_S)
    bst  temp, AVR_H                ; Half Sign
    bld  z_flags, ZFL_H
    bst  temp, AVR_C                ; Overflow
    bld  z_flags, ZFL_P
    ret
```

das Zwischenspeichern des Operanden in parityb sparen kann, und  
bei den Befehlen AND/OR/XOR/DAA das Parityflag direkt berechnen muß.

Oder ich habe eben doch noch etwas übersehen...

Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)

Datum: 21.06.2010 14:14

---

[Leo C. schrieb:](#)

> das Zwischenspeichern des Operanden in parityb sparen kann, und  
> bei den Befehlen AND/OR/XOR/DAA das Parityflag direkt berechnen muß.

Mag sein.

Ich hab mich noch nicht damit beschäftigt, wo man im Code noch einsparen bzw. verändern könnte/müsste.

Ich wollte nicht zu viele Veränderungen auf einmal machen. Erst mal eine Bestandsaufnahme bzw. den Code in eine etwas besser lesbare Form giessen.

Wenn im ersten Durchgang diese Dinge hier alle verschwunden sind

```
andi z_flags,0b11101100
```

und durch besser lesbare Konstrukte in expliziter Bitnamenschreibweise ersetzt sind,

wenn für alle bisherigen Ausführungseinheiten dabeisteht, von welchem Opcode sie benutzt werden und die Flags (bis auf das Parity Flag) überprüft sind,

dann wäre mein nächster Schritt, die Behandlung des Parity Flags in die relevanten OpCodes einzubauen

gefolgt von:

- \* wo gibt es offensichtliche Verbesserungsmöglichkeiten
- \* restliche OpCodes implementieren

Aber das Auseinanderpfriemeln des jetzigen Codes ist mühsam.

PS: Ich kann leider von hier nicht auf das [SVN](#)-Repositor zugreifen (Ich schätze mal, dass da eine Firewall was dagegen hat. Das dürfte ich aber beim Admin nicht durchbringen).

Autor: Leo C. ([rapid](#))

Datum: 21.06.2010 22:28

---

Karl heinz Buchegger schrieb :

> Mag sein.

> Ich hab mich noch nicht damit beschäftigt, wo man im Code noch einsparen  
> bzw. verändern könnte/müsste.

> Ich wollte nicht zu viele Veränderungen auf einmal machen. Erst mal eine  
> Bestandsaufnahme bzw. den Code in eine etwas besser lesbare Form  
> giessen.

Ja das verstehe ich. Mir ging es auch nicht um einsparen im Sinne von Optimieren, sondern darum, daß der Code meiner Meinung nach fehlerhaft ist.

Ich wollte eigentlich da einsteigen, wo Du es vorgeschlagen hast und bin dann über die fehlerhafte P/V-Flag-Behandlung gestolpert. Es macht imho keinen Sinn, überall "; OK" dran zu schreiben, wenn man nachher nochmal alles umwerfen muß.

> PS: Ich kann leider von hier nicht auf das [SVN](#)-Repositor zugreifen (Ich  
> schätze mal, dass da eine Firewall was dagegen hat. Das dürfte ich aber  
> beim Admin nicht durchbringen).

Lesend gehts bei mir schon mal.

Inzwischen läuft mein Steckbrett-CP/M-AVR-Rechner. Mit [ATmega8](#), 12.288MHz und 3.3V. Falls jemand Interesse an meinen Anpassungen an Mega8 hat, würde die etwas polieren, und hier (oder im [SVN](#)) hochladen.

Autor: Karl Heinz Buchegger ([kbuchegg](#)) (Moderator)

Datum: 22.06.2010 09:28

---

[Leo C. schrieb:](#)

> Ich wollte eigentlich da einsteigen, wo Du es vorgeschlagen hast und bin  
> dann über die fehlerhafte P/V-Flag-Behandlung gestolpert. Es macht imho  
> keinen Sinn, überall "; OK" dran zu schreiben, wenn man nachher nochmal  
> alles umwerfen muß.

OK.

Ich dachte du willst das vereinfachen.

Hmm. die P/V Behandlung sollte meiner Meinung nach bei den paar Anweisungen schon stimmen. Auch darauf achten, dass dieses Flag eine Doppelbedeutung hat.

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 22.06.2010 11:51

---

Soweit ich weiss, hat der 8080 nur ein P Flag und kein P/V. Das kam erst beim Z80. Damit sollten sich sogar beide Prozessoren auseinanderhalten lassen:

```
test:      sub    A                ;setzt A auf 0 (kein Overflow)
           jp     pe,code_8080    ;wenn A=0 dann ist Parity even
code_z80:  ....
```

```
code_8080:  ....
```

Jörg

## Bilder von Leo C. Testaufbau

Autor: Leo C. ([rapid](#))

Datum: 22.06.2010 19:15

Angehängte Dateien:

- [avr-m8\\_z80.asm](#) (69 KB, 192 Downloads) | [Codeansicht](#)



[Breadboard-CPMAVR- bersicht.JPG](#)

244,7 KB, 585 Downloads

---

Peter Sieg schrieb:

> @Leo C.: Aber Sicher besteht daran Interesse!

Eigentlich machts ja keinen Sinn, den Mega8 zu nehmen. Bei 3,3V müßte man die L-Version nehmen, und die geht offiziell nur bis 8 MHz.

Andererseits passt das ja ganz gut zu diesem sinnlosen Projekt. :)

> Und ein Bild des Aufbaus wäre auch schön ;-)

Wieso? ;) Bei 2 ICs und einer SD-Karte gibts doch gar nichts zu sehen. Ich habe keine brauchbare Kamera. Aber gestern Morgen war mein Besuch (mit Kamera) noch da, und da habe ich vorsorglich ein paar Bilder gemacht. Allerdings war das RAM da noch nicht verdrahtet.

Autor: Leo C. ([rapid](#))

Datum: 22.06.2010 19:33

Angehängte Dateien:



• [Breadboard-CPMAVR-1.JPG](#)

183,5 KB, 559 Downloads

---

Karl heinz Buchegger schrieb:

> Hmm. die P/V Behandlung sollte meiner Meinung nach bei den paar

> Anweisungen schon stimmen.

Da habe ich eben bedenken.

> Auch darauf achten, dass dieses Flag eine

> Doppelbedeutung hat.

Im jetzigen Programm werden V-Flag und P-Flag getrennt gespeichert.

V im Statusregister und P in "parityb".

Bei PUSH AF wird P ins Statusregister kopiert und auf den Stack gespeichert.

Nach einem anschließenden POP AF ist ein evtl. vorher berechnetes V-Flag hinüber.

Bei reinen 8080-Programmen dürfte das keine Rolle spielen, da sie ja kein

V-Flag kennen. Aber dann kann man sich die V-Flag-Ermittlung auch sparen.

Und bei (zukünftigen) Z80-Programmen wirts dann fehlerhaft.

Voller Z80-Befehlsatz wäre schon interessant, da z.B. Turbo Pascal nur auf Z80 läuft.

# Die erste Platine aus der Sammelbestellung ist fertig

Autor: Joe G. ([feinmechaniker](#))

Datum: 25.06.2010 16:35

Angehängte Dateien:



[AVR\\_CPM.jpg](#)

85,6 KB, 626 Downloads

---

So sieht sie aus, die erste bestückte Platine.

Joe

## Weiter arbeiten an der Software

Autor: Leo C. ([rapid](#))

Datum: 25.06.2010 20:01

---

Juchhu,  
DDT kann jetzt Programme tracen.

Der DAA-Befehl hatte gefehlt, und im RST-Befehl war ein kleiner Fehler.

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)

**[Re: CP/M auf ATmega88](#)**

Autor: Leo C. ([rapid](#))

Datum: 26.06.2010 04:14

Angehängte Dateien:

- [bios.asm](#) (3,7 KB, 216 Downloads) | [Codeansicht](#)

---

Ich habe im BIOS mal die Warmboot-Funktion ergänzt (CCP + BDOS neu laden).

Wer einen funktionierenden Warmboot haben möchte, kann das angehängte BIOS installieren.

Autor: Martin (Gast)

Datum: 26.06.2010 20:13

---

In dem Assemblerfile "" sind einige Adressen festgelegt:

```
ccp: equ $3400+bias ;base of cpm ccp
bdos: equ ccp+$806 ;base of bdos
```

Wo liegt den nun CP/M?

Autor: spess53 (Gast)

Datum: 26.06.2010 20:26

---

Hi

BDOS gehört zum CP/M (+BIOS und CCP).

MfG Spess

Autor: Leo C. ([rapid](#))

Datum: 28.06.2010 19:12

Angehängte Dateien:

- [bios.bin](#) (532 Bytes, 74 Downloads)

---

Peter Sieg schrieb:

```
> @Leo C.: Prima! Super, das jetzt auch der Warmstart geht.. kannst du
> bitte auch das assemblierte Bios.bin hier einstellen..
```

```
> Welchen Z80 Assembler hast du genommen (Link)?
```

<http://www.nongnu.org/z80asm/>

Ich habe den genommen, der in meiner Linux Distri drin ist. Sprite\_tm nimmt wohl den gleichen. Jedenfalls lief sein Makefile damit fehlerfrei durch.

Das Teil ist aber in der Tat etwas eigenartig.

Autor: Leo C. ([rapid](#))

Datum: 29.06.2010 22:43

---

[Peter Sieg schrieb:](#)

```
> Zu einem anderen Aufbau sehe ich das aber letztlich genauso.. was
> ein 'fauler' Kompromiss ist und viel Rechenpower kostet muß durch
> etwas besseres ersetzt werden.. Da kann man beim Ram auch z.B über SRAM
> nachdenken.. dann braucht man keinen Refresh mehr..
```

Mit dem Refresh habe ich mich letzte Woche ausgiebig beschäftigt. Der kostet so gut wie nichts. Ich habe auch mal "hidden refresh" realisiert. Das lohnt einfach nicht. Und für die Zeiten, in denen mal keine RAM-Zugriffe sind (zB. Disk-I/O), bräuchte man den normalen Refresh doch noch.

Viel sinnvoller wäre es, die verwurstete Port-Zuordnung zu verbessern.

Wie's richtig gut geht, findet man (mal wieder) auf der auch hier allseits bekannten Seite von Chan.

Autor: TheMason (Gast)

Datum: 30.06.2010 12:26

---

@Leo

>Viel sinnvoller wäre es, die verwurstete Port-Zuordnung zu verbessern.

Dazu mal eine Frage.

Ist es nicht letztendlich egal wie die Port-Zuordnung zwischen [AVR](#) und DRAM ist ?

Es sind zwar die Adress-Leitungen vertauscht, aber im Prinzip macht es ja nichts aus, solange beim Schreiben und beim Lesen die gleiche Zuordnung verwendet wird. Oder hat das Auswirkungen auf den Refresh (sprich das sequentiell eine Row nach der anderen Refreshed werden muß, und sobald Rows übersprungen werden gehen Daten verloren oder ist das unabhängig davon das sie zu einem anderen Zeitpunkt (aber dann dennoch im selben zeitlichen Raster) refreshed werden.

Also nach meinem Verständnis müsste es eig. egal sein wie die Zuordnung der Adress-Leitungen ist. Hauptsache sie ist beim Schreiben/Lesen gleich, und alle Rows werden im festen Raster (aber unterschiedliche Reihenfolge) refreshed.

Ich gestehe aber das ich mir den DRAM-ASM Code noch nicht angeguckt habe. Hatte nur im Schaltplan gesehen das es etwas "Kraut und Rüben" war.

Autor: Leo C. ([rapid](#))

Datum: 30.06.2010 13:49

---

[TheMason](#) schrieb:

> Dazu mal eine Frage.

> Ist es nicht letztendlich egal wie die Port-Zuordnung zwischen [AVR](#) und  
> DRAM ist ?

Im Prinzip schon. Das beweist ja auch die Originalschaltung.

Wenn aber zB. die 8 niederwertigen Adressleitungen auf einem Port liegen würden, könnte man sie zusammen mit einem out-Befehl ausgeben.

Ansonsten muß man die Adressbits durch Bitschiebereien oder Einzelbit-tests

und -Sets auf die verstreuten Adressleitungen verteilen.

Beim [ATmega88](#) kann man allerdings bei keinem einzigen der 3 Ports alle Bits zusammenhängend verwenden. Auf Port B liegt der Quartz, auf Port D die serielle Schnittstelle, und Port C hat überhaupt nur 6 Bit (wenn man nicht auf Reset verzichten will, sonst halt 7).

Bei den Adressen kann man hier also gar nicht so viel verbessern.

Die Datenbits sind bei der Schaltung aber auch unglücklich verteilt.

Wenn man WE mit einer der Datenleitungen tauscht, liegen die 4 Datenleitungen

auf dem unteren Nibble eines Bytes.

Dann kann man die zwei Hälften eines Bytes zB. so einlesen

```
in temp,PINC          ; 1. Nibble einlesen (untere Hälfte von PC)
andi temp,0x0f        ; obere Hälfte wegmaskieren
swap temp
```

```
    ;Adressen umschalten ...
```

```

in temp2,PINC      ; 2. Nibble einlesen
andi temp2,0x0f
or temp,temp2     ; Beide Hälften zusammen

```

Zusammen 6 Befehle, 6 Takte.

Im Originalprogramm gibt es ein Unterprogramm, das zwei mal aufgerufen wird:

```

;...
rcall dram_getnibble
swap temp

        ;Adressen umschalten ...

rcall dram_getnibble

        ; ...
;...

```

```

dram_getnibble:
andi temp,0xf0
sbic pinc,ram_d0
ori temp,0x1
sbic pinc,ram_d1
ori temp,0x2
sbic pinc,ram_d2
ori temp,0x4
sbic pinc,ram_d3
ori temp,0x8
ret

```

Auch wenn man den Unterprogrammaufruf weglässt (man könnte ja den Code direkt einfügen), sind das schon  $1 + 2 \times 9 = 19$  Takte.

Die Adressen-Ausgabe und -Umschaltung kann man aber auch noch deutlich optimieren. Auch ohne Hardwareänderung.

Durch die Änderungen habe ich eine Geschwindigkeitsverbesserung bei CP/M-Programmen von fast 50% erreicht.

Den Code dazu will ich nachher noch posten. Jetzt ist erst mal Mittag.

**Autor: Peter Sieg (Gast)**

**Datum: 30.06.2010 15:08**

---

@Leo C. Da faellt mir ein, das ich auch gerne das 2te asm Programm als BIN Datei haette (war glaube ich cpm.asm -> cpm.bin), dann man ueber die cpmttools auch ersteinmal ohne z80 Assembler eine neues diskimage erstellen..

Das mit den 50% hoert sich ja schon mal super an!

Gruss Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 30.06.2010 18:44**

---

[TheMason schrieb:](#)

> Na ja. Die Adress-Pin-Belegung ist wirklich nicht die glücklichste. Wäre dann vllt was für eine Folge-Version :-)

Ja, nachdem ich mir das jetzt nochmal angeschaut habe, glaube ich auch, daß eine neue Sortierung einiges bringen würde. An einen Punkt hatte ich heute Mittag nicht gedacht: Die Originalroutinen sind auch deshalb langsam, weil Sprite\_tm viele NOPs eingefügt hat. Alleine das weglassen der überflüssigen NOPs dürfte den Zugriff schon deutlich beschleunigen.

**Autor:** bix (Gast)

**Datum:** 30.06.2010 18:57

---

[TheMason schrieb:](#)

> Na ja. Die Adress-Pin-Belegung ist wirklich nicht die glücklichste. Wäre dann vllt was für eine Folge-Version :-)

Die ungünstige Belegung vom RAM zu den Ports wurde hier schon diskutiert, siehe diverse Posts vom 12.06.2010 und folgende.

Aus diesem Grund wurde auch das Layout geändert und Platz für ein paar Jumper geschaffen, um zwischen alter und neuer Belegung umzuschalten.

Die ungünstigen Leitungen zum [AVR](#) sind gegenüber dem ersten Layout von Joe auf die Lötseite verlegt, so dass man auch bei bestücktem Board nachträglich noch Leiterbahnen auftrennen und Patchen kann.

Gruß, bix

**Autor:** Leo C. ([rapid](#))

**Datum:** 30.06.2010 19:08

**Angehängte Dateien:**

- [avr-m8\\_z80.asm](#) (86,6 KB, 178 Downloads) | [Codeansicht](#)

---

Im Anhang ist mein aktueller Softwarestand. Es sind eine Menge Änderungen und Erweiterungen. Sicher wäre es besser gewesen, die Änderungen schrittweise zu posten, aber das schaffe ich wohl nicht.

Erst mal das wichtigste:

Im CPU-Emulator habe ich alle Flags so eingestellt, daß sie sich (möglichst) wie auf einem 8080 verhalten. Der Emulator kann bisher nur 8080, und Software, die anhand der Flags glaubt, einen Z80 unter sich zu haben, fällt mit großer Wahrscheinlichkeit aufs Kreuz, wenn wie Z80-Befehle dann nicht da sind. Und diese Software gibts für CPM. (siehe auch [Beitrag "Re: CP/M auf ATmega88"](#))

Der 8080 Befehlssatz ist jetzt komplett und weitere Fehler habe ich keine mehr gefunden. (Was aber nichts heißen muß.)

Jetzt kommen meine Spielereien:

\* Schnellere DRAM-Routinen

Geht jetzt nur, wenn man am Prozessor 2 Leitungen vertauscht. Sonst muß man die alten Routinen verwenden.

Es könnte sein, daß das Timing für sehr langsame (alte) RAMs zu scharf eingestellt ist. Das muß nochmal überprüft und evtl. geändert werden.

\* Interrupt und Empfangspuffer für USART Input.  
Mich hats einfach genervt, das immer Zeichen verloren gehen, wenn man mit der Maus in die CP/M-Console pasted.

\* Timer-Interrupt mit 1ms Auflösung. War in erster Linie für Performance-Messungen gedacht, kann aber auch als Echtzeituhr verwendet werden. 32 Bit Sekundenzähler ala UNIX time\_t. Nachdem ich das fertig hatte, habe ich beim simh Altair 8800 Simulator (<http://www.schorn.ch/cpm/intro.php>, tolles Teil übrigens) noch eine andere Idee gesehen, die ich dann auch noch eingebaut habe: Man kann vom CP/M aus einen Timer triggern, der ggf. die abgelaufene Zeit auf die Console schreibt.

Folgende DEFINES kann man dem Assembler auf der Befehlszeile mitgeben (Make-/Projektfile):

DRAM\_DQ\_ORDER : 0 = Original Pinbelegung. 1 = WE und DQ1 vertauscht.  
D.h. auf PC0..PC4 sind die DRAM-Datenbits.  
In meinem DRAM-Datenblatt heißen die Datenleitungen DQ und sind von 0 bis 3 durchnumiert.  
Achtung: Bei Sprite\_tm (und im Datenblatt seines DRAMs) laufen die Nummern von 1 bis 4.  
Bei der alten Pinbelegung (DRAM\_DQ\_ORDER = 0) werden die alten Routinen verwendet, sonst die neuen. Wer es anders haben möchte, kann ja mal nach "CLASSIC\_DRAM" suchen. Neue Software mit alter Pinbelegung geht aber nicht. Das habe ich zwischendurch mal rausgeworfen, als ich bei den vielen #if/#else den Durchblick verloren hatte.

Target-CPU : [atmega8](#), atmega168 oder [atmega88](#). Default ist [atmega88](#)  
Ich habe nur [atmega8](#) getestet, weil ich die anderen erst bestellen müßte (Oder ein anders Gerät fleddern :).

F\_CPU : Taktfrequenz. Default ist 20MHz

BAUD : Default ist 38400

Im Programm gibts noch weitere Macros:

REFR\_RATE : DRAM Refresh-Rate. Eingestellt sind die üblichen 15.6µs, entspr 64KHz. Um den Interrupt-Overhead etwas zu reduzieren, wird der Timer auf die halbe Rate programmiert, und dann immer 2 Reihen auf einmal aufgefrischt.

EM\_Z80 : Auf 1 setzen, wenn man einen 8080 mit Z80-Flags testen will.  
(Oder wenn man alle Z80-Befehle eingebaut hat :)

Kritik in Form von Patches willkommen.

Autor: Leo C. ([rapid](#))

Datum: 01.07.2010 07:04

Angehängte Dateien:

- [avr-m8\\_z80.asm](#) (87 KB, 181 Downloads) | [Codeansicht](#)
- [timer.tgz](#) (1,1 KB, 79 Downloads)

---

Update: Der Timer wird jetzt in einem vernünftigen Format ausgegeben.  
Die Uhr (Uptime) ist der Einfachheit halber im gleichen Format.

Außerdem hänge ich noch ein kleines Programmchen an, mit dem man den Timer bedienen kann.

## Erste Benchmarks

Autor: Leo C. ([rapid](#))

Datum: 01.07.2010 08:11

---

Hier mal ein kleiner Test.

Im Debugger habe ich folgende kleine Schleife laufen lassen:

```
MVI A,01      ;
OUT 40      ;Start Timer      Takte
LXI H,0000   ;                10
DCX H       ;                6 -\
MOV A,H     ;                4 |
ORA L       ;                4 |
PUSH PSW   ;                10 | 44 * 65536 = 2883584
POP PSW    ;                10 |
JNZ 0107   ;                10 -/
MVI A,02   ;                7
OUT 40     ;Stop Timer      11
NOP       ;                gesamt Anzahl Takte: 2883612
```

Und das hier kam raus:

-----				
AVR Atmega mit		12,288 Mhz		20 MHz
-----				
		Laufzeit		Z80
		Gesamt		Speed
-----				
Orig. DRAM R/W		17,187 s		168 KHz
Neue DRAM R/W		9,806 s		273 KHz
-----				
simh AltairZ80		0,022 s		7,629 ns
				131 MHz

Zum Vergleich habe ich noch die Zahlen von einem Simulator auf meinem PC dazugetan. (08/15 PC mit 2,5MHz Athlon 4850e)

Mein [AVR](#) läuft nur mit 12,288 MHz. Deshalb habe ich die Zahlen noch für 20MHz hochgerechnet. Die Zahlen erscheinen mir durchaus plausibel, wenn ich an die Geschwindigkeit von meinem 4MHz CP/M-Rechner zurückdenke.

Ganz oben hat Jörg Wolfram geschrieben, daß er mit einem [AVR](#) ca. 2MHz Z80-Geschwindigkeit erreicht hat. Na, dann gibts hier ja noch einiges an \*Spiel\*raum...

Autor: Leo C. ([rapid](#))

Datum: 03.07.2010 16:18

---

[Joe G. schrieb:](#)

> @Leo C.

> Ich habe heute nochmal beide RAM Varianten getestet. Gestern war der

> Ramtest leider doch aus.

Aha, so macht es mehr Sinn.

> Testumgebung: einmal 8 Mhz Takt / einmal 24 Mhz Takt

Bei 24Mhz ist der Refresh bei langsamen RAMs an einer Stelle an der Grenze. Die Schreib/Leseroutinen von Sprite\_tm haben Luft ohne Ende. Die gehen wahrscheinlich auch bei 100MHz noch. Und meine Routinen nimmst Du ja nicht.

> Ergebnis Taktunabhängig

>

> RAM-Test alte Variante:

> 23 < 00

> 23 < 01

> 22 < 02

> 23 < 03

> 26 < 04

> 27 < 05

> 26 < 06

> Es sieht also so aus, ob Bit 5 und Bit 1 immer 1 sind.

Das ist das Bit, das ich mit WE getauscht habe. Ich hab mir das jetzt nochmal genau angeschaut (Assembler Listing). Testen kann ich leider nicht. Ich meine, daß bei "DRAM\_DQ\_ORDER = 0" alle Bits an der richtigen Stelle für die Originalschaltung sind.

> RAM-Test neue Variante:

> 00 < 00

> 00 < 01

> 00 < 02

> 00 < 03

> als ob er überhaupt nicht angesprochen wird. Ich werde mal WE an den Oszi > klemmen.

Daß, das gehen sollte, stand ja nie zur Diskussion.

> Deine Version vom 01.07, der Option DRAM\_DQ\_ORDER = 0, und dem Mega168. und

> .include "m8def.inc"

> ... hatte ich schon geändert.

Das gibt beim Übersetzen noch einen Fehler:

```
| avr-m8_z80.asm(1230): error: Old hardware can not work with new software!
```

Die Zeile 1230 und drumherum müßttest Du also auch geändert haben, sonst hättest Du nicht fehlerlos übersetzen können. Man kann diese "Sicherheitsabfrage" einfach rauswerfen. Darüber wird (bei DRAM\_DQ\_ORDER = 0) der richtige (dh. der Originaltreiber von Sprite\_tm) genommen. Das meine Routinen mit der Originalpinbelegung nicht gehen, hatte ich ja schon geschrieben.

Sonst fällt mir dazu nichts mehr ein. Wenn die Hardware mit dem Originalprogramm funktioniert, dann müßte sie mit meinem Programm und den oben beschriebenen Anpassungen auch laufen.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 03.07.2010 18:40

---

Das 4:0 hat es gebracht. Die Variante "DRAM\_DQ\_ORDER = 0" geht nun. War mein Fehler, 0 und 1 vertauscht. Die neue Variante "DRAM\_DQ\_ORDER = 1" natürlich mit den beiden getauschten Leitungen bringt immer noch

> 00 < 00  
> 00 < 01  
> 00 < 02  
> 00 < 03

bei RAMORG 1. Bei RAMORG 0 kommt nicht mal mehr eine Ausschrift, hängt sich irgendwo ganz auf.

Autor: Leo C. ([rapid](#))

Datum: 03.07.2010 19:04

---

[Joe G. schrieb:](#)

> bei RAMORG 1. Bei RAMORG 0 kommt nicht mal mehr eine Ausschrift, hängt sich irgendwo ganz auf.

RAMORG=0 ist sozusagen ungetestet. Das hätte ich wohl rauswerfen sollen. RAMORG=1 sollte aber wirklich funktionieren.

Allerdings kommen hier so langsam die Zugriffszeiten, die genaue Chiporganisation usw. ins Spiel....

> Und welchen RAM-Chip hast Du (Genaue Typbezeichnung mit Speed-Code)?

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 04.07.2010 08:04

Angehängte Dateien:



• [MTC4256-80.pdf](#) | [anzeigen](#)

632,8 KB, 264 Downloads

---

[Leo C. schrieb:](#)

>> Und welchen RAM-Chip hast Du (Genaue Typbezeichnung mit Speed-Code)?

einen MT4C4256-80

Autor: Leo C. ([rapid](#))

Datum: 04.07.2010 09:05

---

> einen MT4C4256-80

Da dürfte es eigentlich keine Timing-Probleme geben. Kannst Du Die Schaltung probeweise mit 5V betreiben? Dann ohne SD-Karte um zu sehen, ob sie dann über den RAM-Test kommt.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 04.07.2010 10:37

---

[Leo C. schrieb:](#)

> Kannst Du Die Schaltung probeweise mit 5V betreiben?

> Dann ohne SD-Karte um zu sehen, ob sie dann über den RAM-Test kommt.

Auch mit 5V ohne Erfolg.

Ich habe nun mal die Signale am RAM und am [AVR](#) gemessen. Wenn das Programm in den Ramtest läuft liegt /WE ständig auf Low und I/O1 bis I/O4 liefert keine Daten bzw. es kommen keine vom [AVR](#). /RAS und /CAS liegen aber normal an.

Autor: Leo C. ([rapid](#))

Datum: 04.07.2010 11:34

---

[Joe G. schrieb:](#)

> Wenn das Programm in den Ramtest läuft liegt /WE ständig auf Low  
> und I/O1 bis I/O4 liefert keine Daten bzw. es kommen keine vom [AVR](#).

Du hast anscheinend ein andere Programm als ich. ;)

Du könntest mal schauen, ob die entsprechenden Portbits als Ausgänge programmiert werden, und zb. an der WE-Leitung auch High-Pegel ausgegeben werden. Bei mir ist das der Fall.

```
[avr-m8_z80.lst]
                                ; - Setup Ports
...
000045 e300      ldi temp,PC_OUTPUT_MASK
000046 b907      out DDRC,temp
...

                                dram_write:
0002f6 94f8      cli
...
0002f7 b117      in temp2,DDRC                ;DRAM data ports as outputs
0002f8 601f      ori temp2,RAM_DQ_MASK
0002f9 b917      out DDRC,temp2
...
000340 9a44      sbi PORTC,ram_w
...
000342 b107      in temp,DDRC
000343 7f00      andi temp,~RAM_DQ_MASK
000344 b907      out DDRC,temp
000345 b108      in temp,PORTC
000346 7f00      andi temp,~RAM_DQ_MASK
000347 b908      out PORTC,temp
```

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 06.07.2010 11:06

Angehängte Dateien:

- [avr\\_cpm.asm](#) (88,7 KB, 171 Downloads) | [Codeansicht](#)

---

Im Anhang der derzeit aktuelle Softwarestand:

Die Änderungen umfassen die Änderungen von Leo C. (siehe oben), die Änderungen für ein korrektes Timing des MT4C4256-80 und die Änderungen zur korrekten Initialisierung der SD-Card. Damit sollten nun auch alte langsame Karten zuverlässig laufen.

Bei Hardwareänderungen bitte beachten:  
/WE (PC1 - Pin 24) wird mit IO3 (PC4 - Pin 27) getauscht. Auf der Leiterplatte auch an den rechteckigen Durchkontaktierungen erkennbar. Welche RAM-Version benutzt wird kann als Compileroption angegeben werden.

Kritik erwünscht

## SVN von Leo C. und Wunschliste

Autor: Leo C. ([rapid](#))

Datum: 09.07.2010 12:04

---

### Neuigkeiten

Ich habe "meine" Sourcen in ein [SVN](#) gestellt auf daß man hier zugreifen kann:

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

Meine Änderungen kann man dort nachlesen. Wer Fehler findet, der ... muß sie nicht behalten, sondern darf sie gerne hier mitteilen und am Besten gleich korrigieren. Ich werde die nächsten Tage nicht dazu kommen.

Ich habe eine radikal geänderte Portbelegung getestet. Das Konzept dazu habe ich (mal wieder) bei Chan abgekupfert:

<http://elm-chan.org/works/vp/report.html>

Der Geschwindigkeitsgewinn ist nicht überwältigend, aber doch beachtlich. Und als Belohnung gibt es 3 freie Ports.

Hier mal die Ergebnisse mit meiner weiter oben geposteten, nicht repräsentativen Testschleife:

[AVR](#) Atmega mit 20 MHz:

	Laufzeit Gesamt	Laufzeit pro Z80 Takt	Z80 Speed
Letzter Stand:	5,921 s	2,05 µs	478 KHz
aktuell: (trunk 28)	3,239 s	1,12 µs	890 KHz
geänderte Portbelegung: (experimental 25)	2,974 s	1,03 µs	969 KHz

Inzwischen habe ich einen 20MHz Quarz an meinem Mega8 (undervoltaged/overclocked :). Mit 24.5 MHz macht er allerdings die Grätsche.

Den 1MHZ 8080/Z80 wird man mit dem jetzigen Konzept von Sprite\_tm wahrscheinlich noch erreichen können. Für die 2 MHz wird man wohl den schönen Interpreter wegwerfen/total umkrepeln müssen.

Auf jeden Fall gibts noch viel zu tun. Für den Fall, daß jemand Langeweile hat, hänge ich einfach mal meinen uneditierten Merktzettel mit ein paar Ideen an. Die Liste ist sicher nicht vollständig.

Wünsche schönes Wochenende

## avr\_cpm Wunschliste

-----

- USART IO per Interrupt  
(DONE für Input)
- DRAM
  - Adress- und Daten-Ports neu sortieren für schnelleren Zugriff
  - Adress- und Datenleitungen multiplexen: --> 4 freie Ports  
(ist ohne Performanceverlust möglich)
  - memRead/WriteWord?
  - Memory block transfer?
- DEBUG/trace steuerbar durch Z80-Programm?  
(Über IN/OUT-Interface)
- Andere (größere) Disk-Formate  
dpm muß für avr zugänglich sein?
- Register umorganisieren
  - für effizienteren code
  - R1 als 0-Register? (wie gcc-avr)
- Source aufteilen
  - console I/O (USART)
  - mmc
  - dram
  - z80 emu (?)
- Z80 Befehle
- bios sector blocking/deblocking (avr)
- Mehrere CP/M-Laufwerke auf einer SD ?
  - CP/M-Images in FAT-Dateisystem
  - Diskimages hintereinander kopieren

**Autor: Peter Sieg (Gast)**

**Datum: 12.07.2010 09:16**

---

Hallo. Bin wieder da ;-)

Habe mir gerade den aktuellen Stand von avr\_cpm.asm (Joe G.) gezogen und versucht ersteinmal für die original HW zu assemblieren mit:

```
#ifndef DRAM_DQ_ORDER      /* If this is set to 1, the portbits */
#define DRAM_DQ_ORDER 0    /* for DRAM IO3 and WE are swapped.    */
#endif
```

dabei bekomme ich:

```
C:\E\AVR\avr-cpm\avrcpm\avr\z80.asm(1253): error: Old hardware can not
work with new software!
```

???

Ich finde die Optionen zur DRAM Selektion SEHR unübersichtlich!  
Das versteht doch kein Mensch mehr.. da müssen andere Namen+mehr  
Erläuterungen in den ASM Quellcode..

Nun, was muß ich wie setzen, um ersteinmal für die original HW zu  
assemblieren??

Danach wollte ich dann mal probieren PC1 + PC4 zu tauschen.  
Das sollte dann wohl mit #define DRAM\_DQ\_ORDER 1 gehen?

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 12.07.2010 09:45**

---

ok. Habe mit DRAM\_DQ\_ORDER 1 assembliert und PC1 und PC4 getauscht.  
Damit läuft und MBASIC arbeitet jetzt!! Hurra!

Super Arbeit! Vielen Dank dafür ;-)

Trotzdem wäre es schön, wenn die original HW über die entsprechende  
Option  
weiterhin nutzbar wäre mir der neuen SW...

Jetzt müssen nur noch die Platinen kommen.. dann gehts weiter...

Peter

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 12.07.2010 11:09**

---

@Peter

Willkommen im Backofen!

Die alte HW Variante läuft auch mit der von Dir gezogenen  
Softwareversion. Gib einfach unter Assembler Optionen -DDRAM\_DQ\_ORDER=0  
ein.

Damit wird CLASSIC\_DRAM = 1 gesetzt. Ja, ist unglücklich, einmal 0 und  
einmal 1. Die Software geht nun auch ohne Pintausch.

Zusätzlich muß Du noch die Fehlermeldung deaktivieren!

```
; ----- DRAM routines -----  
; TODO:  
  
#if DRAM_DQ_ORDER == 1  
  #define CLASSIC_DRAM 0  
#else  
  #define CLASSIC_DRAM 1    /* Change manually, if you want new hw w/ old  
sw */  
#endif
```

Die Abfrage hier kannst du streichen

```
; #if DRAM_DQ_ORDER == 0  
; #if CLASSIC_DRAM == 1  
;   #error "Old hardware can not work with new software!"  
; #endif  
; #endif
```

Die Hardware ist auf dem Wege zu Dir.

Gruß

Joe

Autor: Peter Sieg (Gast)  
Datum: 12.07.2010 13:25

---

Danke Joe.

Die von dir genannte Abfrage, welche den Fehler produziert MUSS dann aber raus, da ich ja zuerst mit DRAM\_DQ\_ORDER = 0 **nicht** erfolgreich assembliert hatte.. ;-)

Peter

Autor: Leo C. ([rapid](#))  
Datum: 12.07.2010 13:43

---

[Peter Sieg schrieb:](#)

> Die von dir genannte Abfrage, welche den Fehler produziert MUSS dann  
> aber  
> raus, da ich ja zuerst mit DRAM\_DQ\_ORDER = 0 **nicht** erfolgreich  
> assembliert hatte.. ;-)

Vergiß den Kram einfach. Zieh Dir lieber mal das hier, und sag Deine Meinung dazu:

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

[edit]

Ich sehe gerade, daß beim aktuellen Stand bei den #defines oben die Pintauschmöglichkeit über "DRAM\_DQ\_ORDER" noch drin ist, die Schreib/Lese-Routinen das aber nicht mehr unterstützen. Falls ein überwältigendes Interesse daran besteht, die äußerst unpraktische Originalpinbelegung am Laufen zu halten, werde ich das vielleicht wieder einbauen. Bisher besteht allerdings gar kein Interesse.

[Peter Sieg schrieb:](#)

> Hallo. Bin wieder da ;-)

Ich jetzt auch. :)

> Ich finde die Optionen zur DRAM Selektion SEHR unübersichtlich!  
> Das versteht doch kein Mensch mehr.. da müssen andere Namen+mehr  
> Erläuterungen in den ASM Quellcode..

Das Ganze war nur zum Testen gedacht. Insbesondere die alten Routinen hatte ich zuerst nur dringelassen, damit man mit den neuen vergleichen kann.

Dann wurde mir der Aufwand zu groß, die neuen Routinen für alte Pinbelegungen einzubauen und hatte dann schnell diese Schalter eingebaut und nicht mal richtig getestet.

Autor: Peter Sieg (Gast)  
Datum: 12.07.2010 17:50

---

@Leo C. Solange es ja nur 2 Leitungen/Pins sind, die zu tauschen sind.. sehe ich da auch kein wirkliches Problem.. habe die jetzt auf meiner LR Version umgelötet.. wenn ich die Platinen bekomme.. mach ich das da dann genauso.. ist ja ein prima Geschwindigkeitszuwachs für die kleine Änderung.

Hast du für deine Speedmessungen eigentlich ein fertiges Programm genutzt?

Falls ja, könnte man das ja als Referenz nehmen..?

Anschauen werde ich mir das alles in den nächsten Tagen.. bin aber kein Experte.. daher kann ich da selbst 'nur' lernen.. ;-)

Aber MBASIC.COM läuft jetzt! Das ist doch schon mal was! Gleich mal ein paar Basic Programme raussuchen ;-)

Dann wollte ich mal schauen, ob mal ohne großen Aufwand die das Dickimage vergrößern kann.. z.B mehr Sektoren etc. etc... 250kb sind schnell voll..

und selbst auf einer winzigen 16MB SD Karte ist so viel Platz ;-)

Auf jeden Fall freut es mich riesig, das aus diesem 'sinnlosen' Proof-of-Concept ein richtiges, kleinen Bastelprojekt geworden ist!!

Und die Verbesserungen können sich sehen lassen:

- \* CP/M Warmstart
- \* 8080 Opcodes korrigiert
- \* Dram Access beschleunigt

Peter

Autor: Peter Sieg ([petersieg](#))

Datum: 14.07.2010 19:48

Angehängte Dateien:



• [avrcpm2.jpg](#)

198,1 KB, 465 Downloads

---

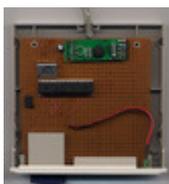
So. Platinen sind angekommen. Nochmals vielen Dank an Joe G. dafür! Erste aufgebaut und läuft ;-)

Peter

Autor: Leo C. ([rapid](#))

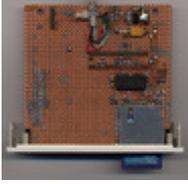
Datum: 14.07.2010 20:50

Angehängte Dateien:



• [avrcpm-compi-4.png](#)

249 KB, 480 Downloads



[avrcpm-compi-r.png](#)

2,5 MB, 590 Downloads

---

Huch, ist hier ein Posting von mir gelöscht worden, oder habe ich das am Montag tatsächlich nicht abgeschickt? Fällt mir jetzt erst auf, weil ich jetzt auch ein Bild von dem "Gerät" schicken wollte, daß ich am Montag angekündigt hatte. Ich habe es gestern aufgebaut, und es läuft.

Also auf ein Neues.

Peter Sieg schrieb:

> @Leo C. Solange es ja nur 2 Leitungen/Pins sind, die zu tauschen sind..  
> sehe ich da auch kein wirkliches Problem.. habe die jetzt auf meiner LR  
> Version umgelötet.. wenn ich die Platinen bekomme.. mach ich das da dann  
> genauso.. ist ja ein prima Geschwindigkeitszuwachs für die kleine  
> Änderung.  
>  
> Hast du für deine Speedmessungen eigentlich ein fertiges Programm  
> genutzt?  
> Falls ja, könnte man das ja als Referenz nehmen..?

Das "Testprogramm" ist hier:

[Beitrag "Re: CP/M auf ATmega88"](#)

> Aber MBASIC.COM läuft jetzt! Das ist doch schon mal was! Gleich mal ein  
> paar Basic Programme raussuchen ;-)

MBASIC ist leider immer noch A-langsam. Ich habe praktisch nur ELIZA ausprobiert, und die ist nicht benutzbar.

In BASIC kannst Du Zeiten z.B. so messen:

```
100 REM Timertest
110 TCTRPORT = &H40
120 REM Start timer
130 OUT TCTRPORT,1
140 REM Do something
150 FOR I=0 TO 200
160 IF I MOD 10 = 0 THEN PRINT ".";
170 NEXT
180 REM Stop timer, print result
190 OUT TCTRPORT,15
200 PRINT
210 PRINT "Done."
220 END
```

> Dann wollte ich mal schauen, ob mal ohne großen Aufwand die das  
> Dickimage  
> vergrößern kann.. z.B mehr Sektoren etc. etc... 250kb sind schnell  
> voll..  
> und selbst auf einer winzigen 16MB SD Karte ist so viel Platz ;-)

Dazu habe ich am Montag auch einiges geschrieben. Jetzt nur kurz  
Stichworte: Bios, Tabellen DPH, DPB, ...

Inzwischen bin ich sozusagen dran. Aber erst kommt das Sector Blocking/Deblocking an die Reihe.

Zu den Bildern:

Das Gehäuse ist dieses hier:

<http://www.pollin.de/shop/dt/MTcwOTcyOTk-/Computer...>

Als (SD-)Kartenleser war das Gerät am PC absolut unbrauchbar. Aber 1€ für Gehäuse + SD-Slot ...

Edit:

Beim 2. Bild habe ich leider daneben gehauen. Da wollte ich eigentlich auch eine verkleinerte Version schicken. Tut mir leid.

Autor: Joe G. ([feinmechaniker](#))

Datum: 15.07.2010 12:35

---

[Peter Sieg schrieb:](#)

> Erste aufgebaut und läuft ;-)

Sieht doch gut aus!

Übrigens, Abblock C's sind dafür da, dass man sie als fehlend im Layout ktitisiert und später nicht bestückt ;-)

Autor: Rene B. ([themason](#))

Datum: 15.07.2010 13:39

---

Meine Platine ist auch da (schon vor 1-2 Wochen, hatte aber erst letztes WE Zeit sie zu bestücken). Lauffähig ist diese, allerdings zickt der [FT232](#) noch rum. Habe den Treiber für den VCP installiert, bekomme aber mit Putty dennoch kein Zeichen rein/rausgeschaufelt. Muß da mal ausprobieren. CP/M hab ich demnach noch nicht getestet. Werde mir aber die Sourcen von [SVN](#) runterladen und mal durchstöbern. Vllt auch CP/M zum laufen bringen. Aber muß mich da erst einlesen.

Dickes Lob an Joe für die Sammelbestellung und Peter für die Initiative dieses Projektes. Weiter so Jungs. Klasse :-)

## Benchmarking die 2te

Autor: Peter Sieg (Gast)

Datum: 16.07.2010 19:51

---

[Leo C. schrieb:](#)

```
> Im Debugger habe ich folgende kleine Schleife laufen lassen:
> MVI A,01          ;
> OUT 40           ;Start Timer          Takte
> LXI H,0000       ;                    10
> DCX H           ;                    6 -\
> MOV A,H         ;                    4 |
> ORA L           ;                    4 |
> PUSH PSW       ;                    10 | 44 * 65536 = 2883584
> POP PSW        ;                    10 |
> JNZ 0107        ;                    10 -/
> MVI A,02       ;                    7
```

```
> OUT 40      ;Stop Timer          11
> NOP                ;                gesamt Anzahl Takte: 2883612
```

Hi Leo C.

wäre es nicht super, diese 'Benchmark' Routine in dein timer (t.com) Programm mit aufzunehmen..? -> t b würde dann den Timer starten, die Schleife ausführen und direkt danach die verstrichene Zeit ausgeben..

So hätten wir ein ungefähres Mass für Optimierungen..

Habe die 2te Platine noch ohne HW-Änderungen am laufen mit D\* = 0.  
=> geht also noch mit den alten Sprite\_tm DRAM Routinen..

BTW: Welches fuses hast du für den [Atmega8](#) gesetzt?  
Möchte da auch mal einen mir 20mhz + 3,3v 'quälen' ;-)

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 19.07.2010 00:00**

---

Hmm.. versuche micg gerade mal in z80 asm und möchte den timer.asm um eine delay-benchroutine erweitern.. Programm läuft auch, aber egal, ob in hl eine 0, 20 oder 50 geladen wird.. das Ergebnis ist immer 2,138/9 Sekunden??

Ist bestimmt wieder so absolut blöder Fehler ;-)

bench:

```
        ld      a,startTimerCmd
    out  (TIMERCTL),a ; start
; loop starts here
    ld      hl,0
11:          ld      b,0
12:          ld      a,0
13:  dec    a
    jp     nz,13      ; 256 x
        dec    b
    jp     nz,12      ; 256 x
    dec   hl
    jp     nz,11      ; 256 x
; loop ends here
        ld      a,printTimerCmd
    out  (TIMERCTL),a ; print elapsed time
    ret                ; and done
```

Peter

**Autor: Joerg Wolfram ([joergwolfram](#))**

**Datum: 19.07.2010 10:58**

---

DEC HL setzt soviel ich weiss keine Flags. Also müsste man wie im ersten Testbeispiel ein

```
MOV A,H
ORA L
```

nach das DEC HL setzen.

Jörg

Autor: Leo C. ([rapid](#))

Datum: 19.07.2010 12:37

Angehängte Dateien:

- [tloop.com](#) (128 Bytes, 103 Downloads)

---

Hallo Peter,

> wäre es nicht super, diese 'Benchmark' Routine in dein timer (t.com)  
> Programm mit aufzunehmen..? -> t b würde dann den Timer starten,  
> die Schleife ausführen und direkt danach die verstrichene Zeit  
> ausgeben..  
> So hätten wir ein ungefähres Mass für Optimierungen..

direkt in das Timersteuerprogramm würde ich es nicht einbauen Aber auch sonst wollte ich es nicht allzu komfortabel machen, damit niemand auf die Idee kommt, das wäre ein richtiger Benchmark. Ich habe die Schleife aber jetzt mal mit einem return zum CCP abgespeichert und hier angehängt.

Autor: Leo C. ([rapid](#))

Datum: 19.07.2010 12:45

---

[Peter Sieg schrieb:](#)

```
> dec    hl
> jp     nz,hl    ; 256 x
```

Jörg hat ja schon geschrieben, daß hier Z\_Flag nicht gesetzt wird. Davon abgesehen wird die Schleife natürlich 65536 mal wiederholt. Da die innere Schleife ja auch schon gute 3 Sekunden braucht, kommst Du insgesamt auf gute 196 Kilosekunden, also über 54 Stunden. Ich glaube nicht, daß Du so lange warten willst.

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 19.07.2010 14:37

---

@ Leo C.

Das stimmt nicht ganz, denn die äussere Schleife wird NIE wiederholt. Das liegt einfach daran, dass das Zero-Flag von der nächstinneren Schleife noch gesetzt ist und der DEC HL Befehl keine Flags ändert.

Jörg

Autor: Leo C. ([rapid](#))

Datum: 19.07.2010 14:45

---

[Joerg Wolfram schrieb:](#)

> Das stimmt nicht ganz, denn die äussere Schleife wird NIE wiederholt.

Das stimmt natürlich.  
Ich hätte wohl schreiben sollen, daß die Schleife 65536 mal wiederholt werden **würde**, **wenn** "dec hl" das Z-Flag beeinflussen würde.

Autor: Peter Sieg (Gast)

Datum: 19.07.2010 15:53

Angehängte Dateien:

- [timer.asm](#) (2,1 KB, 126 Downloads) | [Codeansicht](#)
- [t.com](#) (512 Bytes, 96 Downloads)

---

Ja, es mußte ja sowas 'dummes' sein ;-)  
Danke!

Habe jetzt register c statt hl genommen.. und 10x reicht auch ;-)

bench:

```
        ld      a,startTimerCmd
    out  (TIMERCTL),a ; start
; loop starts here
    ld      c,10
11:    ld      b,0
12:    ld      a,0
13:    dec    a
    jp     nz,13      ; 256 x
        dec    b
    jp     nz,12      ; 256 x
    dec    c
    jp     nz,11      ; 10 x
; loop ends here
        ld      a,printTimerCmd
    out  (TIMERCTL),a ; print elapsed time
    ret                ; and done
```

Damit bekommt man mit:

<a href="#">AVR</a>	MHz	D	Zeit
88	20	0	035,999s
88	20	1	021,385s

Peter

Autor: Leo C. ([rapid](#))

Datum: 19.07.2010 17:12

---

[Peter Sieg schrieb:](#)

```
> Ja, es mußte ja sowas 'dummes' sein ;-)  
> Danke!  
>  
> Habe jetzt register c statt hl genommen.. und 10x reicht auch ;-)
```

```
> 11:    ld      b,0  
> 12:    ld      a,0  
> 13:    dec    a  
>        jp     nz,13      ; 256 x  
>        dec    b  
>        jp     nz,12      ; 256 x  
>        dec    c  
>        jp     nz,11      ; 10 x
```

Statt möglichst schnell zu zählen (und das möglichst oft, damit's nicht so schnell vorbei ist :), wäre es vielleicht besser, in die Schleife einige "interessante" Befehle zu legen. Z.B. "PUSH", "POP", "CALL", "RET", "LD (nnnn),HL", "EX (SP),HL". Also z.B. Befehle, die das RAM-Interface etwas mehr fordern. Oder einen Mix aus Befehlen, der möglichst praxisnah ist...

Peter, da Du doch was für MBASIC übrig hast: Mein Basic-Beispiel weiter oben, könnte man auch noch ausbauen.

```
> Damit bekommt man mit:
> AVR MHz D Zeit
> 88 20 0 035,999s
> 88 20 1 021,385s
Bei mir z.Zt:
8 20 ? 012.180s bb
8 20 - 010.811s exp
```

mit folgenden Pinbelegungen:

```
-----
|          Port D          |          Port B          |          Port C          |
+-----+-----+-----+
| 7 | 6 | 5 | 4 | 3 | 2 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 |
|          CS          | SCK DO DI          |          |          |          |
bb |A7 A6 A5          A8 OE |RAS A0 A1 A2 A3 A4 |CAS W D3 D2 D1 D0|
+-----+-----+-----+
|          CS |SCK DO DI          |          D3 D2 D1 D0|
exp|A7 A6          |A9 A8 WE OE CAS RAS|A5 A4 A3 A2 A1 A0|
-----
```

"exp" findet man in meinem [SVN](#) im "experimental" Zweig.

Autor: Peter Sieg ([petersieg](#))

Datum: 20.07.2010 14:09

---

ok. Na klar kömnte/sollte man ggf. verschiedenen Befehle etc. in der Schleife unterbringen.. aber was ist ein realer Mix? Und es ist ja kein wirklicher Benchmark, sondern nur ein Anhaltspunkt um andere Portbelegungen/Dramroutinen etc. zu 'messen'..

Und die MBASIC Version fand ich deshlab nicht ideal, weil man MBASIC benötigt..

Hier noch mal die D0 (Original Sprita-tm Portbelegung + Routinen)

und die D1 Version (Port C - Pin 1 + 4 umbelegt + neue Routinen)

(in z80.asm zu wählen durch: DRAM\_DQ\_ORDER == 0/1)

```
>-----
> |          Port D          |          Port B          |          Port C          |
> +-----+-----+-----+
> | 7 | 6 | 5 | 4 | 3 | 2 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 |
>D0|          CS          |SCK DO DI          |          |          |          |
> |A7 A6 A5          A8 OE |RAS A0 A1 A2 A3 A4 |CAS D1 D3 D2 W D0|
> +-----+-----+-----+
> | 7 | 6 | 5 | 4 | 3 | 2 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 |
>D1|          CS          |SCK DO DI          |          |          |          |
>bb|A7 A6 A5          A8 OE |RAS A0 A1 A2 A3 A4 |CAS W D3 D2 D1 D0|
>-----
```

D.h, du hast allein durch eine andere Portbelegung und geänderte Routinen,

eine Verbesserung von 021,385s (D1) zu 10,811s (exp( geschafft?!

```
> Damit bekommt man mit:
> AVR MHz D Zeit
> 88 20 0 035,999s
> 88 20 1 021,385s
>Bei mir z.Zt:
> 8 20 ? 012.180s bb
> 8 20 - 010.811s exp
```

mit folgenden Pinbelegungen:

Hmm.. sehe gerade, das D1 und dein bb identisch sind, von den Portbelegungen... hast du dann noch andere Routinen verwenden?? Denn du hast ja damit 12,180s erreicht (wohin ich nur 021,385s erreiche)??

Und das das am [ATmega8](#) liegen soll kann ich nicht wirklich glauben.. Hast du übrigens einen 8L verwendet?

```
>-----+-----+-----+-----+
> |          Port D          |          Port B          |          Port C          |
> +-----+-----+-----+-----+
> | 7 | 6 | 5 | 4 | 3 | 2 | 5 | 4 | 3 | 2 | 1 | 0 | 5 | 4 | 3 | 2 | 1 | 0 |
>D2|                                CS |SCK DO  DI                                |          D3 D2 D1 D0|
>ex|A7 A6                                |A9 A8 WE  OE  CAS RAS|A5 A4 A3 A2 A1 A0|
>-----+-----+-----+-----+
```

Der Gewinn D1 -> D2 ist allerdings nicht 'soo' groß..

Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 20.07.2010 14:42

---

[Peter Sieg schrieb:](#)

```
>> Damit bekommt man mit:
>> AVR   MHz   D   Zeit
>> 88    20    0   035,999s
>> 88    20    1   021,385s
>>Bei mir z.Zt:
>> 8     20    ?   012.180s   bb
>> 8     20    -   010.811s   exp
```

Bei mir:

```
AVR   MHz   D   Zeit
168   24    1   017,62s
```

muß mal den Code von Leo C. für die variante D1 (bb) testen.

Autor: Leo C. ([rapid](#))

Datum: 20.07.2010 14:46

---

[Peter Sieg schrieb:](#)

```
> ok. Na klar kömte/sollte man ggf. verschiedenen Befehle etc. in der
> Schleife unterbringen.. aber was ist ein realer Mix?
```

Darum habe ich mich bisher ja auch gedrückt. ;)

```
> Und es ist ja kein
> wirklicher Benchmark, sondern nur ein Anhaltspunkt um andere
> Portbelegungen/Dramroutinen etc. zu 'messen'..
```

Ja, aber Deine Schleife testet nicht mal das halbe DRAM-Interface. Geschweige, einen nennenswerten Teil des Interpreters.

```
> Und die MBASIC Version fand ich deshlab nicht ideal, weil man MBASIC
> benötigt..
```

Aber das ist ja vorhanden. Und es ist eine reale Anwendung.

```
> mit folgenden Pinbelegungen:
> Hmm.. sehe gerade, das D1 und dein bb identisch sind, von den
```

> Portbelegungen... hast du dann noch andere Routinen verwenden??  
> Denn du hast ja damit 12,180s erreicht (wohin ich nur 021,385s  
> erreiche)??

[Beitrag "Re: CP/M auf ATmega88"](#)

> Hast du übrigens einen 8L verwendet?

Eben nicht. Ich hatte ja auch schon mehrfach geschrieben, daß mein Mega8 mit zu niedriger Spannung und zu hoher Frequenz läuft.

> Der Gewinn D1 -> D2 ist allerdings nicht 'soo' groß..

[Beitrag "Re: CP/M auf ATmega88"](#)

Ich hatte mir allerdings auch etwas mehr erhofft.  
Zur Zeit bin ich gerade wieder an etwas dran, von dem ich mir nochmal einen Schub erhoffe. Details gibts aber erst, wenns wirklich funktioniert.

Übrigens habe ich inzwischen den Sektor (De-)Blocking-Algorithmus, der unnötiges, mehrfaches Lesen und Schreiben der SD-Karte vermeidet, drin. Performance-mässig bringt's leider auch noch nicht so viel, außer vielleicht bei Anwendungen, die viel und wahlfrei auf Disk schreiben. Aber Schreiben habe ich noch nicht wirklich getestet...

Autor: Leo C. ([rapid](#))

Datum: 20.07.2010 14:59

Angehängte Dateien:

- [mbasic.zip](#) (20,8 KB, 59 Downloads)

---

>> Und die MBASIC Version fand ich deshlab nicht ideal, weil man MBASIC  
>> benötigt..

> Aber das ist ja vorhanden. Und es ist eine reale Anwendung.

Da das MBASIC im Archiv von Sprite\_tm kaputt ist, habe ich hier mal eine funktionierende Version drangehängt. Außerdem habe ich gleich noch ELIZA dazugetan.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 20.07.2010 15:43

---

@Leo C.

Ich habe gerade die Versionen 28 und 29 getestet. Beide mit dem gleichen (Fehler)ergebnis.

Version 19 bringt noch als ersten Opcode von Adresse 2000:

opcode 31 decoded 0192 also korrekt.

Version 28 und 29 bringen:

opcode 31 decoded FFFF und dann startet das System neu.

Hast du was am Befehlsinterpreter geändert? Muß wieder ein neuer Schalter gesetzt werden?

Könntest du das cpm.bin nicht gleich wieder erstellen? Unter Windows habe ich keinen Z80 Asssembler gefunden der die Quellen übersetzt.

Autor: Peter Sieg ([petersieg](#))

Datum: 20.07.2010 17:39

---

[Joe G. schrieb:](#)

> muß mal den Code von Leo C. für die variante D1 (bb) testen.

Ja, denn irgendwoher muß der Unterschied:

```
>> 88    20    1  021,385s
```

```
>>Bei mir z.Zt:
```

```
>> 8     20    ?  012.180s  bb
```

ja herkommen.. denn die Portbelegung ist zw. D1 und bb identisch..

>Bei mir:

```
>AVR    MHz    D  Zeit
```

```
>168   24     1  017,62s
```

ja, +20% höherer Takt sollte die Reduzierung von ca. 21 auf 17s erklären.

Ich habe für timer.asm den tniasm.exe verwendet..(google). Der gibt aber bei ipl.asm+bios.asm noch eine Fehlermeldung bei Zeile 71/211? aus..

hatte noch keine Zeit da weiter rein zu schauen.. ist evtl. nur ne Kleinigkeit.. einen anderen Assembler für win32 habe ich auch noch nicht gefunden..

---

@Leo: Wenn die Optimierungen am Interpreter angekommen sind (nicht die schon vorgenommenen Fehlerbereinigungen), kann man ja gerne andere / weitere Befehle in die Benchschleife mit aufnehmen..

Z.Z habe ich eine D0 und eine D1 Variante hier am laufen.. Und die LR-Version

kann ich ggf. ja einfach komplett umbauen.. wenn es sich lohnt..

Ich suche noch 32MHz und 40MHz Quarze (wo gibt es die..?), dann kann ich mal

schauen ob die Realität der Hochrechnung entspricht (womit wir dann bei >1MHz bis an die ~2MHz Z80 sind..)

BTW: Mbasic hatte ich schon lange mal ein andere Version gezogen..

Muß jetzt mal die alten Basic Juwelen durchtesten.. Nim/23 und Bagels laufen einwandfrei.. stellen ich dann alle hier rein..

Peter

Autor: Leo C. ([rapid](#))

Datum: 20.07.2010 17:48

---

[Joe G. schrieb:](#)

> @Leo C.

> Ich habe gerade die Versionen 28 und 29 getestet. Beide mit dem gleichen > (Fehler)ergebnis.

> Version 19 bringt noch als ersten Opcode von Adresse 2000:

> opcode 31 decoded 0192 also korrekt.

> Version 28 und 29 bringen:

> opcode 31 decoded FFFF und dann startet das System neu.

Alle eingeecheckten Revisionen sind bei mir mindestens einmal gelaufen.

Einziger Unterschied zu Deiner Hardware dürfte der Prozessor sein.

Mega88, Megal68 kann ich nicht testen.

Alles in trunk sollte auf Deiner Hardware funktionieren.

(Ab Rev. 32 braucht man allerdings das zugehörige Bios.)

> Hast du was am Befehlsinterpreter geändert?

Einige kleine, lokale Optimierungen. Das Konzept ist immer noch das gleiche.

Hast Du einen Mega88 zum Testen?

> Muß wieder ein neuer Schalter gesetzt werden?

Nein.

> Könntest du das cpm.bin nicht gleich wieder erstellen?

Im [SVN](#) möchte ich eigentlich keine Binaries haben.

> Unter Windows

> habe ich keinen Z80 Asssembler gefunden der die Quellen übersetzt.

Die Assemblerquellen gibt es hier:

<http://savannah.nongnu.org/projects/z80asm/>

Vielleicht ist ja jemand so freundlich, und übersetzt das Ding für Windows.

**Autor: Peter Sieg (Gast)**

**Datum: 20.07.2010 18:14**

---

Hi.

Habe mich gerade daran erinnert, das auf alten Netzwerkkarten ja 25MHz Quarze sind ;-)

>Damit:

>AVR MHz D Zeit

>88 25 1 016,871s

Peter

**Autor: Feinmechaniker (Gast)**

**Datum: 20.07.2010 18:28**

---

[Peter Sieg schrieb:](#)

> Ich habe für timer.asm den tniasm.exe verwendet..(google). Der gibt

> aber bei ipl.asm+bios.asm noch eine Fehlermeldung bei Zeile 71/211?

> aus..

Danke für den Tip! Wenn Du jeweils den "end" Befehl streichst übersetzt der Compiler.

[Leo C. schrieb:](#)

> Einige kleine, lokale Optimierungen. Das Konzept ist immer noch das  
> gleiche.

> Hast Du einen Mega88 zum Testen?

Ich suche mal nach einem Mega88 zum testen. Übrigens ist mir aufgefallen, wenn der Schalter zum RAM-Test gesetzt ist läuft das System nicht stabil. Zork, MBASIC oder der SYS Befehl bringen oft (nicht immer) Op-Codefehler. Ohne RAM Test läuft alles Fehlerfrei. Sieht nach einem Timing Problem aus.

**Autor: Leo C. ([rapid](#))**

Datum: 20.07.2010 18:57

---

Peter Sieg schrieb:

> Habe mich gerade daran erinnert, das auf alten Netzwerkkarten ja 25MHz  
> Quarze sind ;-)

Dann sind meine Netzwerkkarten noch nicht alt genug? Die beiden 20MHz  
Quarze habe ich von dort.

Alte Grafikkarten sind eine gute Quelle für Oszillatoren. Auf meiner  
guten alten ET4000 sind 25.17, 28.322, 36, 44.9 und 62MHz. Ich weiß nur  
nicht, ob die Teile auch bei 3.3V laufen.

8 Stück 256x4MBit DRAMs sind übrigens auch drauf. Davon 4 sogar  
gesockelt.

Autor: Leo C. ([rapid](#))

Datum: 20.07.2010 19:03

---

Feinmechaniker schrieb:

> Übrigens ist mir  
> aufgefallen, wenn der Schalter zum RAM-Test gesetzt ist läuft das System  
> nicht stabil. Zork, MBASIC oder der SYS Befehl bringen oft (nicht immer)  
> Op-Codefehler. Ohne RAM Test läuft alles Fehlerfrei. Sieht nach einem  
> Timing Problem aus.

Das ist aber seltsam. Von dem RAM-Test sollte eigentlich keine  
Erinnerung  
übrig bleiben. Falls etwas nicht richtig initialisiert wird, sollte es  
eher umgekehrt sein. Ich habe den RAM-Test immer eingeschaltet.

Hast Du den MEMFILL\_CB-Schalter an? Ich finde den sehr nützlich, da ein  
ins Nirvana laufendes Programm sofort gestopt wird.

Autor: Leo C. ([rapid](#))

Datum: 26.07.2010 10:24

---

Inzwischen habe ich eine weitere Verdrahtung ausprobiert und dafür auch  
andere RAMs genommen (GM71C4256A-80, statt der schnelleren 3.3V Typen).  
Mit [ATmega8](#) gings problemlos, aber mit dem ATmega328P bekam ich die  
gleichen Probleme, wie Joe G. sie beschrieben hatte.

@Joe G.:

Geholfen hat ein weiterer Wartezyklus beim Lesen, in der [SVN](#)/trunk  
Version also so:

dram\_read:

```
cli
DRAM_SETADDR adrh, ~0, (1<<ram_ras), ~(1<<ram_a8), (1<<ram_oe)
cbi P_RAS,ram_ras
DRAM_SETADDR adrl, ~(1<<ram_ras),0, ~((1<<ram_oe)), (1<<ram_a8)
cbi P_CAS,ram_cas
cbi P_A8,ram_a8      ; 2 Takte
nop                  ; 1 Takt
in temp,P_DQ-2      ; PIN
sbi P_CAS,ram_cas

cbi P_CAS,ram_cas
andi temp,0x0f      ; 1 Takt
swap temp           ; 1 Takt
```

```

nop                ; 1 Takt
in temp2,P_DQ-2   ; PIN
andi temp2,0x0f
or temp,temp2

sbi P_OE,ram_oe
sbi P_CAS,ram_cas
sbi P_RAS,ram_ras
sei
ret

```

Nach den Datenblättern sollten 2 Takte (immerhin noch 83ns bei 24MHz) dicke reichen, aber bei 3.3V statt 5V dürfen wir ja froh sein, daß die RAMS überhaupt laufen.

Autor: Joe G. ([feinmechaniker](#))   
Datum: 26.07.2010 14:42

[Leo C. schrieb:](#)

> Geholfen hat ein weiterer Wartezyklus beim Lesen, in der [SVN](#)/trunk

Leider behebt es bei mir nicht den Fehler im Befehlsinterpreter, siehe

[Joe G. schrieb:](#)

> Version 19 bringt noch als ersten Opcode von Adresse 2000:  
> opcode 31 decoded 0192 also korrekt.  
> Version 28 und 29 bringen:  
> opcode 31 decoded FFFF und dann startet das System neu.

In Version 28 hast du auch die beiden neuen Funktionen dram\_read\_w, dram\_write\_w auch nicht eingebaut. In V29 habe ich die NOP's mal eingefügt, ohne Erfolg.

Autor: Joe G. ([feinmechaniker](#))   
Datum: 26.07.2010 15:31

@Leo C.

Ich habe den Fehler nochmal eingegrenzt. Es scheint wirklich ein Timing Problem mit dem RAM zu sein. V28 und V29 laufen bei 8MHz auf dem mega168 und mega8. Bei 20MHz bzw. 24 MHz tritt das Problem auf. Erstaunlicherweise läuft mein mega8 hier mit 3.3V und 24 MHz.

Autor: Leo C. ([rapid](#))  
Datum: 26.07.2010 15:37

[Joe G. schrieb:](#)

>> Version 19 bringt noch als ersten Opcode von Adresse 2000:  
>> opcode 31 decoded 0192 also korrekt.  
>> Version 28 und 29 bringen:  
>> opcode 31 decoded FFFF und dann startet das System neu.

Da ich die passende Hardware noch auf meinem Steckbrett habe, habe ich die Version 19 grade ausprobiert. Läuft bei mir mit ATmega328P. Für diesen Fehler habe ich auch keine Erklärung. Der Wert 0192 ist ja nur ein Tabelleneintrag (Index 31) in inst\_table. Es sieht so aus als ob der Interpreter bei Dir ins Leere (unprogramierte Flash) greift. Ab der Revision 28 muß die Tabelle auf einer durch 256 teilbaren Adresse liegen. Das müßte aber durch das .org (PC+255) & 0xff00 davor gewährleistet sein. Du könntest mal im Listing nachschauen, wo die

Tabelle bei Dir zu liegen kommt.

Mindestens ein Fehler ist aber drin. Beim Start sollte das Register "\_0" genullt werden. Z.B.:

```
start:
  ldi temp,low(RAMEND) ; top of memory
  out SPL,temp        ; init stack pointer
  ldi temp,high(RAMEND) ; top of memory
  out SPH,temp        ; init stack pointer

  clr _0
```

>  
> In Version 28 hast du auch die beiden neuen Funktionen dram\_read\_w,  
> dram\_write\_w auch nicht eingebaut.

Die Funktionen waren immer als "experimental" gekennzeichnet und abgeschaltet. Der Aufwand lohnt einfach nicht. Problem ist, daß ein 16-bit Wort auf der Grenze zwischen 2 Reihenadressen liegen kann.

>In V29 habe ich die NOP's mal  
> eingefügt, ohne Erfolg.

Ein weiterer Timing-Engpaß könnte noch in der Refresh-Routine liegen. Dort sind 2 auskommentierte nops drin. Wenn ich mich richtig erinnere, ist der 2. an der kritischeren Stelle.

Autor: Leo C. ([rapid](#))

Datum: 26.07.2010 15:59

---

[Joe G. schrieb:](#)

> @Leo C.  
> Ich habe den Fehler nochmal eingegrenzt. Es scheint wirklich ein Timing  
> Problem mit dem RAM zu sein. V28 und V29 laufen bei 8MHz auf dem mega168  
> und mega8. Bei 20MHz bzw. 24 MHz tritt das Problem auf.  
> Erstaunlicherweise läuft mein mega8 hier mit 3.3V und 24 MHz.

Das es mit dem Meaga8 besser geht, ist mir am Wochenende ja auch aufgefallen. Dir Porttreiber sind offensichtlich unterschiedlich (Ich habe gerade in die Datenblätter geschaut, und mein Verdacht hat sich bestätigt).

Man müßte mal ein Scope dranhängen, um zu sehen, was da vor sich geht.

Dein FFFF vs. 0192 Fehler ist damit aber nur schwer zu erklären. Der Opcode wird ja in beiden Fällen aus dem RAM richtig gelesen.

Du kannst das RAM-Timing natürlich weiter dehnen. Zuerst 1 oder 2 weitere nops vor den in-Befehlen. Dann vor/nach den anderen Signalflanken nops einbauen. Zum Schluß landet man vielleicht wieder bei den vielen nops, die Sprite\_tm schon drin hatte. ;-)

Oder beim Mega8 bleiben. Bisher kenne ich noch keinen Grund, der dagegen sprechen würde.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.07.2010 16:33

---

[Leo C. schrieb:](#)

```
> clr _0
```

Das war der Fehler! V29 läuft, vielen Dank für den Hinweis.  
Timer B zeigt nun 9.441 s mit einem mega8 und 24 Mhz

Autor: Leo C. ([rapid](#))

Datum: 26.07.2010 16:55

---

[Joe G. schrieb:](#)

```
>> clr _0
```

```
> Das war der Fehler! V29 läuft, vielen Dank für den Hinweis.
```

```
> Timer B zeigt nun 9.441 s mit einem mega8 und 24 Mhz
```

Bei mir hats komischerweise keinen Unterschied gemacht.

Ab Revision 51 ist der Fehler behoben. Allerdings sehe ich keinen Grund, gleich die HEAD Revision (53) zu nehmen.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.07.2010 17:00

---

ich teste mich mal nach oben ;-)

Bin zunächst gerade dabei mir ein neues BIOS zu erzeugen.

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)

[Re: CP/M auf ATmega88](#)

Autor: Peter Sieg (Gast)

Datum: 26.07.2010 17:22

---

@Leo C. + Joe G. Super!

@Joe G. planst du einen Update der über #define D\* = 1 erreichbaren Routinen,

um den Geschwindigkeitsvorteil nutzen zu können, der weiter oben aufgefallen war (sobald du weißt in welchen Routinen, was geändert wurde..)?

Hatte heute meinen ersten Arbeitstag nach dem Urlaub :-( :-(

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 27.07.2010 11:28

---

@Leo C.

Revision 51 läuft.

@alle

um sich unter Windows ein Bios zu erzeugen sollte man ein DD "für Windows" benutzen welches den Befehl conv=sync kennt. Hier in Link zu UNIX-Tools unter Windows, da ist auch das entsprechende DD dabei.

<http://sourceforge.net/projects/unxutils/files/unx...>

[Peter Sieg schrieb:](#)

```
> @Joe G. planst du einen Update der über #define D* = 1 erreichbaren
```

```
> Routinen
```

Ich habe sie genau so verwendet wie Leo C. sie bereitgestellt hat, bis auf das zusätzliche Einfügen von "clr \_0" in der V29. Aber ab V51 ist der Befehl schon mit drin. Ab V51 mußt du jedoch das neue Bios nutzen.

Autor: Peter Sieg (Gast)  
Datum: 27.07.2010 14:01

---

@Joe G.: In wieweit wirst du die V51 in deine Version übernehmen..?

Und das neue Bios von Leo C. hatte ich schon (hatte er weiter oben als cpm.bin zur Verfügung gestellt..).. oder gibts da schon ein neueres..?

Gruß Peter

Autor: Joe G. ([feinmechaniker](#))   
Datum: 27.07.2010 14:12

---

[Peter Sieg schrieb:](#)

> Und das neue Bios von Leo C. hatte ich schon (hatte er weiter oben als  
> cpm.bin zur Verfügung gestellt..).. oder gibts da schon ein neueres..?

ja, gibt es, siehe hier:

- \* cpm/bios.asm:
  - New disk I/O interface.
  - 16 bit wide track numbers
  - sectran competed. Unused translation table removed.
- \* cpm/ipl.asm:
  - New disk I/O interface.

[Peter Sieg schrieb:](#)

> @Joe G.: In wieweit wirst du die V51 in deine Version übernehmen..?

Wenn es keine Probleme damit gibt, vollständig. Es wäre aber schön, wenn sie noch von anderen getestet wird.

Autor: Peter Sieg ([petersieg](#))  
Datum: 27.07.2010 17:12

---

Man(n) .. ;-)

Mit der Geschwindigkeit kann ich z.Z nicht mithalten.. :-)

Könntet ihr dann bitte ipl.asm, bios.asm und avrcpm.asm hier einhängen.. dann sichere ich erstmal den aktuellen Stand als V1, assembliere dann alles und kann dann auch testen..

BTW: Was sind die Änderungen/Vorteile insbesondere von 16 bit wide track numbers..?

Peter

Autor: Leo C. ([rapid](#))  
Datum: 27.07.2010 17:20

---

[Peter Sieg schrieb:](#)

> Man(n) .. ;-)

> Mit der Geschwindigkeit kann ich z.Z nicht mithalten.. :-)

> Könntet ihr dann bitte ipl.asm, bios.asm und avrcpm.asm hier einhängen..

Warum klickst Du denn nicht einfach hier drauf?

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

> dann sichere ich erstmal den aktuellen Stand als V1, assembliere dann  
> alles  
> und kann dann auch testen..  
>  
> BTW: Was sind die Änderungen/Vorteile insbesondere von 16 bit wide track  
> numbers..?

Warst Du das nicht, der größere Disk-Formate wollte? ;-)

## Größere Diskformate/Partitionen

Autor: Leo C. ([rapid](#))

Datum: 27.07.2010 17:41

---

Da ich an dem Thema sowieso grade wieder dran bin...

[Peter Sieg schrieb:](#)

> BTW: Was sind die Änderungen/Vorteile insbesondere von 16 bit wide track  
> numbers..?

Mit 8-Bit Nummern kommt man maximal bis 8MByte.  
(128 Byte per Sector) x (256 Sectors per Track) x (256 Tracks)

Bei SD-Karten im GB-Bereich ist das etwas wenig.

Jetzt mal ein paar Fragen zu größeren Diskformaten:

Hat sich darüber schon jemand Gedanken gemacht?  
Hat vielleicht jemand schon ein Konzept?  
Oder kennt jemand einen hier brauchbaren Standard / eine fertige Lösung?

Vorschläge?

Meine eigenen Gedanken:

Fernziel vielleicht Disk-Images auf FAT Dateisystem lesen/schreiben.  
(Dafür wird man wahrscheinlich einen Megal68 oder größer brauchen, da ja  
auch noch die Z80 Opcodes ins Flash müssen).

Zwischenschritt:

Die Partitionstabelle von der SD-Karte lesen.  
Erste (oder einzigste) Partition als CP/M partition nutzen.  
Im Bootsektor der Partition die CP/M-Diskparameter ablegen.  
Die Tabelle passt mit dem IPL in die ersten 128 Byte.

Was meint Ihr?

Autor: Peter Sieg ([petersieg](#))

Datum: 27.07.2010 19:00

---

[Leo C. schrieb:](#)

> Warum klickst Du denn nicht einfach hier drauf?  
> <http://cloudbase.homelinux.net/viewvc/avr-cpm/trunk/>  
>  
> Warst Du das nicht, der größere Disk-Formate wollte? ;-)

Na dann schaue ich mal in den 'Kofferraum' (trunk) ;-)

Werde dann mal versuchen für D1 eine V53 komplett zu übersetzen..

Ja, ich wars ;-) Aber den Verhältnissen entsprechend des Konzeptes..  
eine  
EINFACHE Implementierung!! Nichts zu kompliziertes.. Hey, wir reden hier  
über CP/M.. wer sollte selbst bei 8MB großen 'Disketten' ohne  
Unterverzeichnisse noch die Übersicht behalten.. Aktuell sind die  
diskimages sowas um die 250kb.. da passt sogar schon der BDS-C Compiler  
drauf.. also 512kb - 1MB sollte reichen.. besser wären dann mehrere  
Disketten(-images) auf einer SD Karte.. (A:, B:, C: ...)

Just my 2 cents.

Peter

Autor: Leo C. ([rapid](#))

Datum: 27.07.2010 19:30

---

[Peter Sieg schrieb:](#)

> EINFACHE Implementierung!! Nichts zu kompliziertes..

Ja eben.

> Hey, wir reden hier  
> über CP/M.. wer sollte selbst bei 8MB großen 'Disketten' ohne  
> Unterverzeichnisse noch die Übersicht behalten.. Aktuell sind die

Der Punkt ist, daß kleine Images auch nicht einfacher zu implementieren  
sind, als große. Und warum dann so unendlich viel Platz  
verschwenden(?/!)

> diskimages sowas um die 250kb.. da passt sogar schon der BDS-C Compiler  
> drauf.. also 512kb - 1MB sollte reichen..

Wenn ich das mit der Partitionstabelle so wie oben angedeutet mache,  
kannst Du ja eine 512K Partition für CP/M anlegen. Und den Rest  
anderweitig nutzen.

> besser wären dann mehrere  
> Disketten(-images) auf einer SD Karte.. (A:, B:, C: ...)

Dann eben 2 Partitionen, oder 3... Bei 4 wird aber Schluß sein, weil  
erweiterte Partitionen will ich mir nicht antun.

Autor: Peter Sieg ([petersieg](#))

Datum: 27.07.2010 20:07

---

Soo.. hat ein bißchen gedauert.. mußte erstnoch obiges dd  
herunterladen.. und mit dem 'alten' dd unter admin Rechten dann das  
diskimage schreiben (Vista)..

CPM on an AVR, v1.0  
Initing mmc...  
Testing RAM: fill...wait...reread...

Ok, CPU is live!  
ipl

62k cp/m vers 2.2

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>t b
```

Timer running. Elapsed: 007.573s.

A>

Na.. 7.573s allerdings bei 30MHz ;-)  
Target war [ATmega88](#).

Na, wenn das keine super Arbeit ist Leo C. ;-)  
Klasse!!

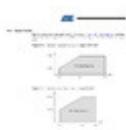
Und VIELEN Dank!

Gruß Peter

Autor: Leo C. ([rapid](#))

Datum: 27.07.2010 20:37

Angehängte Dateien:



- [SpeedGrades.pdf](#) | [anzeigen](#)  
70,7 KB, 151 Downloads

---

[Peter Sieg schrieb:](#)

> Na.. 7.573s allerdings bei 30MHz ;-)

Mir wird schwindelig.

Wenn ich mich nicht verrechnet habe, hast Du bereits bei ca. 13,5MHz den sicheren Bereich verlassen.

Wo sind meine Herztropfen.

Autor: Peter Sieg (Gast)

Datum: 28.07.2010 08:14

---

[Leo C. schrieb:](#)

> Wenn ich mich nicht verrechnet habe, hast Du bereits bei ca. 13,5MHz den  
> sicheren Bereich verlassen.

Haha.. Dann sind wir schon drei (der dritte ist jemand aus dem F64, der einen [ATmega88](#) als SID Ersatz mit 40MHz laufen läßt) ;-)

@Joe G: Mein Vorschlag wäre es, die aktuelle Version um MK [SVN](#) als V1 zu belassen (inkl. ipl+bios.asm) und die R53 von Loe C. als aktuelle Version ins MK [SVN](#) zu spiegeln.. sodaß wir im MK [SVN](#) immer etwas größere Abstände zw. Versionen bekommen..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.07.2010 09:20

---

[Leo C. schrieb:](#)

> Wenn ich mich nicht verrechnet habe, hast Du bereits bei ca. 13,5MHz den  
> sicheren Bereich verlassen.

Das ist ein Fehler im Datenblatt, die lineare Funktion knickt nicht bei 4.5V ab, Peter hat nur verregssen 12V an den [AVR](#) anzulegen. Nicht auszudenken mit welcher Frequenz er bei 220V läuft! Ich vermute mal mit 50Hz ;-)

[Peter Sieg schrieb:](#)

> Mein Vorschlag wäre es, die aktuelle Version um MK [SVN](#) als V1 zu  
> belassen (inkl. ipl+bios.asm)

Ich würde die Version V29 von Leo C. incl. ipl+bios als V1 eindampfen. Mit der derzeitigen Hardware dürfte es damit für weitere Nachbauinteressenten keine Probleme geben. R52 nehme ich als aktuelle Testversion auf.

[Peter Sieg schrieb:](#)

> und mit dem 'alten' dd unter admin Rechten dann das  
> diskimage schreiben

Eignet sich das "neue" DD nicht zum Diskimage schreiben? Ich habe dazu bisher auch nur das "alte" verwendet.

Autor: Peter Sieg (Gast)

Datum: 28.07.2010 11:50

---

@Joe G. Prima. Ich habe hier jetzt wie schon geschrieben, die R53 von Leo C im Einsatz.. die unterstützt aber wohl nicht mehr das ursprüngliche HW Layout (sind ja nur die 2 Pins geändert..). Ich habe mir erlaubt diese Version auch mal wieder dem Entwickler Sprite\_tm zukommen zu lassen.. Er freut sich sicher über solche Updates von Zeit zu Zeit..

Bei 220V erwarte ich auch 50Hz.. nachdem sich der Rauch verzogen hat.. ;-)

Aber ich plane auch noch einen Test mit 40MHz.. da aber wohl mit einem Quarzoszi, da es wohl keine Grundton Quarze für diese Frequenz mehr gibt..

Da werde ich dann meinen LR Aufbau umbauen müssen.. ;-)

[Joe G. schrieb:](#)

> Eignet sich das "neue" DD nicht zum Diskimage schreiben? Ich habe dazu  
> bisher auch nur das "alte" verwendet.

Anscheinend nicht.. ich habe es damit nicht hinbekommen.. evtl. findet mal jemand eines, das beides kann..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.07.2010 12:19

---

@alle

Der aktuelle Bearbeitungsstand liegt nun wieder auf dem [SVN](#)-Repositor.  
Unter V 1.0 findet ihr den letzten Stand der noch mit der alten Hardware  
läuft. Bei den folgenden Versionen (Verzeichnis /Work) müssen die zwei  
Pins, wie weiter oben schon beschrieben, getauscht werden.

**Autor:** Zwölfliter (Gast)

**Datum:** 28.07.2010 12:36

---

Laufen denn auch schon große Programme wie Wordstar?

**Autor:** Joe G. ([feinmechaniker](#)) 

**Datum:** 28.07.2010 12:40

---

Zwölfliter schrieb:

> Laufen denn auch schon große Programme wie Wordstar?

Derzeit laufen alle für den 8080 codierten Programme, egal ob groß oder  
klein. Keine Ahnung ob es Wordstar für den 8080 gab, ich kenne es nur  
für Z80.

**Autor:** Peter Sieg (Gast)

**Datum:** 28.07.2010 12:56

---

@Leo C. Bezüglich mehrere Disketten (A: , B:, ...) wäre mein erster  
Ansatz gewesen, alle diskimages mit fester Größe ('disksize') direkt mit  
z.b dd hintereinander zu hängen und dies dann so auch per dd auf die SD  
Karte zu bringen.. Im [AVR](#) wäre dann ein 'Offset' zu halten, der beim  
Start=0 ist und  
beim Wechseln von A: nach z.B B: dann entsprechend auf 1 wechselt..  
(C=2,..)  
Beim Zugriff auf die SD Karte ist dieser Offset dann entsprechend zu  
berücksichtigen (Offset \* disksize = Block 0 der aktuellen 'Diskette')..

Peter

**Autor:** Leo C. ([rapid](#))

**Datum:** 28.07.2010 13:16

---

Peter Sieg schrieb:

> @Leo C. Bezüglich mehrere Disketten (A: , B:, ...) wäre mein erster  
> Ansatz gewesen, alle diskimages mit fester Größe ('disksize') direkt mit  
> z.b dd hintereinander zu hängen und dies dann so auch per dd auf die SD  
> Karte zu bringen..

zu spät ;-)

Ich habe den code, um CPM-Laufwerke (A: .. D:) von Partitionen zu lesen,  
fast fertig. Z.Zt. kämpfe ich allerdings mit der SD-Karte/dem  
MMC-Treiber,  
weil die Karte, die ich zum Testen nehmen wollte, Schwierigkeiten macht.  
Werde jetzt erst mal meine alte Karte umpartitionieren.

Vorteil der Partitionen ist vor allem, daß man den nicht für  
CPM-genutzten Platz für andere Zwecke verwenden kann. Ich habe z.B. auf  
jeder Karte ein [MP3](#)-Verzeichnis fürs Auto.

Unter Linux ist es kein Problem, Partitionen mit beliebigen IDs  
anzulegen. Für CP/M ist 0x52 reserviert, und die möchte ich dann auch  
nehmen.

Ich weiß nicht, ob es für Windows auch gängige Partitionierungstools gibt, die das können. Aber notfalls gehts ja mit dd. Ist dann auch nicht umständlicher als ohne Partitionen.

Autor: Leo C. ([rapid](#))

Datum: 28.07.2010 13:23

---

[Joe G. schrieb:](#)

> Keine Ahnung ob es Wordstar für den 8080 gab, ich kenne es nur  
> für Z80.

Seltsam, Wordstar ist ja imho **das** CP/M-Programm schlechthin. Ich bin eigentlich davon überzeugt, daß alle CP/M-80 Versionen von Wordstar auch auf 8080 laufen. Es sei denn, jemand hat in der Printer patch area Z80-Code verwendet (Oder so ähnlich).

Ich bin noch nicht dazu gekommen, einen Wordstar an meine Terminalemulation anzupassen. Der Wordstar-Installer, den ich ausprobiert habe, hatte kein VT10x und kein ANSI im Angebot. Gelaufen ist er aber.

## Wordstar läuft

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.07.2010 14:11

Angehängte Dateien:



[ws30.jpg](#)

105,7 KB, 444 Downloads

---

Ich hab es gerade mal ausprobiert. WS 3.0 läuft siehe Bild.

## Der Entwickler möchte Leo C. und Joe G. im Source erwähnen

Autor: Peter Sieg (Gast)

Datum: 28.07.2010 14:48

---

@Leo C. + Joe G. -> das solltet ihr entscheiden:

Hi again,

Just looked through the source. Seems the two guys made some truly non-trivial improvements in the code, so I want to mention them on the

site and in the copyright. Do you know if it's OK if I just put down 'Leo C.' and 'Joe G.' as their names, or do they want another moniker in there?

Jeroen

On Wed, 28 Jul 2010, Peter Sieg wrote:

```
> Hi Jeroen.
>
> Please find attached the actual version of your avrcpm project.
> 8080 opcodes fixed, warm start fixed, speed up of app. 100%,
> target are: atmega8,88,168 and 328.
> The improvements are not from me, but rather Leo C. and Joe G. from
> the german mikrocontroller forum!
> (2 pins have to be exchanged:
> Pin 24 - PC1 - /WE
> Pin 27 - PC4 - IO3)
>
> Many thanks, Peter
```

Am besten mir/hier kurz bescheid geben.. dann gebe ichs weiter oder auch gerne natürlich direkt an:

jeroen (at) spritesmods.com

Peter

## Weiter mit Partitionen

Autor: Leo C. ([rapid](#))

Datum: 28.07.2010 23:09

---

Zum Testen habe ich eine etwas chaotische Partitionierung erzeugt:

```
leo@cb:~$ /sbin/fdisk -luc /dev/sdb
```

```
Disk /dev/sdb: 1009 MB, 1009254400 bytes
32 heads, 61 sectors/track, 1009 cylinders, total 1971200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0001fc66
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		8192	1890303	941056	6	FAT16
/dev/sdb2		1890304	1906687	8192	52	CP/M
/dev/sdb3		2048	8191	3072	52	CP/M

Partition table entries are not in disk order

```
leo@cb:~$
```

Die erste CP/M-Partition (A:) liegt also ganz hinten auf der SD-Karte.

Und das kommt raus:

```
A>CPM on an AVR, v1.0
Testing RAM: fill...wait...reread...
Initing mmc...
CP/M partition at: 1890304, size: 8192KB.
CP/M partition at: 2048, size: 3072KB.
```

Ok, CPU is live!  
ipl

62k cp/m vers 2.2

A>stat a:dsk:

```
A: Drive Characteristics
1944: 128 Byte Record Capacity
243: Kilobyte Drive Capacity
64: 32 Byte Directory Entries
64: Checked Directory Entries
128: Records/ Extent
8: Records/ Block
26: Sectors/ Track
2: Reserved Tracks
```

A>stat b:dsk:

```
B: Drive Characteristics
1944: 128 Byte Record Capacity
243: Kilobyte Drive Capacity
64: 32 Byte Directory Entries
64: Checked Directory Entries
128: Records/ Extent
8: Records/ Block
26: Sectors/ Track
2: Reserved Tracks
```

A>

Das Disk-Format is noch das alte mit nur 243 Kilobyte. Die zugehörigen Tabellen sind jetzt noch fest im BIOS und werden demnächst dynamisiert.

Wie immer auf:

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

Neues BIOS ist auch notwendig, damits klappt.

**Autor: Peter Sieg (Gast)**

**Datum: 29.07.2010 08:18**

---

@Leo C.: Frage: diskimages werden nach wie vor mit z.B dd aufgespielt?  
Also z.B dd if=diskimage of=/dev/sdb2 bs=512 ...(unter Linux)..?

Super! Man(n).. selbst ein Vollzeit-Entwickler könnte da nicht schneller dran arbeiten.. ;-)) Da komme ich ja mit dem wechseln der Versionen nicht nach..

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 29.07.2010 13:08**

---

[Peter Sieg schrieb:](#)

> @Leo C.: Frage: diskimages werden nach wie vor mit z.B dd aufgespielt?  
> Also z.B dd if=diskimage of=/dev/sdb2 bs=512 ...(unter Linux)..?

Genau.

Übrigens,  
für linux gibt es sfdisk, mit dem man Partitionstabellenänderungen  
scripten kann, zB. im Makefile. Mit "sfdisk -c /dev/sdb 2 52" kann man  
z.B. die Partitions-ID von Partition 2 auf 52 einstellen.

Autor: Peter Sieg ([petersieg](#))

Datum: 29.07.2010 15:40

---

Soo.. klappt!

CPM on an AVR, v1.0

Testing RAM: fill...wait...reread...

Initing mmc...

CP/M partition at: 401625, size: 8032KB.

CP/M partition at: 417690, size: 8032KB.

CP/M partition at: 433755, size: 8032KB.

Ok, CPU is live!

ipl

62k cp/m vers 2.2

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>b:

B>dir

```
B: TTT2        COM : OTHELLO       COM : MMIND        COM : ED          COM
B: LOAD        COM : STONE         COM : PIP          COM : STAT        COM
B: SUBMIT      COM : XSUB          COM
```

B>

...allerdings Partitionierung, Typ=52 setzen nur mit Linux Live-CD und  
cfdisk o.ä. Tools.. Auch bespielen per dd ging nur unter Linux, da das  
Windows dd die Typ=52 Partitionen nicht kennt/anzeigt..  
=> Wer kennt ein s/cfdisk unter Windows..?

Ansonsten wiederum Super!

Gruß Peter

Autor: Leo C. ([rapid](#))

Datum: 29.07.2010 16:07

---

[Peter Sieg schrieb:](#)

```
> ...allerdings Partitionierung, Typ=52 setzen nur mit Linux Live-CD und
> cfdisk o.ä. Tools.. Auch bespielen per dd ging nur unter Linux, da das
> Windows dd die Typ=52 Partitionen nicht kennt/anzeigt..
```

Etwas bequemer als PC mit Live-CD booten ist dann vielleicht dd und Binär/Hex-Editor.

```
dd if=/dev/sdb of=mbr-b.bin bs=512 count=1
```

mbr-b.bin editieren.

Der Aufbau der Partitionstabelle ist im Wikipedia-Artikel gut dargestellt:

<http://de.wikipedia.org/wiki/Partitionstabelle>

Der Partitionstyp steht also für die erste Partition auf 1C2. Für die nächsten 3 Partitionen entsprechend auf 1D2, 1E2, 1F2.

Dieses Byte auf 52 ändern und die Datei zurückschreiben:

```
dd of=/dev/sdb if=mbr-b.bin bs=512 count=1
```

Fertig.

> => Wer kennt ein s/cfdisk unter Windows..?

Ich hab auch mal kurz geschaut, ob es z.B. (g)parted für Windows gibt und bin gleich fündig geworden. Hurra... Linux-Live-CD!

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 29.07.2010 18:07**

---

@Leo C.

Meine SD Karte ist unter Linux etwa so partitioniert:

```
/dev/sdh1 200MB FAT16
/dev/sdh2 8MB CP/M
/dev/sdh3 8MB CP/M
/dev/sdh4 8MB CP/M
(app. 30MB unused)
```

Diese Karte unter dem windows dd zeigt NUR die FAT16 Partition an!  
Wenn ich da ein dd if=diskimage of=\\...die einzige angezeigte Part...  
bs=512 mache, läuft avrcpm damit als A:

Wie gesagt, die Typ=52 = CP/M Partitionen werden erst gar nicht von dem Tool unter Windows angezeigt.. ignoriert als wenn es sie gar nicht gäbe..

Daher vermute ich, das wenn man vom Type CP/M hin zu FAT16 auch die anderen Partitionen eben wieder zugreifbar macht.. und dann auch wieder unter Windows mit dd beschreiben kann..

diskpart unter Windows gibt mir:

Datentr	###	Status	Größe	Frei	Dyn	GPT
0		Online	149 GB	0 B		
1		Online	244 MB	23 MB		
2		Kein Mediu	0 B	0 B		
3		Kein Mediu	0 B	0 B		
4		Kein Mediu	0 B	0 B		
5		Kein Mediu	0 B	0 B		
6		Kein Mediu	0 B	0 B		
7		Kein Mediu	0 B	0 B		
8		Kein Mediu	0 B	0 B		

DISKPART> select disk 1

Datenträger 1 ist jetzt der gewählte Datenträger.

DISKPART> list part

Partition ###	Typ	Größe	Offset
-----	-----	-----	-----
Partition 1	Primär	196 MB	32 KB
Partition 0	Primär	8033 KB	196 MB
Partition 0	Primär	8033 KB	204 MB
Partition 0	Primär	8033 KB	212 MB

Ich persönlich habe kein wirkliches Problem mit einer Linux Live-CD..  
Finde aber es aber schade, das wir nun ohne Linux - allein mit Windows  
Mitteln - nicht mehr auskommen.. wobei wir ja noch 'das Tool' evtl. noch  
finden.. (diskpart ist ein Krampf..)

Gruß Peter

Autor: Peter Sieg ([petersieg](#))

Datum: 29.07.2010 18:31

---

[Leo C. schrieb:](#)

> Partitionieren kann man doch noch immer unter Windows, oder?  
> Warum also nicht mit dem Windows Partitionierer die Karte nach Wunsch  
> aufteilen, und den CP/M-Partitionen vorläufig einen Typ geben, den  
> Windows kann.  
>  
> Dann das Windows fdisk oder Festplattenverwaltungswerkzeug, oder wie  
> auch immer das Teil unter NT, XP,... heißt, verlassen und anschließend  
> dd und Hex-Editor?

Das könnte so gehen.. aber das ist nicht einfacher als gleich eine  
Live-CD zu nehmen.. (just my 2 cents)..

Peter

Autor: Peter Sieg (Gast)

Datum: 29.07.2010 19:27

---

@Leo C.: Na, nu lass mal gut sein :-) Windows User, die keine Linux  
Live-CD nehmen wollen, haben ja durch den sehr guten Fall-Back  
Mechanismus immerhin das was alle vorher hatten.. nämlich LW A: =  
gerechte Strafe würden jetzt Linux Nutzer sagen - alle anderen haben  
halt bis zu 4 Laufwerke.. ;-)  
=> kann ICH mit leben..

Trotzdem suche ich weiter, das unter Windows noch hinzubekommen.. ok?  
;-)

(Cygwin hat cfdisk.. vergesst es!! Bloß die Finger weg davon.. erkennt  
LW/Part. ebenfalls nicht.. und trägt sich noch nichtmal sauber ein zum  
deinstallieren..)

Bin übrigens am WE nicht/wenig online..

Peter

## Update Wunschliste

Autor: Peter Sieg ([petersieg](#))

Datum: 29.07.2010 21:35

---

Um mal etwas von der Diskussion der Partitionierung unter Windows abzulenken.. ;-)

Aus meiner Sicht könnten nächste Schritte sein:

\* Diskettenimage von 243kb vergrößern Richtung 1-xMB.

(Den riesigen Platz kann man ohne Unterverzeichnisse CP/M-like über andere

USER x etwas übersichtlicher gestalten.. und die cpmttools erlauben die Befüllung unterschiedlicher Userbereiche)

Da hat man dann auch mit nur einer 'Diskette' was davon ;-)

\* Eine Möglichkeit auch ohne cpmttools einzelne Dateien zwischen der CP/M und

der Host Welt auszutauschen. Das N8VEM Projekt verwendet dazu klassisch

ein XModem (XM.COM) Programm.. das wäre sicher auch für uns eine gute Wahl.

Andere Ideen..?

\* Dann kämen ggf. die Z80 Opcodes dran...

\* Und wie wäre es mit einem passenden, seriellen Terminal mit BAS/VGA/LCD als

Bildschirm und PS/2 Tastatur ala PropTerm.. aber auch auf [AVR](#) Basis..?

Peter

Autor: Leo C. ([rapid](#))

Datum: 29.07.2010 22:13

---

Peter Sieg schrieb:

> Um mal etwas von der Diskussion der Partitionierung unter Windows  
> abzulenken.. ;-)

Oh, heute komme ich gar nicht mehr zum Essen.

> Aus meiner Sicht könnten nächste Schritte sein:

> \* Diskettenimage von 243kb vergrößern Richtung 1-xMB.

Klar, das muß kommen. Mit **einem** festen Format wärs auch schnell gemacht.

> (Den riesigen Platz kann man ohne Unterverzeichnisse CP/M-like über  
> andere

> USER x etwas übersichtlicher gestalten.. und die cpmttools erlauben die  
> Befüllung unterschiedlicher Userbereiche)

Leider werden die Userbereiche von CP/M 2.2 sehr schlecht unterstützt.  
Mit CP/M 3 oder einer der 2.2 Alternativen wäre man da besser dran.

> \* Eine Möglichkeit auch ohne cpmttools einzelne Dateien zwischen der CP/M  
> und

> der Host Welt auszutauschen. Das N8VEM Projekt verwendet dazu  
> klassisch

> ein XModem (XM.COM) Programm.. das wäre sicher auch für uns eine gute  
> Wahl.

Sollte kein Problem sein. Einfach mal ausprobieren.

> Andere Ideen..?

CP/NET? ;-)

> \* Dann kämen ggf. die Z80 Opcodes dran...

Irgendwann sollte man das mal angehen, bevor das Flash mit anderen Sachen voll ist. ;) Der Interpreter wird dadurch aber auf jeden Fall langsamer werden. Ein Grund, warum ich das bisher nicht angegangen bin. Wenn da nicht das ein oder andere wichtige Programm wäre, das Z80 voraussetzt (mir fällt gerade nur Turbopascal ein), würde ich es sogar ganz bleiben lassen.

> \* Und wie wäre es mit einem passenden, seriellen Terminal mit  
> BAS/VGA/LCD als  
> Bildschirm und PS/2 Tastatur ala PropTerm.. aber auch auf [AVR](#) Basis..?

Anderes Projekt. :-)

Ansonsten will ich noch eine RAM-Disk einbauen. Beim Originalaufbau hat man dafür zwar nur 64KByte, aber man könnte schonmal CCP+BDOS dort deponieren und beim Warmstart von dort laden. Außerdem habe ich bei meinem Aufbau jetzt 2 4x4MBit Chips drin. Das gibt dann eine RAM-Disk mit knapp 4 MByte. :)

## VGA-Term

Autor: hschuetz (Gast)

Datum: 30.07.2010 08:58

---

Hallo Leute,  
das Modul gibts für 28,49 Euro, hat alles on Board, auch eine 3,3V Stromversorgung (200mA)...

<http://www.watterott.com/de/Schnittstellen/Video-VGA>

Gruß

Hans- Werner

Autor: Peter Sieg (Gast)

Datum: 30.07.2010 09:59

---

[hschuetz schrieb:](#)

> das Modul gibts für 28,49 Euro, hat alles on Board, auch eine 3,3V  
> Stromversorgung (200mA)...

> <http://www.watterott.com/de/Schnittstellen/Video-VGA>

@hschuetz: Super, Danke! Genau was ich gesucht habe.. gleich mal eins bestellt.. ;-)

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.07.2010 10:35

---

Es gab Anfragen ob noch Leiterplatten verfügbar wären. Leider nein. Bei genug Interessenten könnte natürlich eine neue Serie mit den entsprechenden Änderungen aufgelegt werden. Mir fällt da spontan ein:  
- geänderte Pinbelegung am RAM (RAM Disk)  
- Video-VGA siehe weiter oben.

- weitere Vorschläge ???

**Autor: Dagobert (Gast)**

**Datum: 30.07.2010 10:51**

---

Das µVGA-Modul gibt es auch hier:

<http://www.derelektronikershop.de//themes/kategori...>

**Autor: Peter Sieg (Gast)**

**Datum: 30.07.2010 11:16**

---

Joe G. schrieb:

> Es gab Anfragen ob noch Leiterplatten verfügbar wären. Leider nein.  
> Bei genug Interessenten könnte natürlich eine neue Serie mit den  
> entsprechenden Änderungen aufgelegt werden.

@Joe G. Ja, die beiden Pin's am Ram sollte mindestens aufgrund der heutigen Erfahrungen getauscht werden.. ;-) Größerer Ram./ Ramdisk.. warten, was Leo C. hier zaubert.. beim Thema Terminal wirds etwas schwieriger..

Was ich bisher so gefunden hatte:

[Beitrag "AVR ASCII Video Terminal - 40 x 25 - BAS Signal"](http://www.serasidis.gr/circuits/AVR_VGA/avr_vga.htm)

[http://www.serasidis.gr/circuits/AVR\\_VGA/avr\\_vga.htm](http://www.serasidis.gr/circuits/AVR_VGA/avr_vga.htm)

[Beitrag "Einfacher Low Cost LCD Controller für 320x240 LCD im Textmodus"](http://www.shaels.net/index.php/propterm)

<http://www.shaels.net/index.php/propterm>

Obigen 3 fehlt noch die PS/2 Tastatur inkl. senden zum [AVR](#).. Code dafür ist

sicher verfügbar.. also entweder noch mit einbauen oder separat im weiteren [AVR](#) aufbauen.. ideal würde ich auch die Realisierung mit [ATmega88](#)/168 ansehen.. gleicher Baustein.. kann locker mir 20MHz - sogar 30MHz.. ein Programmierer..

Propterm ist von dem was es kann sicher ideal.. aber ganz andere HW Welt..

---

Wenn daran überhaupt genügend Interesse besteht (an einem passenden Selbstbau-Terminal..) würde ich einen separaten Thread dazu vorschlagen..

Und ein solches Terminal sollte ersteinmal entwickelt und auf Steckbrett/LR aufgebaut werden, um dann zu entscheiden, ob es eine separate Platine/abtrennbare Platine oder eine gemeinsame mit [AVR](#) CP/M werden soll.. ich werde ersteinmal das uVGA testen.. da es auch 3,3V zur Verfügung stellt.. fast schon ideal.. und für ca. 30€ kaum zu schlagen..

In jedem Fall wäre ich bei einer Neuauflage mit 1 Platine dabei ;-)

Dann sollte man auch mal prüfen, inwieweit aus den Ideen von Jörg Wolfram (eigene Hardware mit BAS+PS/2 und SRAM-8bit) inzwischen etwas geworden ist..

Peter

**Autor: Dagobert (Gast)**

**Datum: 30.07.2010 11:44**

---

Das Problem bei fast allen Tastatur-Routinen ist es, dass meist nur Normal und Shift-Ebene unterstützt wird, Alt-Ebene oder gar Sondertasten kann man lange suchen. Von Caps-Lock-Steuerung rede ich gar nicht erst.  
;)

Eine vollständige Tastatursteuerung, wie in VGATerm enthalten, hat es in sich.

Auch die Ansteuerung des µVGA-Moduls ist sehr Tückenreich. Z.B. gibt es oft Timing-Probleme, wodurch man viele Warteschleifen einbauen muss. Das wurde mit VGATerm ebenso, dank eines 32 kB Empfangspuffers, eliminiert.

Aber wenn man die Unwegsamkeiten erst einmal überwunden hat, hat man eine wirklich nutzbare VGA-Ansteuerung. Bei den oben genannten Systemem fallen viele CP/M-Programme weg, weil diese einen 80x25-Zeichen Bildschirm erfordern.

**Autor: Peter Sieg (Gast)**

**Datum: 30.07.2010 12:53**

---

Dagobert schrieb:

> Bei den oben genannten Systemem  
> fallen viele CP/M-Programme weg, weil diese einen 80x25-Zeichen  
> Bildschirm erfordern.

Ja.. alerdings basiert die BAS Lösung auf [ATmega8](#) mit 16MHz.. Mit 88/168 und 25-30MHz sollte sich da ggf. noch was dran machen lassen..  
Propterm hat schon alles was es braucht..  
Allerdings zeigenja auch gerade VGATerm + uVGA das es grundsätzlich geht  
;-)  
Cool wäre aber aus meiner Sicht auch eine LCD Lösung..

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 30.07.2010 13:23**

---

Peter Sieg schrieb:

> Warum nicht.. ist doch jetzt auch ein festes Format.. man muß doch nicht  
> immer alles variabel halten..

Dann denk dir ein schönes Format aus (Größe u. Anzahl Dir-Einträge),  
schnapp Dir das Alteration Guide oder eine andere Beschreibung und  
erstelle die beiden Tabelleneinträge. Dann einfach in bios.asm  
eintragen. den dpb braucht man dann nur einmal. Ich hatte nur 4 Plätze  
für maximal 4 verschiedene Diskformate vorgesehen. Gestern Abend dachte  
ich noch, damit wäre schon alles fertig.

Leider hatte ich nicht dran gedacht, daß bei Diskgrößen über 256KB die  
Blockgröße mindestens 2048 sein muß. Und die ist z.Zt. noch als  
Konstante im avr-Teil.

```
also muß man noch in z80.asm unter
;*****
;*          CP/M to host disk constants          *
;*****
```

ändern auf

```
.equ blksize = 2048      ;CP/M allocation size
```

(oder größer)

Bei der Gelegenheit kann man auch die Anzahl Sektoren pro Track auf  
einen etwas größeren und "runderen" Wert setzen. Das muß aber nicht  
sein. Mit 26 SPT kommt man auch bis 218MB Diskgröße. zB:

```
.equ CPMSPT = 32 ; oder 64 oder ...
```

Fertig. Ergebnis ggf. hier reinstellen oder mir zuschicken.  
Natürlich darf sich auch jede(r) andere hier angesprochen fühlen.

Ach ja, ein passender Formateintrag für die cpmttools wäre dann auch nicht schlecht, damit man damit einfach vom alten Format aufs neue umkopieren kann.

```
> besser wäre es da wenn das ein Format wäre,  
> das man auch dem AltairZ80 Emulator beibringen könnte.. dann kann man  
> Images  
> im Emulator laufen lassen und per R/W auch Dateien zw. CP/M und  
> Hostsystem transferieren..
```

Falls Du den simh Altair 8800 Simulator meinst (den kenne ich, es gibt aber wahrscheinlich noch andere):

Der hat variable Formate! Das Disk Image ist sicher irgendwo dokumentiert (auf jedenfall im source, natürlich) aber ich bin bis jetzt noch nicht über die Doku gestolpert. Vor dem eigentlichen CP/M-Diskimage ist auf jeden Fall noch ein nicht gerade kleiner Header.

## Wir gehen auf 8-bit Dram

Autor: Leo C. ([rapid](#))

Datum: 30.07.2010 13:46

---

Joe G. schrieb:

```
> Es gab Anfragen ob noch Leiterplatten verfügbar wären. Leider nein.  
> Bei genug Interessenten könnte natürlich eine neue Serie mit den  
> entsprechenden Änderungen aufgelegt werden. Mir fällt da spontan ein:  
> - geänderte Pinbelegung am RAM (RAM Disk)  
> - Video-VGA siehe weiter oben.  
> - weitere Vorschläge ???
```

Wie wärs mit 2 RAM-Chips (8 Bit breiter Datenbus)? Die Chips sind ja massenweise kostenlos vorhanden und der Schaltungsmehraufwand ist imho minimal. Also immer noch ein Lowcost Projekt.

Damit sich das lohnt, braucht man am Prozessor einen 8 Bit breiten Port. Den bekommt man, wenn man die Serielle Schnittstelle verlegt. So sieht dann das Ergebnis aus:

```
-----  
          | Peters |           Leos           |  
          | Schleife |           Schleife       |  
-----+-----+-----+-----+-----+-----  
AVR  MHz | Zeit [s] | Zeit [s] | "Speed" | HW  | SW  
-----+-----+-----+-----+-----+-----  
  8   20 | 12,180   |           |           | A   | 0  
  8   20 | 12,453   | 3.488    | 827 KHz  | A   | 1  
  8   20 | 11.500   | 3.488    | 827 KHz  | A   | 3  
-----+-----+-----+-----+-----+-----  
  8   20 | 7.599    | 2.039    | 1380 KHz | C   | 2  
  8   20 | 6.680    | 2.039    | 1380 KHz | C   | 3  
-----+-----+-----+-----+-----+-----
```

8 20 6.418 1.958 1470 KHz C 4

"HW A" ist die jetzt aktuelle Verdrahtung und "HW C" die 8-Bit Variante.

Es ist also fast nochmal eine Geschwindigkeitsverdopplung drin.

Zu den Details hatte ich am Montag schon mal einen Artikel geschrieben, der aber hier nicht (mehr?) vorhanden ist. Vielleicht drücke ich ja öfters auf Vorschau statt Absenden, oder...

Die Software dazu gibts in meinem [SVN](#) im softuart Zweig.

**Autor: Peter Sieg (Gast)**

**Datum: 30.07.2010 13:58**

---

[Leo C. schrieb:](#)

> Wie wärs mit 2 RAM-Chips (8 Bit breiter Datenbus)? Die Chips sind ja  
> massenweise kostenlos vorhanden und der Schaltungsmehraufwand ist imho  
> minimal. Also immer noch ein Lowcost Projekt.

Die Idee finde ich super! Welche Baudrate ist dann mit Soft-Uart noch sinnvoll? Alternativ wäre natürlich auch ein [AVR](#) mit mehr Pin's :-)

Leider bin ich dieses WE in Familienfeier unterwegs.. ich werde mir aber nächste Woche mal das CP/M Alteration Guide besorgen und mal versuchen, das diskimage Format zu vergrößern.. soll aber niemanden davon abhalten das auch zu machen.. bei mir kann das schon etwas dauern.. ;-)

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 30.07.2010 14:13**

---

[Peter Sieg schrieb:](#)

> Die Idee finde ich super! Welche Baudrate ist dann mit Soft-Uart noch  
> sinnvoll? Alternativ wäre natürlich auch ein [AVR](#) mit mehr Pin's :-)

Senden geht mit 115k problemlos.

Empfangen habe ich bis 56700 getestet. Das ging, als die größten Fehler draußen waren. Es könnte auch noch mehr dring sein. Der Code ist aber noch etwas chaotisch und müßte dringend aufgeräumt werden. Dann könnte man auch noch eine etwas bessere Fehlererkennung/behandlung einbauen. "Leider" gibts bei der üblichen Verbindung mit PC ja so gut wie keine Übertragungsfehler.

Gilt alles für 20Mhz Prozessortakt.

> Leider bin ich dieses WE in Familienfeier unterwegs.. ich werde mir aber  
> nächste Woche mal das CP/M Alteration Guide besorgen und mal versuchen,  
> das diskimage Format zu vergrößern.. soll aber niemanden davon abhalten  
> das auch zu machen.. bei mir kann das schon etwas dauern.. ;-)

Ich werde mich bis auf Weiteres auch nicht dransetzen, da ich ja gar kein größeres Format brauche. ;-)

**Autor: Joe G. ([feinmechaniker](#))**

**Datum: 30.07.2010 14:38**

---

[Leo C. schrieb:](#)

> Wie wärs mit 2 RAM-Chips (8 Bit breiter Datenbus)? Die Chips sind ja  
> massenweise kostenlos vorhanden und der Schaltungsmehraufwand ist imho  
> minimal. Also immer noch ein Lowcost Projekt.

Gute Idee Port D doppelt zu belegen, auch PC4 und PC5 frei zu lassen.  
Was möchtest du dort angeschlossen haben, ein RTC, eine PIO, ...  
Ich erstelle mal einen neuen Schaltplan zu der Soft-Uart Variante.

**Autor:** Leo C. ([rapid](#))

**Datum:** 30.07.2010 15:38

---

[Joe G. schrieb:](#)

> Was möchtest du dort angeschlossen haben, ein RTC, eine PIO, ...

Wg. I2C drängt sich eine RTC ja förmlich auf. Aber irgend was sträubt  
sich in mir. Gibt es I2C-(Flash-)ROMs im ca. einstelligen Megabit  
Bereich? So als ROM-Disk oder SSD?

Ursprünglich dachte ich an Inputs, z.B. Taster für Z80 Reset, bzw.  
Interrupt.  
Damit könnte man z.B. ein hängendes Programm in den Debugger  
zurückholen.

Frontpanel?

CS-Leitungen für SPI-Erweiterungen wäre auch denkbar.

Am Besten ein großes Lochrasterfeld vorsehen. Und/Oder Pfoften für  
Huckepackererweiterungen.

**Autor:** Joe G. ([feinmechaniker](#))

**Datum:** 30.07.2010 16:56

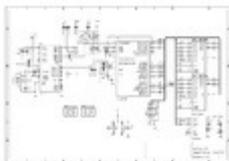
---

Als DRAM habe ich 1Mx4 im SOJ26-20 Package vorgesehen. Das sollte noch  
gut zu löten sein und außerdem gut verfügbar. Bei I2C bin ich mir noch  
unsicher, Portererweiterung ist auch denkbar.

**Autor:** Joe G. ([feinmechaniker](#))

**Datum:** 30.07.2010 17:53

**Angehängte Dateien:**



• [avrcp\\_m.pdf](#) | [anzeigen](#)  
27,4 KB, 157 Downloads

---

@Leo C. alle anderen natürlich auch.

Bitte mal durchsehen ob der RAM / MMC mit der Soft-Uart Variante  
übereinstimmt.

Autor: Leo C. ([rapid](#))

Datum: 30.07.2010 18:20

---

Das ist ja mal wieder ein Mist. (sorry, mußte raus)

Ich habe gerade meine Bestände durchgesehen, und festgestellt, daß ich fast nur 4Mx4 habe. Ich dachte mir, macht ja nix, man kann ja einfach die 2 Adressleitungen zusätzlich verbinden. In dem einen Datenblatt, daß ich von 1Mx4 hier habe, habe ich aber gerade gesehen, daß die anscheinend eine völlig andere Pinbelegung haben.

Joe G. schrieb:

> @Leo C. alle anderen natürlich auch.  
>  
> Bitte mal durchsehen ob der RAM / MMC mit der Soft-Uart Variante  
> übereinstimmt.

Ich habe bis jetzt nur mal kurz drauf geschaut. Und es stimmt leider. Deine Verdrahtung paßt zu dem 1Mx4 Datenblatt, und die 4Mx4 Datenblätter sind alle anders. Beides vorsehen, für wahlweise Bestückung, wird wohl viel zu aufwendig sein. Mir ist es egal, weil ich keine Platine brauche.

In meiner PC-Restekiste habe ich 7 verschiedene 72-polige RAM-Modul Sorten.  
Davon nur eines mit 1Mx4 bestückt.

Autor: Leo C. ([rapid](#))

Datum: 30.07.2010 19:04

---

Joe G. schrieb:

> @Leo C. alle anderen natürlich auch.  
>  
> Bitte mal durchsehen ob der RAM / MMC mit der Soft-Uart Variante  
> übereinstimmt.

An die Datenleitungen von IC5 (oder IC4) sollten D4\_A4 bis D7\_A7 angeschlossen werden.

1Mx4 Chips haben auch eine A9. Nächster Pin wäre PB4 (MISO).  
Und falls es 4Mx4 Chips werden sollten, käme noch eine A10 dazu (--> PB5).

Der Rest scheint ok zu sein. RAS bis WE, A8 bis A9/10 und D0\_A0 bis D7\_A7 kann man sowieso frei untereinander tauschen, wenn dadurch z.B. das Layout einfacher wird.

Pullups am SD-Slot?

Chan empfiehlt sie sehr (aber einen Pulldown an CLK), hat sie aber bei seinem neuesten GPS-Logger auch weggelassen.

Ich mache immer eine [LED](#) an CS mit R nach VCC. Macht sich hier besonders gut als Drive-Select-Lampe.

Autor: Peter Sieg (Gast)

Datum: 31.07.2010 11:33

---

@all: An Rams kann man aber auch die 256x4 2-mal nehmen.. 2ter Chip in jetziger Platine huckepack, nur D0-D3 an D4-D7.. damit sollte die

vorhandene

Platine auch auf 8-bit Dram erweiterbar sein (4 Leitungen zum Huckepackram legen + Uart umverdrahten..)

(PS: Ich finde die SOJ SCH\*\*\* zum löten ;-)- aber habe ich auch schon hinbekommen..)

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 31.07.2010 11:47

---

[Leo C. schrieb:](#)

> An die Datenleitungen von IC5 (oder IC4) sollten D4\_A4 bis D7\_A7  
> angeschlossen werden.

Hab ich beim abspeichern gemerkt, ist schon geändert.

[Leo C. schrieb:](#)

> Pullups am SD-Slot?  
> Chan empfiehlt sie sehr (aber einen Pulldown an CLK), hat sie aber bei  
> seinem neuesten GPS-Logger auch weggelassen.

Pullups sind ja normalerweise im [AVR](#) schon vorhanden, Pulldown an CLK?  
steilere Flanken? Mein Oszi zeigt kein Unterschied.

[Leo C. schrieb:](#)

> Ich mache immer eine [LED](#) an CS mit R nach VCC. Macht sich hier besonders  
> gut als Drive-Select-Lampe.

Nette Idee.

RAM Frage:

1M x4 oder 4M x4, mir ist es egal. Peter möchte nicht löten ;-)- Ich  
nehme meine ursprüngliche Idee mal wieder auf. Für den RAM gibt es eine  
28 polige IC Fassung und da steckt jeder drauf was er hat. Als  
"Default-RAM-Stecker-Layout" gib es drei kleine RAM-Adapterplatinen, 1M  
x4, 4M x4, 256 x 4.

Autor: Leo C. ([rapid](#))

Datum: 31.07.2010 13:21

---

[Joe G. schrieb:](#)

> Pullups sind ja normalerweise im [AVR](#) schon vorhanden, Pulldown an CLK?  
> steilere Flanken? Mein Oszi zeigt kein Unterschied.

Am CLK ist Low-Pegel der Normal/Ruhezustand. Bei den anderen Pins ist es  
High.

Zur Zeit werden die internen Pullups komplett abgeschaltet, damit man  
ganze Bytes auf die Ports schreiben kann, ohne daß Pullups in Portbits,  
die als Input programmiert sind, ständig ein- und ausgeschaltet werden.  
Für die jetzige Schaltung kann man das aber ändern, wenn man PC4 und PC5  
nicht als Inputs benutzt, oder das Pullupflattern egal ist.

Bleibt noch der Resetzustand...

> RAM Frage:

> 1M x4 oder 4M x4, mir ist es egal. Peter möchte nicht löten ;-)- Ich  
> nehme meine ursprüngliche Idee mal wieder auf. Für den RAM gibt es eine  
> 28 polige IC Fassung und da steckt jeder drauf was er hat. Als

> "Default-RAM-Stecker-Layout" gib es drei kleine RAM-Adapterplatinen, 1M  
> x4, 4M x4, 256 x 4.

Mir ists auch egal. Und Peter hat ja nur geschrieben, daß er nicht **gerne** lötet.

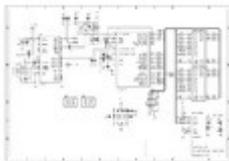
Ich habe bei mir das 2. SOJ-RAM übrigens auch Huckepack auf das erste gelötet. Wobei es jetzt eigentlich kein SO **J** mehr ist. ;-)

Schönes Wochenende  
Leo  
(Nach Diktat verreist)

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 31.07.2010 18:53

Angehängte Dateien:



• [avrcp\\_m.pdf](#) | [anzeigen](#)  
28,5 KB, 156 Downloads

---

Hier nun die SOJ26-24 Variante, also CP/M mit 4 MB. D0-D7 nun korrekt, Pullup und Pulldown an MMC, [LED](#) an CS noch nicht (kommt noch) und I2C wartet noch auf Ideen.

Autor: Peter Sieg (Gast)

Datum: 01.08.2010 11:29

---

@Joe G. Können gerne SOJ nehmen.. kriege ich hin.. ist aber für sonst nur DIL Verbauer eine kleine Herausforderung ;-). Deshalb: Einfach auf eine Version einigen..und fertig. Keep it simple.. (Ich habe eine ganze Schachtel voll alter Ramriegel.. muß selbst erstmal schauen, was ich da noch so finde..)

Eine Idee hätte ich allerdings noch:  
Quarz+2xC ersetzen durch Quarzoszillator.. es ist so gut wie unmöglich ab >25MHz noch Grundtonquarze zu bekommen (für 30MHz ist mir das allerdings noch gelungen..).. Aber Quarzoszi's für 32/36/40MHz ist kein Problem..

--- zum softuart und 8-bit dram access ---

```
; - Soft UART TX (OC1A/OCR1A).
```

```
; - Soft UART RX (ICP1/ICR1).
```

Das sin die beiden Uart Pins?

Ansonsten sehe ich nur D0-D3 definiert..? Ich hatte D4-D7 erwarten..? Normalerweise dachte ich 2tes Dram Huckepack aufzulöten - alle Pins - bis auf D0-D3 und diese 4 Pins ergeben dann das 2te Nibble..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 01.08.2010 11:44

---

Peter Sieg schrieb:

> Quarz+2xC ersetzen durch Quarzoszillator..

Ja, ist sicher besser. Schon in Hinblick auf mögliche Erweiterungen in Richtung VGA (25.179 MHz) ;-)

Peter Sieg schrieb:

> Ansonsten sehe ich nur D0-D3 definiert..?

Schau mal im letzten Schaltplan, den Tag davor hatte ich geträumt. Die Soft UART Pins sind so wie in Leo C. Quelle.

C4 oder C5 könnte man auch als /CS für ein weiteres SPI-Gerät nutzen. Ich habe hier noch ein Vinculum Vdrive rumliegen. Damit hat Leo C. sein SSD im mehrstelligen GB-Bereich oder was ein USB Stick so hergibt. FAT ist auch schon dabei, außerdem könnte man CP/M Programme über USB-Stick austauschen.

Joe

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 02.08.2010 08:24

---

@ Peter Sieg

ich habe damit angefangen und es funktioniert auch recht gut, allerdings mehr als die Testprogramme hier im Thread habe ich nicht laufen lassen. Aus Mangel an Interesse sowohl bei mir als auch im ChipBasic Thread habe ich das Ganze aber erstmal wieder auf Eis gelegt. Beim nächsten Release wird der 8080-Emulator mit dabei sein, mehr aber wahrscheinlich nicht.

Jörg

## BDS-C Compiler

Autor: Peter Sieg ([petersieg](#))

Datum: 05.08.2010 08:24

Angehängte Dateien:

- [bdscimage.zip](#) (74 KB, 46 Downloads)

---

Hallo. Hier noch ein weiteres diskimage (um die Partitionen zu füllen ;-).

Es enthält den BDS C-Compiler und ein paar C-Sourcen von Spielen.

Übersetzung mit cc <datei.c> und dann clink <datei>.

Belegt nur ca. 100kb auf dem 243kb Image.

ED als Zeileneditor ist ebenfalls enthalten. Leider läuft der Fullscreen-Editor VDE nicht..? Falls jemand noch einen kleinen, einfachen Fullschreeneditor für CP/M-80 hat..

---

MicroVGA läuft auch am avr cp/m. 115200 baud schafft es aber nicht ohne HW Handschake.. läuft jetzt mit 38400 baud.. stelle noch zwei Bilder ein..

Der 3,3V Ausgang des MicroVGA ist hier wirklich sehr praktisch!

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 05.08.2010 09:08

---

Peter Sieg schrieb:

> MicroVGA läuft auch am avr cp/m.

Hat deine einstabiles Bild? Bei mir sehen die Buchstaben arg zerfranst aus. Hast du mal die Pegel der seriellen Schnittstelle gemessen, 5V oder 3.3V ? Auf der Leiterplatte ist ja nicht so richtig zu erkennen wo die 3.3V Leitung bleibt und die 5V Leitung ist äußerst dünn ausgeführt.

Ich würde das neue Layout so ausführen, dass die MicroVGA Platine direkt aufgeschraubt und verbunnen werden kann. Für die 5V gibt es dann eine übliche Klinkenbuchse.

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 05.08.2010 10:11

---

Joe G. schrieb:

> Hat deine einstabiles Bild? Bei mir sehen die Buchstaben arg zerfranst  
> aus.

Das Bild ist stabil. Aber die Buchstaben sehen hier auf dem 22" TFT auch nicht schön aus :- ( Habe sowas aber schon bei einigen Sachen gesehen, wo 800x600/640x480/80x25 Text (also geringe Auflösungen) versucht wird auf solche großen Bildschirme 'hochzurechnen'..

Die Anschlüsse so zu legen, das MicroVGA direkt passt ist eine gute Idee!

Ist wie gesagt eine doch noch recht preiswerte Möglichkeit einen Stand-Alone CP/M Rechner zu basteln.. Ca. 30€ sind aus meiner Sicht ok. Muß noch mal sehen, das auch Programme mit VT100/Ansi Sequenzen auch gut laufen..

Eine Alternative könnte PropTerm sein.. Habe dafür aber noch kein Fertigmodul/Bausatz/Platinenlayout/etc. gefunden..

Was ich z.Z suche ist ein einfacher Fullscreen Ascii/Text-Editor..

Wäre auch schön, wenn der einer oder andere auch mal ein diskimage zusammen stellen würde mit z.B anderen Programmiersprachen oder Anwendungen etc.

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 05.08.2010 12:20

---

Schaltug / Layout

Da der Wunsch nach Taktfrequenzen über 20 Mhz besteht und Quarze in diesem Bereich dünn werden, ist ja der Einsatz eines Quarzoszillators geplant. Nun laufen dieses (DIL Variante) wiederum nur mit 5V und nicht

mit 3.3V. Was haltet ihr von dieser Variante:

- [AVR](#) und RAM laufen auf 5V (kann damit höher übertaktet werden)
- MMC wird über 74VHC08 auf 3.3V betrieben.
- als Spannungsregler kommt wieder der interne Regler des FTDI zum Einsatz
- alle weiteren Busgeräte I2C, SPI laufen auf 5V oder SPI auf 3.3V

Das Konzept erfordert zusätzlich genau einen 74VHC08.

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 05.08.2010 13:18

---

Hmm.. SMD Oszi gibt z.B bei [reichelt](#) zw. 25-50MHz (XO32 40,00000). Die laufen mit 3,3V...

Ich würde ungern das Layout/die nötigen Bauteile weiter 'aufblähen' ;-)

BTW: Warum Man(n) auch unbedingt 2 x Dram's a 4-bit nehmen muß und nicht 1 x Sram (628128-70) muß Man(n) auch nicht wirklich verstehen.. ;-)  
(Nein, kein ez80 etc. ;-)

BTW: Hat jemand eine Lösung, wie ich cmd.exe unter XP so laufen lassen kann, das ich mit dem AltairSimh Emulator Programme testen kann, die VT52/VT100/Ansi Escape-Sequenzen nutzen..? Kann gerne auch ein anderer Shell-Ersatz etc. sein.. evtl. könnte ja telnetd + telnet hier gehen.. ist aber ein wenig oversized an Aufwand - oder?

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 05.08.2010 13:28

---

[Peter Sieg schrieb:](#)

> Warum Man(n) auch unbedingt 2 x Dram's a 4-bit nehmen muß und nicht  
> 1 x Sram (628128-70) muß Man(n) auch nicht wirklich verstehen.. ;-)

Wegen der fehlenden Pins - A0 bis A16.

Alles mit 5V laufen zu lassen hatte noch einen Grund. Meine VGA-Testschaltung läuft auch mit 25 MHz, das Schieberegister und die Leitungstreiber sowie die Tastaturschnittstelle mit 5V. Diese IC's habe ich nicht als VHC-Type gefunden. Diese Schaltung würde ich gleich mit anstückeln. So kann sie für weitere Experimente benutzt werden oder bei Bedarf durch die MicroVGA überbaut werden. Der beanspruchte Platz und die Pinbelegung ist gleich.

Joe

## Bilder von VGA-Term/uVGA

Autor: Peter Sieg ([petersieg](#))

Datum: 05.08.2010 15:09

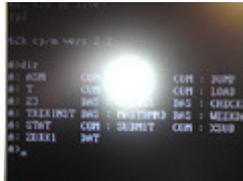
Angehängte Dateien:



- [DSC00489.JPG](#)  
96,7 KB, 444 Downloads



- [DSC00490.JPG](#)  
62,7 KB, 424 Downloads



- [DSC00491.JPG](#)  
51,8 KB, 370 Downloads

---

Hier die Bilder von [AVR](#) CP/M + MicroVGA.

---

@Joe: Was die HW angeht, bin ich fast für alles offen, wenn es dem CP/M Gedanken folgt, so einfach wie möglich - nur nicht einfacher!  
Mit dem bisher erreichten kann man schon sehr zufrieden sein!

2 x Drums mit mehr Bytes bringt:

a. Faktor 2 in der Geschwindigkeit

b. Mögliche Erweiterungen in Richtung Ramdisk etc.

=> Ergo. Sollte in einer nächste Version mit aufgenommen werden.

Quarzoszillator ermöglicht übertakten bis 36/40/?? MHz, was wiederum mehr Geschwindigkeit bringt und nicht viel kostet und die HW auch nicht komplizierter macht.

=> Ergo. Reinnehmen.

I2C eröffnet ebenfalls noch ungeahnte Möglichkeiten und kostet=nichts.

=> Ergo. Reinnehmen.

Das mit dem Vinculum USB Drive dürfte auch sehr interessant sein..

Eine einfache Möglichkeit Dateien zw. CP/M und Host-Welt auszutauschen wäre schon schön..

Alles andere, wenn es denn das System sinnvoll erweitert bzw. ergänzt.. warum nicht.. ein Selbstbauterminal (VGA-Out+PS/2 Tast.-In) sollte allerdings ersteinmal als LR/Steckbrettaufbau funktionieren..

Just my 2 cents.

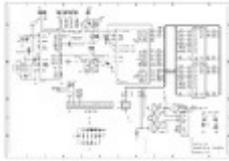
Peter

## Layout 2

Autor: Joe G. ([feinmechaniker](#))

Datum: 05.08.2010 15:56

Angehängte Dateien:



• [avr\\_cp\\_m.pdf](#) | [anzeigen](#)  
31,7 KB, 177 Downloads

---

Hier der Vorschlag mit folgenden Änderungen:

- 4MB RAM
- [LED](#) an CS MMC
- 3.3V an MMC, nicht mehr versehentlich auf 5V steckbar
- 3.3V wahlweise über FTDI oder MicroVGA
- [AVR](#) und RAM kann wahlweise mit 3.3V oder 5V betrieben werden (hat keinen Einfluß auf Spannung der MMC)
- Takt über FTDI oder Quarzoszillator
- mögliche Steckvarianten der V24 sind:
  - USB - CP/M
  - CP/M - VGA
  - USB - VGA

noch offen:

- Vinculum USB Drive
- I2C

Es geht nur eines von beiden, da bei einem weiteren SPI Device ein zusätzliches CS benötigt wird. Wir sollten uns auf eine Variante festlegen oder es als Lochraster auslegen.

[Peter Sieg schrieb:](#)

> sollte  
> allerdings ersteinmal als LR/Steckbrettaufbau funktionieren..  
Da funktioniert es Lochrastermäßig ;-)- H-Sync und V-Sync kommen exakt, Pixeltakt und Zeichen auch. Aber durch Steckbrettaufbau mit viiiiel Jitter. Schaltugsvorschlag dazu folgt noch.  
Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 05.08.2010 17:50

---

[Joe G. schrieb:](#)

> noch offen:  
> - Vinculum USB Drive  
> - I2C  
> Es geht nur eines von beiden, da bei einem weiteren SPI Device ein  
> zusätzliches CS benötigt wird.

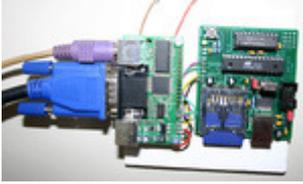
Hmm. VDRIVE2/VDIP1+2 können auch per UART gesteuert werden..  
Bräuchte eine 2te serielle Schnittstelle (9600 8N1) und ein entsprechendes Terminalprogramm unter CP/M, was über diese Schnittstelle mit dem V\* Device spricht und mit den vorhandenen Kommandos Dateien zw. CP/M und V\* Device liest und schreibt..

Peter

Autor: Joe G. ([feinmechaniker](#))

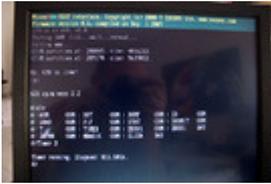
Datum: 05.08.2010 17:55

Angehängte Dateien:



[AVR\\_CPM\\_VGA\\_1.jpg](#)

102,8 KB, 559 Downloads



[AVR\\_CPM\\_VGA\\_2.jpg](#)

90,2 KB, 568 Downloads

---

na dann will ich auch mal...

hier mein Aufbau...

Joe

Autor: Joe G. ([feinmechaniker](#))

Datum: 05.08.2010 18:13

---

[Peter Sieg schrieb:](#)

> VDRIVE2/VDIP1+2 können auch per UART gesteuert werden..

> Bräuchte eine 2te serielle Schnittstelle (9600 8N1) und ein

> entsprechendes Terminalprogramm unter CP/M

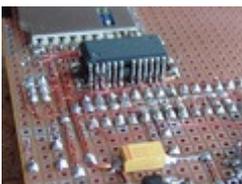
Das wird aber arg langsam...

Mir scheint SPI die bessere Lösung zumal man ja dem BIOS beibringen kann, dass genau dieses SPI Gerät die COM1 oder COM2, ... ist. Dann sollte auch ein Terminal gehen.

Autor: Leo C. ([rapid](#))

Datum: 06.08.2010 00:43

Angehängte Dateien:



[avrcpm-v2-r-a.jpeg](#)

265,6 KB, 564 Downloads

---

[Peter Sieg schrieb:](#)

> BTW: Warum Man(n) auch unbedingt 2 x Dram's a 4-bit nehmen muß und nicht  
> 1 x Sram (628128-70) muß Man(n) auch nicht wirklich verstehen.. ;-)

Für mich ist das Ganze ein Bastelkistenprojekt. D.h. es wird nur aus  
Teilen zusammengebaut, die in der Bastel-/Reste-/Ramschkiste vorhanden  
sind. D-Rams sind da reichlich drin. Große statische Rams eher nicht.  
(ATmegas habe ich allerdings auch nicht im Überfluß, und meinem  
GPS-Logger habe ich schon den 328P geklaut. ;)

Peter Sieg schrieb:

> Hier die Bilder von [AVR](#) CP/M + MicroVGA.

Am Besten gefallen mir die geschraubten Nagelschellen. ;-)

Joe G. schrieb:

> noch offen:

> - Vinculum USB Drive

Kommt für mich nicht in Frage. (siehe oben)

> - I2C

> Es geht nur eines von beiden, da bei einem weiteren SPI Device ein  
> zusätzliches CS benötigt wird. Wir sollten uns auf eine Variante  
> festlegen oder es als Lochraster auslegen.

Ich fände ein paar Inputs ganz brauchbar. ZB. vom SD-Kartensockel  
Card-Detect und evtl Write-Protect. Ferner Z80 Interupt/Reset. Wenn die  
2 Pins nicht reichen: Schieberegister.

Im Anhang mal ein Bild vom Huckepack-Ram. (ex SOJ:)

Autor: Leo C. ([rapid](#))

Datum: 06.08.2010 01:50

Angehängte Dateien:

- [cpm.bin](#) (6,4 KB, 64 Downloads)

---

In meinem [SVN](#) Repo gibts ein paar Updates. Im trunk ("alte" Hardware)  
eigentlich nur ein paar Bugfixes. Bei einer partitionierten SD-Karte  
wird jetzt sichergestellt, daß nicht außerhalb der CP/M-Partitionen  
gelesen, und vor allem geschrieben wird. Wer also seine  
CP/M-Partition(en) kleiner als das darin befindliche CP/M-Dateisystem  
definiert hat, und noch andere, wichtige Daten auf der Karte hat, sollte  
unbedingt updaten.

Im softuart-Zweig (8-Bit DRAM) das Gleiche. Plus RAM-Disk (Laufwerk I:).  
Die RAM-Disk ist im Moment nur 64KB groß, damit sie auch bei den  
kleinsten RAM-Chips noch funktioniert. Davon gehen dann noch 8KB für 2  
Systemspuren ab.

Beim Kaltstart wird das CP/M von A: auf die RAM-Disk kopiert und beim  
Warmstart von dort neu geladen. Sinn des ganzen sollte sein, daß man  
nach dem booten die SD-Karte ziehen kann, und ohne sie weiterarbeiten  
kann.

Leider macht uns CP/M hier einen Strich durch die Rechnung: Beim  
Warmstart wird grundsätzlich auf A: zugegriffen, auch wenn der Benutzer  
auf ein anderes Laufwerk umgeschaltet hat.

Abhilfe:

1. BDOS patchen. Andere (z.B. simh) machen das auch so.
2. Die RAM-Disk als Laufwerk A: verdrahten.
3. Alternatives CP/M (BDOS) verwenden? (Bekannte, gute Alternativen setzen Z80 voraus)

Anderes, damit im Zusammenhang stehendes Problem:

Wenn man die SD-Karte zieht und wieder einsteckt, wird beim nächsten Zugriff das System komplett neu gestartet. Das ist erst mal leicht zu korrigieren, in dem z.B. bei einem Warmstart (^C) die Karte neu initialisiert wird. Allerdings muß man dann auch die Partitionierung neu einlesen. Wenn bei jedem Warmstart die Liste der gefundenen Partitionen erscheint, nervt das aber...

Für Vorschläge, wie man das System an dieser Ecke rund macht, bin ich dankbar. Hat überhaupt schon jemand die 8-Bit-Ram Variante nachgebaut, und kann damit die RAM-Disk mal testen? Ich weiß noch nicht, ob ich die in die 4-Bit-Ram Variante überhaupt noch einbauen soll.

Beide neuen Versionen laufen nur mit neuem BIOS, und das neue BIOS nur mit neuem IPL. Beides ist auch im [SVN](#). Ich habe aber auch mal ein fertigs System hier angehängt.

Autor: Peter Sieg ([petersieg](#))

Datum: 06.08.2010 08:46

---

[Leo C. schrieb:](#)

> Hat überhaupt schon jemand die 8-Bit-Ram Variante nachgebaut,  
> und kann damit die RAM-Disk mal testen? Ich weiß noch nicht, ob ich die  
> in die 4-Bit-Ram Variante überhaupt noch einbauen soll.

Hi Leo C.

Ich bin noch nicht dazu gekommen, die 8-bit Variante nachzubauen. Ich plane dazu meine LR Version umzubauen. Könntest du bitte nochmal aufzeigen, wie genau ich was verdrahten muß? Ich gehe im Moment davon aus:

1. 2ten Dram Chip 1:1 mit dem ersten verbinden (Huckepack) - bis auf D0-D3!
2. Dram 1ter Chip verbinden:  
D0 - 2 - PD0  
D1 - 3 - PD1  
D2 - 4 - PD2  
D3 - 5 - PD3
3. Dram 2ter Chip verbinden:  
D0 - 6 - PD4  
D1 - 11 - PD5  
D2 - 12 - PD6  
D3 - 13 - PD7
4. Uart RX liegt dann an [AVR](#) Pin 14 = ICP1/ICR1
5. Uart Tx liegt dann an [AVR](#) Pin 15 = OC1A/OCR1A

Ups.. ich sehe gerade, das ist eine komplette Umverdrahtung!! Da kann ich wahrscheinlich schneller einen kompletten Neuaufbau auf LR machen.. ;-)  
Meine Platinen werde ich jedenfalls nicht so vergewaltigen.. Ich glaube da brauchen wir die nächste Platinenversion doch eher als ich gedacht hatte.. ;-)

> LW A: in Ramdisk umkopieren..

Das würde ich so nicht machen.. Evtl. hat man z.B seinen C Compiler auf C: und möchte eher diesen in der Ramdisk haben (wozu allerdings 64kb zu wenig wäre..) .. eher sowas wie eine autoexec.bat ausführen, wo man dann selbst definieren kann, was evtl. wohin kopiert werden soll..

>Wenn man die SD-Karte zieht und wieder einsteckt, wird beim nächsten  
>Zugriff das System komplett neu gestartet. Das ist erst mal leicht zu  
>korrigieren, in dem z.B. bei einem Warmstart (^C) die Karte neu  
>initialisiert wird. Allerdings muß man dann auch die Partitionierung neu  
>einlesen. Wenn bei jedem Warmstart die Liste der gefundenen Partitionen  
>erscheint, nervt das aber...

Ach.. macht doch nichts.. oder wenn gefundene Part. den vorher  
gefundenen entsprechen.. Meldungen unterdrücken.. ;-)

>Ich weiß noch nicht, ob ich die  
>in die 4-Bit-Ram Variante überhaupt noch einbauen soll.  
Da dann nur 64k zur Verfügung wären.. eher nein.. oder kann man das für  
temp. Dateien eines C Compilers nutzen..?

Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 06.08.2010 09:33

---

[Leo C. schrieb:](#)

> Ich fände ein paar Inputs ganz brauchbar. ZB. vom SD-Kartensockel  
> Card-Detect und evtl Write-Protect. Ferner Z80 Interupt/Reset. Wenn die  
> 2 Pins nicht reichen: Schieberegister.

- [PCF8574](#) an I2C? Damit gibt es nochmal 8 Bit.

> Im Anhang mal ein Bild vom Huckepack-Ram. (ex SOJ:)

Deshalb brauche ich eine Platine, solche Lötarbeiten werden mit  
zunehmenden Alter unmöglich.

> Abhilfe:

> 3. Alternatives CP/M (BDOS) verwenden? (Bekannte, gute Alternativen  
> setzen Z80 voraus)

Ich denke diese Variante ist die flexibelste Lösung.

> Hat überhaupt schon jemand die 8-Bit-Ram Variante nachgebaut,  
> und kann damit die RAM-Disk mal testen? Ich weiß noch nicht, ob ich die  
> in die 4-Bit-Ram Variante überhaupt noch einbauen soll.

Ich noch nicht, werde ich auch ohne neue Platine nicht  
(Altersaugenpulver ;-)). Von mir aus reicht die RAM-Disk in der 8-Bit  
Variante.

@alle

Gibt es denn außer beim Leo C. Peter und mir noch weitere lauffähige  
Systeme? Wenn nicht, können wir irgendwie helfen?

Autor: Leo C. ([rapid](#))

Datum: 06.08.2010 12:57

---

[Peter Sieg schrieb:](#)

> Könntest du bitte nochmal  
> aufzeigen, wie genau ich was verdrahten muß? Ich gehe im Moment davon  
> aus:

Du kannst Den neuen Schaltplan von Joe nehmen.

> Ups.. ich sehe gerade, das ist eine komplette Umverdrahtung!!

Überraschung ;-)

> Da kann  
> ich wahrscheinlich schneller einen kompletten Neuaufbau auf LR machen..  
> ;-) Meine Platinen werde ich jedenfalls nicht so vergewaltigen.. Ich  
> glaube da brauchen wir die nächste Platinenversion doch eher als ich  
> gedacht hatte.. ;-)

Ich hatte mich eh schon gewundert, als Du mal vom Platinenumverdrahten geschrieben hattest.

>> Wenn bei jedem Warmstart die Liste der gefundenen Partitionen  
>> erscheint, nervt das aber...  
> Ach.. macht doch nichts..

Doch, die Warmstarts werden ja auch bei (fast) jeder Beendigung eines Programms ausgeführt. Bei Batchdateien nach jeder Zeile.

> oder wenn gefundene Part. den vorher  
> gefundenen entsprechen.. Meldungen unterdrücken.. ;-)

Oder umgekehrt: Nur ausgeben, wenn Partitionen geändert. ;)  
So werde ichs machen. Kartenwechsel ohne Kaltstart muß einfach möglich sein, da die "usability" sonst in den Keller sinkt.

>>Ich weiß noch nicht, ob ich die  
>>in die 4-Bit-Ram Variante überhaupt noch einbauen soll.  
> Da dann nur 64k zur Verfügung wären.. eher nein.. oder kann man das für  
> temp. Dateien eines C Compilers nutzen..?

Kommt wahrscheinlich auf die Größe Deiner C-Programme an. :)  
Ansonsten ist das natürlich der Hauptzweck einer RAM-Disk.

Autor: Leo C. ([rapid](#))

Datum: 06.08.2010 13:18

---

[Joe G. schrieb:](#)

> Deshalb brauche ich eine Platine, solche Lötarbeiten werden mit  
> zunehmenden Alter unmöglich.

Da sind wir Kurzsichtigen wohl im Vorteil. Im Nahbereich gehts bei mir immer noch ganz gut.

> Von mir aus reicht die RAM-Disk in der 8-Bit  
> Variante.

Ich habe sie jetzt trotzdem mal in die alte 4-Bit Variante eingebaut.  
Ist im [SVN](#) zu finden, und braucht das neue BIOS, wie oben geschrieben.

Ich hoffe, das damit ein paar mehr Leute testen können. Ich brauche etwas Feedback in Sachen Benutzbarkeit. Außerdem sind wahrscheinlich noch Fehler drin, die ich alleine nicht finde.

> @alle  
> Gibt es denn außer bein Leo C. Peter und mir noch weitere lauffähige  
> Systeme? Wenn nicht, können wir irgendwie helfen?

Das würde mich auch mal interessieren. Insbesondere interessiert mich, **warum** jemand das Ding hier aufbaut, und was er/sie damit tun möchte. Grundsätzlich wurde das zwar am Anfang des Threads ausgiebig diskutiert, aber was die Leute antreibt, die jetzt tatsächlich bauen, kann ich mir gar nicht so richtig vorstellen. ;)

Autor: Peter Sieg ([petersieg](#))

Datum: 06.08.2010 14:47

---

[Leo C. schrieb:](#)

> Ich habe sie jetzt trotzdem mal in die alte 4-Bit Variante eingebaut.  
> Ist im [SVN](#) zu finden, und braucht das neue BIOS, wie oben geschrieben.

Habe jetzt mal eine Platine damit ausgestattet..

I>a:stat

A: R/W, Space: 42k  
B: R/W, Space: 163k  
C: R/W, Space: 129k  
I: R/W, Space: 55k

läuft ersteinmal soweit gut..  
aber das sieht komisch aus:

I>a:pip ed.com=c:ed.com

I>dir

I: ED           COM  
I>ed

Bdos Err On M: Select

I>

Nur mal zu den Gründen, die mich antreiben:

- \* CP/M ist DAS Ur-Betriebssystem.
- \* Das mit den heutigen Mikrocontrollern verbandelt ist ansich schon interessant
- \* Man ist automatisch gleichzeitig 'Retro' und modern ;-)
- \* Man hat eine 'arbeitsfähige' retro-Umgebung mit 2 Chips+SD Karte
- \* MBasic, C Compiler, Wordstar, dbase-II, etc. etc.
- \* Aber evtl. kann man Retro+Modern noch weiter verheiraten:  
->z.B ADC über [AVR](#) nach CP/M verfügbar machen, Frequenzzähler, I2C Bausteine

Peter

Autor: Leo C. ([rapid](#))

Datum: 06.08.2010 15:20

---

[Peter Sieg schrieb:](#)

> I>a:pip ed.com=c:ed.com  
> Bdos Err On M: Select  
> I>

Danke dafür.

Ich hatte heute schon mal was ähnliches, konnte es aber nicht reproduzieren. Bei mir war nur der Select auf J, also (genau) eins größer als ein existierendes Laufwerk. Jetzt kann ich davon ausgehen, daß der Buchstabe Zufall war, der eigentliche Fehler aber nicht.

Gerade nochmal propiert: Fehler versteckt sich. :-(

> Nur mal zu den Gründen, die mich antreiben:  
> \* CP/M ist DAS Ur-Betriebssystem.

Ich bin sozusagen damit groß geworden. ;-)  
Vor allem aber ist mein erster Selbstbaucomputer damit groß geworden.

> \* Das mit den heutigen Mikrocontrollern verbandelt ist ansich schon  
> interessant  
> \* Man ist automatisch gleichzeitig 'Retro' und modern ;-)

Genau. Besonders beim Programmieren empfinde ich das so.

> \* Man hat eine 'arbeitsfähige' retro-Umgebung mit 2 Chips+SD Karte  
> \* MBasic, C Compiler, Wordstar, dbase-II, etc. etc.

Es sind jetzt schon 3 Chips. ;) Und beim "arbeitsfähig" bin ich immer noch ein wenig skeptisch, wg. (fehlender) Geschwindigkeit. Aber prinzipiell ja.

> \* Aber evtl. kann man Retro+Modern noch weiter verheiraten:  
> ->z.B ADC über [AVR](#) nach CP/M verfügbar machen, Frequenzzähler, I2C  
> Bausteine

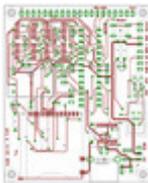
Dafür fehlt mir jetzt der Sinn. Geräte mit ADC, Frequenzzähler, etc, programmiert man besser mit was anderem, als ausgerechnet CP/M. Außerdem sollte man dann endgültig einen größeren [AVR](#) nehmen...

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 06.08.2010 18:03

Angehängte Dateien:

- [avrcp\\_m.zip](#) (91,8 KB, 47 Downloads)



- [AVR\\_CPM.png](#)  
13,7 KB, 454 Downloads

---

So, es ist soweit...

Die Schaltung und das Layout der neuen Platine - sagen wir Version 2.0 ;-)

- das Board ist etwas breiter, so das die VGA direkt angeschlossen werden kann.  
- auf den freien Pins der VGA liegen zusätzlich die 25 MHz, und SDA/SCL zur Erweiterung für I2C

Bitte Änderungen, Vorschläge, Kritik...

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 06.08.2010 21:24

---

@Joe. Den Quarzosi 3,3V von [reichelt](#) gibt es nur in SMD.. die DIL's sind alle für 5V..? Soll dann die VGA+PS/2 Mimik nicht mit auf die Platine..?

(wobei das auch ok wäre.. und später macht man das zur MicroVGA kompatibel von den Anschlüssen her gesehen).

Ich frage mich nur, wie viele sich für weitere Platinen interessieren..? Bisher hat sich noch kein weiterer 'Nachbauer' geoutet.. ;-)

Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 07.08.2010 08:47

---

@Peter

Den Quarzoszillator als SMD hatte ich verworfen. Damit ist der schnelle Austausch nicht mehr möglich. Die Variante im DIL Gehäuse kann sehr gut auf eine IC Fassung gesteckt werden. Mögliche Quarzoszillatoren bei Rei\*\*\*\*t z.B. unter Artikel-Nr.: OSZI 25,000000. Meine Testvarianten laufen auch sauber mit 3.3V. Damit kann also nun die ganze Schaltung wahlweise mit 5V oder 3.3V betrieben werden, wobei die MMC natürlich nur mit 3.3V betrieben wird. Versehentliches "falschjumpfern" nicht mehr möglich. Die Test-VGA hat die gleiche Anschlussbelegung wie die MicroVGA, kommt also von hinten an die Platine. Die Schaltung folgt heute noch, bin noch nicht dazu gekommen. Wir könnten auch eine Platine machen, die an der VGA Seite getrennt werden kann.

Peter Sieg schrieb:

> Ich frage mich nur, wie viele sich für weitere Platinen interessieren..?

5 Stück werden es schon mal, schätze mit der VGA kommen noch einige Interessenten dazu.

@alle

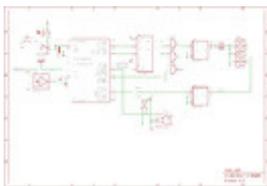
Wer hat Interesse an einer Platine? Bitte kurze Mail an mich.

Joe

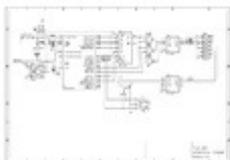
Autor: Joe G. ([feinmechaniker](#))

Datum: 07.08.2010 17:25

Angehängte Dateien:



- [AVR\\_VGA.png](#)  
17,9 KB, 478 Downloads



- [AVR\\_VGA.pdf](#) | [anzeigen](#)  
24,9 KB, 171 Downloads

Hier die Schaltung zur Diskussion. Sie geht auf eine Idee von Sebastian B. hier im Forum zurück (siehe [Beitrag "AVR VGA Terminal"](#))

Ich habe sie auf den ATmega88 portiert und um eine Farbdarstellung erweitert. Wobei das Wort Farbe nicht wörtlich zu nehmen ist. Es wird für einen Text immer nur EINE Farbe angezeigt. Der AVR hat nur exakt 8 Takte um das Zeichen in das Schieberegister zu bringen und da ist keine Zeit mehr für die Farbdarstellung jedes einzelnen Zeichens.

Joe

**Autor: hschuetz (Gast)**

**Datum: 07.08.2010 18:55**

---

Hallo Leute,  
ich bin noch eine Woche im Urlaub, dann will ich bauen!!  
Die Microvga hängt z.Z. am MC CPM Computer wo sie gut funktioniert!  
Welche Hardware Version sollte ich bauen und welche Software passt dann dazu?? Ich habe vor das System mit der Microvga auszurüsten und kein USB zu seriell chip einsetzen. Also bietet die fertige Platine Vorteile?  
oder ist es besser gleich auf Lochraster zu bauen?  
Vorschläge sind willkommen  
gruß  
Hans- Werner

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 07.08.2010 19:47**

---

Hallo und Willkommen Hans- Werner.

#### 1. Hardware

An Hardware gibt es zur Zeit die fast ursprüngliche Version von `sprite_tm`, für die Joe G. eine kleine Platinenaufgabe gemacht hat (vergriffen). Aktuell sind da nur 2 Leitungen vertauscht um Dramzugriff per Nibble zu ermöglichen.

Das kann man natürlich auch auf Lochraster aufbauen.. Schaltplan etc. hier weiter oben, im Artikel oder auf meiner Projektseite..

Z.Z diskutieren bzw. planen wir eine Version 2 der Hardware.

Erweiterungen:

2 x Dramchips mit 4MBit. Dadurch 8-bit Ramzugriff und viel Platz für Ramdisk.

MicroVGA direkt anschließbar bzw. sogar ein Selbstbauersatz dazu.

#### 2. Software

Z.Z ist Dank an Leo C. es fast schon schwieriger die jeweils aktuellste Version zu flashen ;-). Es gibt auf seiner Seite im trunk Zweig die aktuellste Version für die aktuelle 4-bit Dramversion. 8080 CPU Emulation

scheint fehlerfrei zu sein, bis zu 4 Partitionen (Typ=52=CP/M) werden als

LW a: - d: gemappt. Zusätzlich 64kb Ramdisk als i:

Im Zweig Software gibt es die Software für die neue 8-bit Dramversion, deren Platine gerade hier diskutiert wird..(2 x Dram Chips).

@Joe G. Das sieht doch prima aus mit dem VGATerm.. wie ich sehe ist da auch PS/2 schon dabei.. Sind schon ein paar ESC Codes dekodiert..(CLS)? Lt. Entwickler ist das eng.. Nun Vorteil ist zumindest schon mal, das man ein deutsches Tast.Layout einbauen kann.. Und evtl. schaffen wir es ja doch noch durch z.B Übertaktung (30-40MHz) zumindest CLS und CursorPos zu

implementieren.. ich würde vorschlagen, diese VGA+PS/2 Lösung mit so auf die Platinen zu nehmen, das sie wie die MicroVGA-Anschlüsse direkt zu funktioniert, wer aber doch MicroVGA möchte (oder nur das..?) dies direkt dort durchtrennen kann..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 08.08.2010 10:36

---

Peter Sieg schrieb:

> Und evtl. schaffen wir es  
> ja doch noch durch z.B Übertaktung (30-40MHz) zumindest CLS und  
> CursorPos zu  
> implementieren..

Die VGA kann leider nicht beliebig übertaktet werden, da der Pixeltakt fest steht (25 MHz) und starr mit dem [AVR](#) Takt verbunden sein muß. Deshalb auch der eigene Quarzoszillator auf der VGA. Ideal wären natürlich 50 MHz ;-). Da hätte man dann 8 zusätzliche Takte für den [AVR](#). An der Software muß natürlich gearbeitet werden, die Hardware stellt ja nur die Experimentierbasis.

Joe

Autor: Horst (Gast)

Datum: 08.08.2010 13:32

---

Moin moin,

bin gerade über das Projekt gestolpert.

Ich habe mal mit einem ISP8A600 angefangen.

Dann folgten 8080, Z80 und über QRL dann zwangsweise [6502](#)er.

Wobei der Z80 meine Lieblings CPU war.

Irgendwo müsste bei mir auch noch eigene CP/M Software sein.

Werde mal suchen. Hatte mal nen eigenen Compiler geschrieben.

Pascal ähnlich, UPN. Compiler erzeugte ASM-Code. Das ganze war für Steuerungen gedacht.

Ich würde mich gern an das Projekt mit drann hängen.

Leider sind hier keine DRAMs mehr vorhanden.

Wobei wird noch Unterstützung benötigt.

EAGLE, AVRSTUDIO (GCC,ASM) und ein bisschen was an Erfahrung sind vorhanden.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 08.08.2010 13:40

---

Horst schrieb:

> Ich würde mich gern an das Projekt mit drann hängen.

> Leider sind hier keine DRAMs mehr vorhanden.

> Wobei wird noch Unterstützung benötigt.

Willkommen im Team!

DRAM's sind genug da. Derzeit sehe zwei offene Baustellen.

- dem [AVR](#) Z80 Code beibringen

- die VGA programmieren (VGA, PS2, Terminal)

Autor: Peter Sieg ([petersieg](#))

Datum: 08.08.2010 14:01

---

@Horst: Auch von meiner Seite ein herzliches Willkommen!

Welche SOJ Dram's kommen jetzt höchstwahrscheinlich zum Einsatz?  
4M x 4 SOJ haben 26 Pin?  
oder doch die 1M x 4 SOJ mit 24 Pin (Auf der Platine zähle ich 24-Pin's)

Habe ihr dann mal ein paar Bezeichnungen passender Dram's..?  
Dann kann ich schon mal meine Bastelkiste durchsuchen..

Peter

**Autor: Horst (Gast)**

**Datum: 08.08.2010 14:06**

---

Moin mion,

OK, ich schau mal in die [AVR](#)-Soft rein.

Ich habe hier noch nen MYAVR Smart-Controller liegen.

Hat den schon jemand erweitert?

Ich werde wohl eher beim PC als Terminal bleiben.

VT50/100 müsste ich auch noch mit Pascal oder C-Quellen irgendwo rumliegen haben.

Wenn du noch DRAMS für mich hättest wäre das klasse joe.

Sonst würde ich eine Platine für 8x4164 machen, die sind noch zu bekommen.

Das ganze mit SRAM zumachen ist sicher nett, aber die Hardwareanpassung wird dann langsam etwas umfangreich.

Ich könnte z.B. ein Layout für meinen Hardwarebestand machen, MYAVR-Smart, 8x4164 plus schon festgelegter Hardware von euch. Hat da jemand Interesse drann?

Das Übertakten der [AVR](#)'s muss aus meiner Sicht nicht unbedingt sein.

Wenn ich wirklich Tempo haben will wäre eine andere CPU angesagt (ARM ab 400 MHz).

Netter wäre ein BUS-System um die weitere Hardware auch per [AVR](#) zu realisieren. Z.B. Ein [AVR](#) der Parallelport macht, einen für USB (per Software), acht bis zehnmal Servoansteuerung u.s.w. ES gibt doch schon so einiges was man da gut anflanschen könnte. Liese sich auch alles gut ins TINY-Basic ins BIOS einbauen.

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 08.08.2010 14:09**

---

[Peter Sieg schrieb:](#)

> Welche SOJ Dram's kommen jetzt höchstwahrscheinlich zum Einsatz?

> 4M x 4 SOJ haben 26 Pin?

4Mx4 SOJ kommen zum Einsatz. Die haben 24 Pins und werden von 1 bis 26, unter Auslassung der 7 und der 20, durchnummeriert. Ein typischer Vertreter ist z.B. HY5117404C

Joe

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 08.08.2010 14:16**

---

[Horst schrieb:](#)

> Wenn du noch DRAMS für mich hättest wäre das klasse joe.

Ram's kannst du bekommen, sie sind noch oft auf den alten Speicherriegeln zu finden.

>Netter wäre ein BUS-System um die weitere Hardware auch per [AVR](#) zu realisieren. Z.B. Ein [AVR](#) der Parallelport macht, einen für USB(per Software), acht bis zehnmal Servoansteuerung u.s.w. ES gibt doch schon so einiges was man da gut anflanschen könnte. Liesse sich auch alles gut ins TINY-Basic ins BIOS einbauen.

Hier kannst du dich auch austoben, Pin14/Pin15 der VGA-Schnittstelle sind mit I2C belegt (die einzigen freien Pins). Hier kann noch viel angeschlossen werden.

Joe

Autor: Leo C. ([rapid](#))

Datum: 08.08.2010 15:09

---

[Horst schrieb:](#)

> Moin mion,  
> OK, ich schau mal in die [AVR](#)-Soft rein.

Hallo Horst,  
genau wie Joe würde auch vorschlagen, daß Du Dir dann den 8080 Interpreter vornimmst. Bevor man ihn auf Z80 aufbohrt, müßte man mal schauen, ob sich an der Performance noch was machen läßt. Ich fürchte aber, das man das Teil nur schneller bekommt, wenn man **wesentlich** mehr Flash investiert. Und das ist bei Mega8 und Mega88 nicht mehr vorhanden.

> Ich werde wohl eher beim PC als Terminal bleiben.  
> VT50/100 müßte ich auch noch mit Pascal oder C-Quellen irgendwo rumliegen haben.

Bei der Gelegenheit könntest Du auch mal einen Wordstar auf VT100/ansi anpassen, oder einen angepaßten irgendwo auftreiben. Ich bin immer noch nicht dazu gekommen, und kopiere zur Zeit fleißig Dateien zwischen Host und Emulator bzw avrcpm hin und her.

> Sonst würde ich eine Platine für 8x4164 machen, die sind noch zu bekommen.

Bloß nicht! Viel zu iel Aufwand. Evtl. müßte auch die Software gändert werden. Und dann nicht mal eine RAM-Disk.

> Das ganze mit SRAM zumachen ist sicher nett, aber die Hardwareanpassung wird dann langsam etwas umfangreich.

> Ich könnte z.B. ein Layout für meinen Hardwarebestand machen,  
> MYAVR-Smart, 8x4164 plus schon festgelegter Hardware von euch. Hat da jemand Intresse drann?

Hoffentlich nicht. ;-)

> Netter wäre ein BUS-System um die weitere Hardware auch per [AVR](#) zu realisieren. Z.B. Ein [AVR](#) der Parallelport macht, einen für USB(per Software), acht bis zehnmal Servoansteuerung u.s.w. ES gibt doch schon so einiges was man da gut anflanschen könnte. Liesse sich auch alles gut ins TINY-Basic ins BIOS einbauen.

Kleine Erweiterungen sich sicher ok, aber ein Bus-System würde meiner

Meinung nach den Rahmen sprengen. --> anderes System.

Autor: Leo C. ([rapid](#))

Datum: 08.08.2010 16:40

Angehängte Dateien:

- [bios-m80.zip](#) (9 KB, 38 Downloads)

---

A>stat dsk:

A: Drive Characteristics  
1944: 128 Byte Record Capacity  
243: Kilobyte Drive Capacity  
64: 32 Byte Directory Entries  
64: Checked Directory Entries  
128: Records/ Extent  
8: Records/ Block  
26: Sectors/ Track  
2: Reserved Tracks

I: Drive Characteristics  
8128: 128 Byte Record Capacity  
1016: Kilobyte Drive Capacity  
192: 32 Byte Directory Entries  
0: Checked Directory Entries  
128: Records/ Extent  
16: Records/ Block  
32: Sectors/ Track  
2: Reserved Tracks

J: Drive Characteristics  
8192: 128 Byte Record Capacity  
1024: Kilobyte Drive Capacity  
192: 32 Byte Directory Entries  
0: Checked Directory Entries  
128: Records/ Extent  
16: Records/ Block  
32: Sectors/ Track  
0: Reserved Tracks

K: Drive Characteristics  
8192: 128 Byte Record Capacity  
1024: Kilobyte Drive Capacity  
192: 32 Byte Directory Entries  
0: Checked Directory Entries  
128: Records/ Extent  
16: Records/ Block  
32: Sectors/ Track  
0: Reserved Tracks

L: Drive Characteristics  
7680: 128 Byte Record Capacity  
960: Kilobyte Drive Capacity  
192: 32 Byte Directory Entries  
0: Checked Directory Entries  
128: Records/ Extent  
16: Records/ Block  
32: Sectors/ Track  
0: Reserved Tracks

So sieht das jetzt bei mir mit 2 4Mx4 Chips aus.  
Für jedes Megabyte ein Laufwerk.  
Vom ersten Laufwerk gehen 2 Systemspuren ab (8K), für CCP+BDOS, die von dort beim Warmstart neu geladen werden.  
Vom letzten Laufwerk gehen 64Kbyte ab. Schließlich braucht der Rechner ja auch etwas Arbeitsspeicher. ;-)

Im [AVR](#)-Teil sind 32 Sektoren pro Spur und maximal 1 MByte pro Laufwerk fest vorgegeben. Die obersten 64K werden immer als Arbeitsspeicher verwendet.

Der Rest ist über die CP/M Parametertabellen frei einstellbar.  
Dafür verwendet man am Besten die Macros, die ich hier mal angehängt habe.

Ich war so frei, die DR-Macros für unsere Zwecke etwas anzupassen.

Das ganze kommt auch noch ins [SVN](#), wenn ich etwas aufgeräumt und ein paar Build-Scripte erstellt habe.

Autor: Joe G. ([feinmechaniker](#))

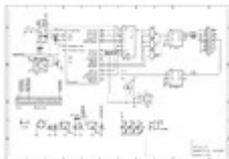
Datum: 08.08.2010 17:37

Angehängte Dateien:



• [AVR\\_CPM.png](#)

22,1 KB, 288 Downloads



• [avr\\_vga.pdf](#) | [anzeigen](#)

30,3 KB, 193 Downloads

---

Noch nicht ganz perfekt aber in diese Richtung wird es gehen...

- VGA kann abgetrennt und separat verwendet werden
- Stromversorgung über Steckernetzteil an VGA
- Bereitstellung von 5V und 3.3V

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 08.08.2010 20:00

---

@Joe: Rx und Tx liegen sich 1:1 gegenüber.. müßten die nicht gekreuzt sein?

Und wenn es geht würde ich die Verbindungen, die benötigt werden, wenn man beide zusammen betreiben will auch gleich durchverbinden.. dann muß man keine Jumper einlöten und verbinden..

Peter

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)  
**[Re: CP/M auf ATmega88](#)**

Autor: Horst (Gast)

Datum: 08.08.2010 21:03

---

moin moin,

war ein langer Tag heute..... mit Familie unterwegs. :-)

Wo ist oder wer hat die aktuelle Software des [AVR](#)?

Meine Kisten hatten früher ECB-BUS.

Der uVGA-Bus ist auch OK.

Gibt's schon eine Backplan und ein Gehäuse dafür?

Würde schon lustig ausschauen, so alles auf halben EURO-Kartenformat.

[AVR](#)-CPU-Board mit SD-FD und RAM-FD

zweites SD-FD mit RAM-FD

uVGA.

Parallele IO's.

Serielle IO's.

USB-Master.

Netzwerk.

RFM12-Board für Haussteuerung und ähnliches.

BT, wofür auch immer.

u.s.w

Und wenn dann mal einer Lust hat noch ein [ARM](#)-CPU-Board.

Kann was richtig grosses werden :-)

Autor: Leo C. ([rapid](#))

Datum: 08.08.2010 21:34

Angehängte Dateien:



[avrcpm-sdcase.jpeg](#)

112,3 KB, 490 Downloads

---

[Horst schrieb:](#)

> Wo ist oder wer hat die aktuelle Software des [AVR](#)?

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

Und hier unter 2. steht noch was dazu:

[Beitrag "Re: CP/M auf ATmega88"](#)

> Meine Kisten hatten früher ECB-BUS.

> Der uVGA-Bus ist auch OK.

> Gibt's schon eine Backplan und ein Gehäuse dafür?

Ja, siehe Bild. ;- ) [0]

> Würde schon lustig ausschauen, so alles auf halben EURO-Kartenformat.  
>  
> [AVR](#)-CPU-Board mit SD-FD und RAM-FD  
> zweites SD-FD mit RAM-FD  
> uVGA.  
> Parallele IO's.  
> Serielle IO's.  
> USB-Master.  
> Netzwerk.  
> RFM12-Board für Haussteuerung und ähnliches.  
> BT, wofür auch immer.  
> u.s.w

Wozu das alles auf einem simulierten 8080 mit C/PM?

> Und wenn dann mal einer Lust hat noch ein [ARM](#)-CPU-Board.

Für CP/M dann doch eher ein eZ80.

> Kann was richtig grosses werden :-)

Aber mit der Basis, an der wir basteln macht das imho keinen Sinn.

[1] Das ist ein "Schrottteil" von Pollin für einen Euro.  
<http://www.pollin.de/shop/dt/MTcwOTcyOTk-/Computer...>

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 09.08.2010 09:42**

---

Die neue Platine hat eigentlich drei Neuerungen:

1. 8-bit Dram -> damit ca. Faktor 2 im Dram-Zugriff = ca. doppelt so schnell
2. VGA-Terminal (ala MicroVGA) direkt mit auf der Platine (trennbar)
3. I2C Schnittstelle

Einzig 3. eröffnet erweiterte Möglichkeiten.. Hier würde mich Interessieren,  
was für Möglichkeiten man hier sieht..(Ideen)?

Ansonsten fehlt mir noch eine elegante Möglichkeit mal eine Datei zw.  
CP/M Welt und Host-Welt auszutauschen..

Gäbe XModem (muß noch 8080 Version finden). Ohne MicroVGA sollte das  
direkt klappen.. aber mit wird die serielle Schnittstelle verwendet..  
MicroVGA hat zwar einen 'Pass Through' Modus.. aber ob das klappt..?  
Was gäbe es sonst noch für Möglichkeiten../Ideen..?

Peter

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 09.08.2010 11:38**

**Angehängte Dateien:**



- [AVR\\_CPM.png](#)

33,8 KB, 442 Downloads



- [V24\\_Varianten.pdf](#) | [anzeigen](#)

17,6 KB, 162 Downloads

---

[Peter Sieg schrieb:](#)

> Rx und Tx liegen sich 1:1 gegenüber.. müßten die nicht gekreuzt  
> sein?  
> Und wenn es geht würde ich die Verbindungen, die benötigt werden, wenn  
> man beide zusammen betreiben will auch gleich durchverbinden.. dann muß  
> man keine Jumper einlöten und verbinden..

Ich glaube es ist schon korrekt so, trotzdem bitte nochmal Schaltung und Layout überprüfen. Die Belegung ist recht "tricky" deshalb hier mal eine Übersicht dazu. Das sollte im späteren Betrieb auch Einstellung erleichtern.

Wie ja schon weiter oben beschrieben, können neben der Kommunikation des CP/M mit der VGA auch die Varianten CP/M an PC Terminal oder VGA an PC Terminal eingestellt werden. Gerade die letzte Variante schein mir zum Test und zur Inbetriebnahme der VGA recht nützlich. Außerdem verseehe ich die [AVR](#) im Normalfall mit einem Bootloader, so das über die USB der jeweilige Controller programmiert werden kann.

Die Verbindungen CP/M VGA sind schon auf der Platine, sie liegen auf der Unterseite.

> Ansonsten fehlt mir noch eine elegante Möglichkeit mal eine Datei zw.  
> CP/M Welt und Host-Welt auszutauschen..

Dir gefällt wohl die Variante Vinculum usb nicht? Das CP/M kennt ja derzeit noch keine COM xx. Also im [AVR](#) die SPI für Vinculum usb programmieren und im BIOS als COM xx anmelden.

@Ronny Minow

Ein Bausatz im Sinne von BAUSATZ ist eigentlich nicht geplant. Es wird die Platinen und wieder einige Bauelemente (MMC Fassung, SMD IC's, [AVR](#), Quarze..) geben.

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 09.08.2010 12:47**

---

[Joe G. schrieb:](#)

> Dir gefällt wohl die Variante Vinculum usb nicht?

Doch. Nur kann ich mir die genaue Funktion noch nicht so richtig vorstellen..

Es wird ein USB Stick mit FAT unterstützt. Per seriellen Befehlen wird das ganze gesteuert.. Also stelle ich mir das so vor, das im [AVR+Bios](#)

eine 2te serielle Schnittstelle implementiert werden müßte. Dazu dann ein passendes Steuerprogramm auf CP/M Ebene das letztlich auch eine Datei vom USB Stick lesen und auf CP/M Diskette schreiben kann (und umgekehrt..) Als Alternative könnte auch ein Xmodem diese 2te serielle IF nutzen um z.B mit einem PC Daten auszutauschen.. Richtig..?

Peter

Autor: Detlev T. ([detlevt](#))

Datum: 09.08.2010 12:47

---

[Joe G. schrieb:](#)

> Noch nicht ganz perfekt aber in diese Richtung wird es gehen...

Da hätte ich noch ein paar Verbesserungsvorschläge:

- 1.) Statt des [74HC165](#) sollte man eher ein [74HC166](#) nehmen bei dem das Laden mit dem Takt synchronisiert ist.
- 2.) Mir leuchtet nicht ganz ein, wozu die [74HC244](#) Treiber nötig sind. Laut Philips/NXP Datenblatt könnte ein 74HC-Standardausgang problemlos 4mA treiben, das würde sogar für alle drei Farben gleichzeitig reichen:

$$I = 3 \times \frac{5V}{470\Omega + 75\Omega} = 2.75mA$$

Gruß, DetlevT

Autor: Horst (Gast)

Datum: 09.08.2010 15:44

---

Moin moin,

Ich hab mal einen Blick in die Software getan. Schön, aber nicht auf Tempo optimert. Da kann man noch reichlich arbeit reinstecken.

1.) Für die Z80 Codes EXX und EX AF,AF' muss noch ein zweiter Registersatz her. Ich würde den ins normale SRAM des [AVR](#) legen. Diese Befehle sind nicht so häufig verwendet worden. Die relativen Sprungbefehle sollten kein Problem sein.

2.) Die IN und OUT würde ich wahlweise auf einen der seriellen Busse umlegen.

So hat man die möglichkeit damit eigene Hardware zu steuern. Z.B. könnte man über I<sup>2</sup>C oder so PIO's und SIO's anhängen. Ein Mega8 tut per software so als ob er eine PIO oder SIO wäre.

Besteht die möglichkeit die uVGA so auch zu steuern?

3.) Für die Interrupt Codes (HALT, u.s.w) bräuchte man noch eine freie Portleitung. Ich würde die SD-Floppy auf eine eigene Platine legen ( zwei Drives) und an den ser. BUS hängen. So wäre wieder reichlich Hardware frei.

Das alles würde auch das BIOS vereinfachen.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 09.08.2010 16:35

---

[Detlev T. schrieb:](#)

> Statt des [74HC165](#) sollte man eher ein [74HC166](#) nehmen

Gute Idee, schon geändert.

>Mir leuchtet nicht ganz ein, wozu die [74HC244](#) Treiber nötig sind.

Das waren Relikte aus der s/w Darstellung - auch geändert.

Peter Sieg schrieb:

> Also stelle ich mir das so vor, das im [AVR+Bios](#)  
> eine 2te serielle Schnittstelle implementiert werden müßte. Dazu dann  
> ein passendes Steuerprogramm auf CP/M Ebene das letztlich auch eine Datei  
> vom USB Stick lesen und auf CP/M Diskette schreiben kann (und  
> umgekehrt..) Als Alternative könnte auch ein Xmodem diese 2te serielle  
> IF nutzen um z.B mit einem PC Daten auszutauschen.. Richtig..?

Ja Peter, genauso dachte ich.

Autor: Detlev T. ([detlevt](#))

Datum: 09.08.2010 18:34

---

Hallo Joe,

ich bin zwar nicht direkt ein Interessent für Platinen, aber die  
VGA-Geschichte interessiert mich.

Mir ist da eine Unstimmigkeit aufgefallen. Du beziehst dich auf das  
[AVR-VGA-Projekt](#), dass 50 Zeilen a 80 Spalten hat. Das macht einen  
SRAM-Bedarf von 4000 Bytes. Eingeplant hast du ein [ATMEGA88](#) mit 1024  
Bytes.

Es gibt mit dem 328p eine Version mit 2kiB, damit würde man noch 25  
Zeilen schaffen, aber niemals 50. Das dürfte der Grund gewesen sein,  
warum das Original-Projekt den Mega644 (4kiB) nutzt.

Außerdem könnte man mit einem "großen" ATMEGA alle 8 Bit des  
Schieberegister nutzen und mit einem zweiten Port, einem 4-fach 2 zu  
eins Multiplexer (statt AND) und dem DAC wie bei Jörg Wolframs Chipbasic  
für Vorder- und Hintergrundfarbe je eine aus 16 auswählen ohne mehr ICs  
zu brauchen.

Gruß, DetlevT

Autor: Peter Sieg ([petersieg](#))

Datum: 09.08.2010 19:41

---

@Joe: Ich denke wir sollten jetzt auch nicht zu schnell die Platinen  
ordern..

Ein paar Tage mehr zum drüber nachdenken, sollte nicht schaden.. Ich  
persönlich finde, das jetzt ohne den VGA-Terminal Part, unser avr cpm  
eigentlich nur von 4 auf 8-bit Ram gewachsen ist (mit dann größerer  
Ramdisk).. ich bin noch nicht gänzlich überzeugt, ob wir nicht mit einem  
größeren [AVR \(ATmega32/644/1284\)](#) mehr gewinnen, mit dem ATmegaX8 sind  
wir mit den Pin's am Ende..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 09.08.2010 19:56

Angehängte Dateien:

- [avrcp\\_m.zip](#) (111,6 KB, 64 Downloads)

---

@Peter

Ich habe kein Problem noch einige Tage damit zu warten. Einen größeren [AVR](#) würde ich auch nicht nehmen. Der Reiz besteht ja gerade darin mit einem sehr einfachen System zu arbeiten. Wenn der Programmspeicher nicht reicht gibt es ja noch den ATmega168. Benötigt man noch weitere Pins, so lässt sich das System bequem über I2C erweitern.

@alle

Hier die letzte Schaltungs- und Layoutversion. Ich würde nun erst mal Änderungen, Kritik, Wünsche oder Zustimmungen (Ablehnungen) sammeln, ehe ich sie noch mal anfasse.

Joe

Autor: Leo C. ([rapid](#))

Datum: 09.08.2010 20:03

---

[Horst schrieb:](#)

> Ich hab mal einen Blick in die Software getan. Schön, aber nicht auf  
> Tempooptimert. Da kann man noch reichlich arbeit reinstecken.

Wunderbar, dann kanns ja losgehen.

> 1.) Für die Z80 Codes EXX und EX AF,AF' muss noch ein zweiter  
> Registersatz her. Ich würde den ins normale SRAM des [AVR](#) legen.

Klar, und die restlichen, fehlenden natürlich auch. In der 8-Bit Variante sind eh schon einige Register ausgelagert.

> Diese  
> Befehle sind nicht so häufig verwendet worden. Die relativen  
> Sprungbefehle sollten kein Problem sein.

>  
> 2.) Die IN und OUT würde ich wahlweise auf einen der seriellen Busse  
> umlegen.  
> So hat man die möglichkeit damit eigene Hardware zu steuern. Z.B. könnte  
> man über I<sup>2</sup>C oder so PIO's und SIO's anhängen. Ein Mega8 tut per  
> software so als ob er eine PIO oder SIO wäre.  
> Besteht die möglichkeit die uVGA so auch zu steuern?

>  
> 3.) Für die Interrupt Codes (HALT, u.s.w) bräuchte man noch eine freie  
> Portleitung. Ich würde die SD-Floppy auf eine eigene Platine legen (  
> zwei Drives) und an den ser. BUS hängen. So wäre wieder reichlich  
> Hardware frei.

>  
> Das alles würde auch das BIOS vereinfachen.

Wir freuen uns auf Deine Verbesserungen und Erweiterungen.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 09.08.2010 20:17

---

[Detlev T. schrieb:](#)

> Mir ist da eine Unstimmigkeit aufgefallen. Du beziehst dich auf das  
> [AVR](#)-VGA-Projekt, dass 50 Zeilen a 80 Spalten hat. Das macht einen  
> SRAM-Bedarf von 4000 Bytes. Eingeplant hast du ein [ATMEGA88](#) mit 1024  
> Bytes.

Der [ATmega88](#) steht in der Schaltung nur stellvertretend (EAGLE Footprint) für den ATmega328. Zum Einsatz kommen soll tatsächlich der 328. Das gibt dann genau die von dir genannten 25 Zeilen zu 80 Spalten.

Ja, es könnten mehr Ports, ein DAC, ein Multiplexer... aber auch ein [CPLD](#) oder ein [FPGA](#) verwendet werden. Ich habe aus zwei Gründen darauf verzichtet.

1. Das [AVR](#) CP/M ist bewusst mit sehr einfacher Hardware aufgebaut, sozusagen ein minimalistisches System. So soll es auch bleiben. Mich schmerzt eigentlich schon das dritte IC. Es ist keine Konkurrenz für [ARM](#)'s, eZ80, ... Ein Schüler mit Bastelerfahrung und einer kleinen Kramkiste soll es aufbauen können und ein vollwertiges Betriebssystem mit allen dazugehörigen Tools nutzen können.

2. Die VGA soll die gleichen Bedingungen erfüllen. Bunte Buchstaben (außer grün auf Schwarz ;-)) machen da eigentlich keinen Sinn.

3. Mit der microVGA gibt es schon etwas für diejenigen die mehr wollen.

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 10.08.2010 09:53

Angehängte Dateien:

- [partlist.txt](#) (5,9 KB, 229 Downloads)

---

Ich hänge hier mal die partlist aus eagle dran..

Hier ein Auszug ohne C, R, Jumper etc.:

Part	Value	Package	Library
B1	DBDB101G	DB	rectifier
CON1	SCDA5A0201	SCDA5A0201	SCDA5A0201
IC1	ATmega88-20P	DIL28-3	atmel
IC2	74S08D	SO14	74xx-eu
IC3	FT232RL	SSOP28	ftdichip
IC4	ATmega88-20P	DIL28-3	atmel
IC7	74AC08D	SO14	74xx-eu
IC8	LF50CDT	DPACK	v-reg
IC9	LF33CDT	DPACK	v-reg
IC10	74HC166D	SO16	74xx-eu
J1	DCJ0202	DCJ0202	con-jack
LED1	Grün	LED3MM	led
LED2	Gelb	LED3MM	led
LED3	Grün	LED3MM	led
QG1	25 MHz	DIL14S	crystal
QG2	25 MHz	DIL14S	crystal
U\$1	DRAM_4M_4	SOJ26-24	AMESYS
U\$2	DRAM_4M_4	SOJ26-24	AMESYS
X1	PN61729-S	PN61729-S	con-berg
X2	PS2	SSV-BC06	con-yamaichi
X3	VGA	HDF15H	con-subd

Dazu ggf. noch ein paar Hinweise..

[FT232RL](#) und entsprechende Bauteile + USB Buchse braucht man nur, wenn man diese PC Verbindung auch nutzen möchte..

Ich gehe davon aus, das der SD Slot auch mit der Platine geliefert werden kann (wie beim letzten mal..)

Der [AVR](#) CP/M läuft bei mir stabil mit 30MHz. In ein paar Tagen bekomme ich 40MHz Quarzoszillatoren und weiß dann, ob es auch damit geht.. d.h, wer seinen [AVR](#) CP/M höher takten möchte, sollte anstatt 1x25MHz dann einen höher getakteten Quarzosi bestellen..

Peter

## Benchmark Übersicht

Autor: Peter Sieg ([petersieg](#))

Datum: 10.08.2010 10:30

Ich habe hier noch mal verschiedene 'Meßwerte' aus weiter oben zusammen getragen:

		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 8-Bit Variante.

D.h z.Z liegt man bei 4-bit und Übertaktung bei ca. 1,3MHz 8080 Speed. Das erreicht man ebendso mit 8-bit Dram Ansteuerung ohne Übertaktung. Mit 8-bit + Übertaktung sind ca. 2MHz 8080 Speed drin.. Und noch weitere Optimierungen des Codes kommen on Top.

Peter

## Erste Z80 Erweiterungen

Autor: Horst S. ([h3aau](#))

Datum: 12.08.2010 01:29

---

Moin moin,

Die ersten Aenderungen für den Z80-Emu sind fertig.

Neuer opcode-Interpreter  
EX AF,AF'  
EXX  
Relative Jumps

Da ich noch keine Hardware habe bräuchte ich mal jemanden der die Software testen kann. Wer hat Lust und für welchen [AVR](#) soll ich es compilieren?

Autor: Peter Sieg ([petersieg](#))

Datum: 12.08.2010 09:30

---

[Horst S. schrieb:](#)

```
> Neuer opcode-Interpreter  
> EX AF,AF'  
> EXX  
> Relative Jumps
```

Das alles sagt mir nicht soo viel ;-)  
Ist der z80 jetzt damit komplett drin..?

Generell kann ich schon testen.. aber bei dem kleinsten Problem.. geht das dann wie ein Ping-Pong Spiel los.. ist halt sicher nicht ideal ohne eigene Hardware.. Hast du das für die 4-bit Dram oder die 8-bit Dramversion gemacht?

Außer Leo C. dürfte z.Z wohl eher niemand die 8-bit Hardware haben.. ;-)

---

Nun, ich habe die 4-bit Version (3x) hier. 30MHz. 38400 Baud.  
Ich habe Sargon.com (Schachprogramm) auf einer 'Diskette'. Das Programm steigt mit illegal Opcode aus, da es vermutlich z80 Code enthält..

---

Alternativ kann ich auch 1 x 4-bit Hardware verkaufen (-> Mail).

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 12.08.2010 09:47

---

Die originalen VGA Quellen findet ihr nun auch hier:

<http://www.mikrocontroller.net/svn/list>

Autor: Horst S. ([h3aau](#))

Datum: 12.08.2010 13:13

---

Moin moin,

@joe

>Sende mir doch einfach die Quelle zu und ich stelle sie über [SVN](#) bereit.

danke, aber so völlig ungetestet möchte ich das nicht in die weite welt verteilen. wenn es zur zeit nur einen mit 8bit HW gibt wäre es wohl ganz gut wenn ich meine änderungen auch in die 4bit SW einbaue. ich bin nach schlechten erfahrungen bei anderen projekten kein freund von svn mehr.

kannst du die SW irgend wo für einen normalen download bereitstellen?

@Peter Sieg

danke fürs angebot, sende mir das mal als PM zu.

ich hätte schon gern die 8Bit hardware wann sind die platinen den so weit?

ich lasse meine sonst immer beim platinenbelichter machen. ist die DRAM verfügbarkeit/beschaffung eigentlich schon geklärt?

ich denke ich habe hier die SW als 8bit version.

leider ist in der SW selber keine doku und/oder change history drinn.

**Autor: Leo C. ([rapid](#))**

**Datum: 12.08.2010 16:52**

---

Hallo Horst,

prinzipiell läuft Dein Interpreter. So siehts im Moment aus:

CPM on an AVR, v1.0

Testing RAM: fill...wait...reread...

Initing mmc...

CP/M partition at: 001, size: 7811KB.

CP/M partition at: 15624, size: 7812KB.

CP/M partition at: 31248, size: 7812KB.

Partinit done.

Ok, CPU is live!

ipl

62k cp/m vers 2.2

Danach sollte der CP/M Prompt "A>" kommen....

Der Bootloader (ipl) wird ausgeführt, der Anfang vom BIOS.

Im Bios hängt das Programm zuerst in der Console-Inputschleife (wo es garnicht hinkommen sollte), und danach endlos in der Bootschleife, weil der avr-teil beim Sektor lesen immer einen Fehler zurückmeldet.

Ich habe den Verdacht, das bei call/return etwas schief läuft.

[Horst S. schrieb:](#)

> wenn es zur zeit nur einen mit 8bit HW gibt wäre es wohl ganz

> gut wenn ich meine änderungen auch in die 4bit SW einbaue.

Im Moment scheine ich der einzige zu sein, der die 8-Bit Variante aufgebaut hat. Es reicht aber, wenn Du Deine Software einmal schreibst. Vorläufig kann ich sie dann in die verschiedenen Versionen integrieren.

Auf längere Sicht möchte ich aber nur noch eine Version haben, allerdings modularisiert. Vielleicht sollte ich das jetzt endlich mal machen.

> ich bin nach

> schlechten erfahrungen bei anderen projekten kein freund von svn mehr.

Ohne Versionsverwaltung wirds aber echt schwierig mit mehreren Leuten an einem Projekt. Oder richtet sich Deine Aversion speziell gegen Subversion, und eine andere Versionsverwaltung wäre ok?

> ich denke ich habe hier die SW als 8bit version.

> leider ist in der SW selber keine doku und/oder change history drinn.

Die change history ist im svn.  
Die Doku ist hier im Thread. ;-)  
Ansonsten: "Use the source luke!" ;-)  
Im Source sind ganz oben die Ports definiert. Dort kannst Du sehen, das für das RAM nur 4 Datenbits definiert sind.

Die aktuellen Quellen sind hier:

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

Das ist die "stabile" Version für die aktuelle (4-Bit-RAM) Hardware.

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

Hier ist z.Zt. die Version für 8-Bit RAM.

Sorry für svn.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 12.08.2010 17:01

---

[Leo C. schrieb:](#)

> Die Doku ist hier im Thread. ;-)

Es muss ja nicht jeder programmieren oder Hardware entwerfen. Vielleicht findet sich hier einer der sich mal an eine ordentliche Doku macht.

Gibt es noch Interessenten für die neue LP? Alle von mir schon Erfassten sollten im Mailverteiler sein, aber möglicherweise habe ich noch jemanden vergessen. Einfach bei mir melden.

Autor: Hans- w. Schütz ([hschuetz](#))

Datum: 12.08.2010 17:43

---

Hallo Leute,  
wenn mein System endlich läuft werde ich es dokumentieren und alles auf meiner Webseite veröffentlichen....  
Wer seine Softwareversionen dann mit veröffentlichen will...  
kein Problem.  
Webseite schuetz.thtec (dot)org  
Gruß  
Hans- Werner

Autor: Horst S. ([h3aau](#))

Datum: 15.08.2010 09:51

---

moin moin  
mal eine frage an die asm-profis hier.

.dw label erzeugt 2byte mit der adr des label. so weit ok.  
.db low(label) erzeugt leider auch 2byte und eine warnung.  
beides möchte ich aber nicht, sonder nur das lowbyte der adr.

ich arbeite mit avr-studio.  
schon mal danke.

Autor: Horst S. ([h3aau](#))

Datum: 18.08.2010 21:26

Angehängte Dateien:

- [AVR\\_8080\\_4Bit.zip](#) (29,1 KB, 55 Downloads)

---

Moin moin,

Die 8080 Version ist so weit fertig.  
Der Code passt noch locker in einen [ATMega8](#).  
Es ist noch Platz für zusätzliche Wünsche.

Der Z80 hat über 700 Befehle, das passt leider nicht mehr in die 8K. Hier für ist dann ein ATMega168 nötig.

Im wesentlichen habe ich den Opcode-Interpreter etwas auf Tempo gebracht und für den Z80 vorbereitet.

Wer weiter mit arbeiten möchte kann sich direkt bei mir melden.

Danke noch mal an Leo fürs Testen da ich selbst noch keine Schaltung habe.

Wie schaut es mit der Hardware aus?  
Neue Ideen, neue Wünsche?

Autor: Leo C. ([rapid](#))

Datum: 19.08.2010 01:49

Angehängte Dateien:

- [avrcpm-1.asm.zip](#) (43,9 KB, 40 Downloads)

---

[Horst S. schrieb:](#)

> Die 8080 Version ist so weit fertig.

Wenn Du meinst. ;-)

Wie wärs, wenn man z.B. die Tabellen über Makros aufbauen würde. Z.B. so:

```
; instr fetch, op, store  
;
```

```
.macro instr  
  .db low(@2), low(@1), high(@1), low(@0)  
.endm
```

```
;-----
```

```
instr do_fetch_DIR16, do_nop, do_store_BC ;01 nn nn ;LD BC,nn  
instr do_fetch_DIR8, do_op_CPFA, do_nop ;FE nn ;CP n
```

Nur mal zwei Befehle als Beispiel. Die Tabellen wären übersichtlicher, und bei Änderungen an der Tabellenstruktur müsste man nur das Makro ändern.

> Im wesentlichen habe ich den Opcode-Interpreter etwas auf Tempo gebracht und für den Z80 vorbereitet.

So sieht es jetzt aus:

```
Zeit "Speed" HW Interpreter
```

[s]	[KHz]		
3.364	857	4	alt
3.173	1088	4	neu (Horst S.)
1.924	1499	8	alt
1.753	1645	8	neu (Horst S.)

Hardware mit 4bit/8bit-RAM, jeweils bei 20MHz.

>  
 > Wer weiter mit arbeiten möchte kann sich direkt bei mir melden.  
 >  
 > Danke noch mal an Leo fürs Testen da ich selbst noch keine  
 > Schaltung habe.

Wie man an der Tabelle sieht, läuft es jetzt auch mit der neuen Hardware.

Ich habe nochmal eine Kleinigkeit am Interpreter ändern müssen, weil die fetch- und store-routinen (dank der RAM-Makros in der 8-Bit Version) nicht mehr zusammen in eine Page passen.

Ich habe angefangen, das Programm in mehrere Dateien zu zerlegen. Die Struktur, wie das Ganze werden soll, kann man sicher schon erkennen. Es gibt jetzt einen neuen Schalter, mit dem man die 8-Bit oder 4-Bit auswählen kann. Weitere Schalter, zb. für die Größe des RAMs, sind geplant. Die RAM-Disk ist z.Zt. gerade abgeschaltet. Wer (Verbesserungs-) Vorschläge hat, bitte (hier) melden. Ich suche z.B. noch nach einer Möglichkeit, die DRAM-Makros zu verunübeln.

Die neueste Version habe ich hier angehängt. Demnächst kommt sie auch ins [SVN](#). Z.Zt. in branches/modules. Später, wenn alle damit glücklich sind, nach trunk.

Gute Nacht

**Autor: Horst S. (Gast)**  
**Datum: 19.08.2010 09:03**  
**Angehängte Dateien:**

- [todo table macro.txt](#) (14,7 KB, 135 Downloads)

---

Moin Moin,

Frühstückspause!  
 und schnell die Tabelle gebaut. Gut das ich die in Excel verwalte.  
 Baust du das noch mit ein Leo? Danke.  
 Ich kümmere mich weiter um den Z80.

Danke.  
**Autor: Peter Sieg ([petersieg](#))**

**Datum: 19.08.2010 09:07**

---

@Leo C + Horst S: SUPER!! Herzlichen Dank! Mit euch kommt das Projekt softwaretechnisch wirklich in 'ungeahnte Dimensionen' ;-)  
 (..and boldly go were no one has gone before..)

Prima!  
 (Habe gerade mal die 4-bit Version durch den Assembler gejagt;

heute nachmittag kann ich es erst den [AVR](#) flashen.. dann mehr..)

Peter

Autor: h3aau (Gast)

Datum: 19.08.2010 09:08

Angehängte Dateien:

- [todo\\_table\\_macro\\_A.txt](#) (14,7 KB, 120 Downloads)

---

und hier noch mit vertauschten fetch und store.....

Autor: Leo C. ([rapid](#))

Datum: 19.08.2010 17:46

Angehängte Dateien:

- [avrcpm-2.zip.zip](#) (45,6 KB, 38 Downloads)
- [avrcpm-2.zip](#) (45,6 KB, 45 Downloads)

---

[Peter Sieg schrieb:](#)

> ok. Sorry.

Ich habe halt etwas länger suchen müssen. Alle "sinnvollen" Kombinationen liefen bei mir. Mega328 mit 4-Bit ist bei mir nicht sinnvoll, weil ich gerade keine Hardware dafür habe. ;-) Ob's durch den Assembler läuft, teste ich aber trotzdem gelegentlich.

So wie im Anhang, müßte es jetzt gehen. Mit dieser Lösung bin ich aber keinesfalls glücklich. Wer das ganze Paket mal in Form bringen will, soll sich keinen Zwang antun...

Auf längere Sicht wird man bei den größeren Prozessoren nicht mehr ohne absolute Sprünge auskommen. Die können ja nicht "hinten rum" springen, wie der 8 oder 48.

> Und von 7,589s auf 7,004s runter ;-) (30MHz)

More to come... stay tuned.

Edit: sorry für die zip.zip Datei. Kann man so ein Versehen nicht wieder löschen?

Autor: Leo C. ([rapid](#))

Datum: 19.08.2010 19:41

---

[Horst S. schrieb:](#)

> ich suche noch jemanden der die todo\_tabs für die Z80-oopcodes füllt.  
> ist eine fleissarbeit, würde aber helfen.  
> hat jemand lust?

Gut, daß Du das selber hier reinstellst, ich wollte Deine E-Mail schon umleiten. ;-)

[Peter Sieg schrieb:](#)

> @Horst S.: Hänge es mal hier rein und sage was ich machen muß.. dann  
> kümmere ich mich morgen drum..

Das ist eine gute Frage. Mir ist das auch nicht klar. Das dürfte nämlich sehr stark vom Interpreter abhängen, der ja auch noch nicht da ist.

Die wenigen Befehle, die in der jetztigen Tabelle fehlen, wird Horst nicht meinen.

Für 0xDD und 0xFD braucht man imho gar keine Tabellen. Das sind Umschaltpräfixe für Befehle mit H, L, HL, (HL) als Operanden.

0xED ist nur schwach besetzt. Da wird man vielleicht eine irgendwie komprimierte Tabelle machen.

CB ist sehr regelmäßig. Da kann man wahrscheinlich auch was machen, um nicht eine 3 oder 4 mal 256 Byte Tabelle anlegen zu müssen.

Andererseits, wenn man mit größerem Prozessor mindestens 8KByte zusätzlichen Tabellenplatz hat...

> @Leo C.: 168 Zweig assembliert ohne Fehler und läuft auch. Prima!

Danke fürs Testen.

Autor: Horst S. ([h3aau](#))

Datum: 19.08.2010 20:00

Angehängte Dateien:

- [DOTO\\_TAB\\_4byte\\_macro.xls](#) (563,5 KB, 121 Downloads)

---

Ist eigentlich recht einfach.

```
do_fetch_xxx      ; woher kommen die daten LD ??,A = do_fetch_A
do_store_xxx      ; wo sollen sie hin      LD A,?? = do_store_A
```

im zweifelsfall einfach do\_???? eintragen.

```
do_op_xxx         ; hier steck die eigentliche finktion
```

im zweifelsfall bei do\_op\_xxxx einfach do\_op\_inv eintragen.

Autor: Peter Sieg ([petersieg](#))

Datum: 20.08.2010 08:48

Angehängte Dateien:

- [DOTO\\_TAB\\_4byte\\_macro.xls](#) (565 KB, 127 Downloads)

---

Hier mal einige Einträge im Tab: cb + ed.

Bitte prüfen, ob ich das so richtig verstanden+gemacht habe..?  
Dann ziehe ich auch die anderen durch..

Peter

Autor: Horst S. (Gast)

Datum: 20.08.2010 12:13

---

Danke Peter,  
Schaut schon ganz gut aus.  
Ich werde zum Wochenende mal eine Rohversion für den Z80 fertig machen  
und an Leo senden.  
Soll ich mal eine Beschreibung machen wie der Opcode-Interpreter nun  
arbeitet?

Gruus Horst

Autor: Peter Sieg ([petersieg](#))

Datum: 20.08.2010 12:49

---

[Horst S. schrieb:](#)

> Soll ich mal eine Beschreibung machen wie der Opcode-Interpreter nun  
> arbeitet?

Aber immer und gerne!

Den Aufruf von Leo C. zur Dokuerstellung habe ich auch nicht vergessen..  
nur  
macht das erst Sinn, wenn das Projekt sich noch etwas mehr 'gesetzt'  
hat.. z.Z ist noch viel zu viel im Umbruch (zum Besseren = Gut so!)..

Peter

Autor: Leo C. ([rapid](#))

Datum: 20.08.2010 13:41

Angehängte Dateien:

- [avrcpm-3.zip](#) (67,3 KB, 51 Downloads)

---

Im Anhang sind meine neuesten Spielereien. Es sind zwei neue  
Interpretervarianten. Hier mal eine Tabelle mit den Unterschieden:

Interpreter	End	[.cseg]			avrdude	Laufzeit	"Speed"
		Code	Data	Used		[s]	[KHz]
z80int	0x001900	5052	1178	6230	6400	1.924	1499
8080int	0x001b00	4848	1690	6538	6912	1.733	1644
8080int-t3	0x001b00	4936	1434	6370	6912	1.733	1644
8080int-t3-jmp	0x001d00	5240	1434	6674	7424	1.366	2111

-t3 heißt, das die Opcode-Tabelle von 4 auf 3 Byte je Eintrag reduziert  
wurde. Spart 256 Byte Tabellenplatz, kostet aber 88 Byte zusätzlichen  
Code für eine Sprungtabelle. Witzigerweise genau gleiche Laufzeit, wie  
die 4-Byte Variante. Eine Optimierung ist mir dann aber noch  
eingefallen. War eigentlich für die -t3-jmp Variante gedacht, dort aber  
nicht anwendbar.;

8080int-t3	0x001b00	4934	1434	6368	6912	1.712	1684
------------	----------	------	------	------	------	-------	------

-t3-jmp: Statt push/return in der Interpreterschleife wird zwischen den  
Phasen 'load', 'do\_op' und 'store' direkt gesprungen. Kostet Platz, der  
aber noch vorhanden ist.

Meinungen würden mich interessieren.

[Peter Sieg schrieb:](#)

> Den Aufruf von Leo C. zur Dokuerstellung habe ich auch nicht vergessen..

Der (konstruktive) Vorschlag war von Jörg.  
(Nach einer ironischen Bemerkung von mir zum Thema.)

Autor: Leo C. ([rapid](#))

Datum: 20.08.2010 14:04

---

[Peter Sieg schrieb:](#)

> Hier noch der Rest, den ich soweit gefüllt habe..

@Peter, @Horst:

> do\_op\_ADC

> do\_op\_SBC

Nach der bisherigen Nomenklatur müßten diese wohl do\_op\_ADCHL und do\_op\_SBCHL heißen.

Aber in den DD/FD-Tabellen gibts ja auch noch IX und IY als Zielregister.

Vorschlag:

op\_ADDHL --> op\_ADD16

op\_ADC --> op\_ADC16

op\_SBC --> op\_SBC16

Allgemeiner Vorschlag:

In den (Excel-) Tabellen grundsätzlich

do\_fetch\_xxx --> fetch\_xxx

do\_op\_xxx --> op\_xxx

do\_store\_xxx --> store\_xxx

Das "do\_" wird (nur) bei Bedarf über die Makros wieder zugefügt.

Autor: Horst S. (Gast)

Datum: 20.08.2010 15:11

---

Moin moin,

Leo. schrieb:

< Meinungen würden mich interessieren.

Guter Trick das vom Stack in Register zu verlegen.

Damit bin ich bei meiner Frage die ich dir eigentlich zusammen mit der Z80-Rohversion zusenden wollte.

Wie ist den nun die Registeraufteilung für den Z80?

Eigentlich wollte ich die neu festlegen und dort einiges aufräumen.

Die bestehende Aufteilung ist für den 8080 ja ok.

Kann man eigentlich noch Code in den DRAM-Routinen sparen?

Hier wird doch auch reichlich Zeit verbraten.

Gruss Horst

Autor: Leo C. ([rapid](#))

Datum: 20.08.2010 16:27

---

[Horst S. schrieb:](#)

> Wie ist den nun die Registeraufteilung für den Z80?

> Eigentlich wollte ich die neu festlegen und dort einiges aufräumen.

Sicher möglich. Als endgültig festgemauert sehe ich da noch nichts. Die 2 Register, die ich für die serielle Schnittstelle reserviert habe, kann man wahrscheinlich freischaufeln. Allerdings muß der RX-Code unbedingt noch überarbeitet und getestet werden.

> Kann man eigentlich noch Code in den DRAM-Routinen sparen?  
> Hier wird doch auch reichlich Zeit verbraten.

Möchtest Du Code oder Zeit einsparen?

Ich weiß nicht, ob Du Dir die 8-Bit Version schon angeschaut hast, aber dort habe ich massig Code gegen Zeit eingetauscht (Macros statt Unterprogramme).

An unkritischen Stellen kann man sicher wieder Unterprogrammaufrufe einbauen, und so etwas Flash einsparen, wenns eng wird.

Ansonsten ist die DRAM-Ecke ausgereizt. Imho. Jedenfalls mit den beiden Hardwarevarianten, die wir jetzt haben.

Ach ja, in den DRAM-Routinen ist zur Zeit über die config-Datei ein (1) "wait state" eingestellt. Bei schnellen RAMS kan man den weglassen. Das sieht dann so aus:

		[.cseg]		avrdude	Laufzeit	"Speed"	
		Code	Data	Used		[s]	[KHz]
	+	+	+	+	+	+	+
-jmp	0x001d00	5240	1434	6674	7424	1.366	2111
" (jp to main)	0x001b00	5000	1434	6434	6912	1.449	1990
" " (0 wait st)	0x001b00	4954	1434	6388	6912	1.415	2038

Wegen der Makros spart das 46 Byte Flash. Geschwindigkeit brings aber gar nicht so viel. Bei der Interpretervariante, die ich gerade teste, hebt es aber den Speed-Faktor wieder über die magische 2 Mhz Marke. ;-)

Autor: Horst S. ([h3aau](#))

Datum: 20.08.2010 17:16

---

Moin moin,

Leo C. schreib:

>Möchtest Du Code oder Zeit einsparen?

Zeit, das ist schliesslich Sinn und Zweck der Übung. Wie bekomme ich eine möglichst schnelle Simulation der Ziel-CPU hin. Ob die RAM-Floppy 1 oder 2 Sekunden braucht ist mir da völlig egal.

Wenn wir dabei bleiben [ATMega8](#) = 8080 und ATMega168 = Z80 hast du noch reichlich Flash zum spielen. :-)

Wobei ich ja immer noch mit einem SRAM liebäugle. (Joe, wie wäre es?)

Wieviel Codes braust du eigentlich für einen normalen DRAM read/write? Dann könnte man mal abschätzen ob das noch Tempo bringt.

Gruss Horst

Autor: Peter Sieg ([petersieg](#))

Datum: 20.08.2010 17:23

---

[Leo C. schrieb:](#)

> -t3-jmp: Statt push/return in der Interpreterschleife wird zwischen den

> Phasen 'load', 'do\_op' und 'store' direkt gesprungen. Kostet Platz, der  
> aber noch vorhanden ist.  
>  
> Meinungen würden mich interessieren.

My 2 cents:

Dank Joe G. sind ja die 4-bit HW Version 1 mit 168 [AVR](#)'s ausgeliefert..  
Wir haben also max. 16kb zur Verfügung.. das sieht bei z.Z ca. 6,5kb  
also  
noch nach 'reichlich' Luft aus.. (auch für den z80 Interpreter)..  
Und ansonsten dient der 328 als 'letzte' Reserve..

Aber genauso wichtig wie speed ist die Übersichtlichkeit und  
Verständlichkeit des Code!

Daraus folgt:

Speed vor kleinerem Code, solange Code übersichtlich und verständlich  
bleibt.

Peter

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 20.08.2010 17:38**

---

Und läuft (88 - 30MHz - 4-bit):

CPM on an AVR, v1.0

Testing RAM: fill...wait...reread...

Initing mmc...

CP/M partition at: **401625**, size: **8032KB**.

CP/M partition at: **417690**, size: **8032KB**.

CP/M partition at: **433755**, size: **8032KB**.

Partinit done.

Ok, CPU is live!

ipl

**62k cp/m vers 2.2**

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>t b

Timer running. Elapsed: **006.212s**.

A>

Down to 6,212s ;-)

BTW: Meldet sich immer noch mit Version 1.0..? Sollte man da nicht mal  
den Zähler etwas höherstellen.. so 1.x und wenn z80 drin ist 2.x..

Was mögliche, künftige HW Redesigns angeht.. wurde schon mehrfach

angesprochen.. Jörg Wolfram hatte Überlegungen dazu.. aber mangels Interesse (von möglichen Usern) diese nicht weiter verfolgt..  
Ich schlage vor, insbesondere, da ja gerade die 8-bit Version erst noch gebaut wird (von der 'Masse') ersteinmal hier an der vorhandenen Basis weiter zu arbeiten.. dann mal sehen, was die Zukunft zu bringt..  
Wichtiger als der Ram ist für mich ggf. I2C Hardware und deren Einbindung ins CP/M..

Peter

Autor: Leo C. ([rapid](#))

Datum: 20.08.2010 21:11

---

Ich habe mal wieder was neues ausprobiert. D.h., eigentlich habe ich nur schon dagewesenes neu zusammengewürfelt.

	End	Code	Data	Used		[s]	[KHz]
-----+-----+-----+-----+-----+-----+-----+-----							
letzte schnellste	0x001d00	5240	1434	6674	7424	1.366	2111
neu	0x001f00	6492	666	7158	7936	1.206	2391

Die Tabellen sind jetzt in Form von r/jmp/call/s von Data nach Code gerutscht. Leider belegen sie so aber nochmal deutlich mehr Platz.

[Peter Sieg schrieb:](#)

> Und ansonsten dient der 328 als 'letzte' Reserve..

Hier wollte ich jetzt hinschreiben, daß das hier das ideale Projekt für den [ATmega8](#) ist, weil der deutlich billiger ist, als seine Nachfolger. Das scheint sich aber gerade zu ändern. Der 8 ist im Moment weder bei CSD noch bei [Reichelt](#) lieferbar. Und CSD hat einen neuen Preis, der deutlich über dem 88 liegt.

> Aber genauso wichtig wie speed ist die Übersichtlichkeit und  
> Verständlichkeit des Code!

An der Übersichtlichkeit hat sich imho bisher nichts geändert, da das Interpreterprinzip ja immer noch das gleiche ist. Und wenns unverständlich ist, liegst vielleicht nur an der Doku. ;-)

> BTW: Meldet sich immer noch mit Version 1.0..? Sollte man da nicht mal  
> den Zähler etwas höherstellen.. so 1.x und wenn z80 drin ist 2.x..

Bei mir gibts keine Versionen, nur Revisionen der einzelnen Dateien. ;) Ich will schon seit längerem eine Funktion einbauen, die beim booten die Revisionen der wichtigsten Teile anzeigt.

Vielleicht möchte ja jemand den "Releasemanager" spielen, und ausgesuchte Revisionen als Version x.x.x taggen.

Autor: Leo C. ([rapid](#))

Datum: 20.08.2010 22:03

---

[Peter Sieg schrieb:](#)

> A>t b  
>  
>  
>

> Timer running. Elapsed: 006.212s.  
> A>  
>  
> Down to 6,212s ;-)

Kann ich ja auch mal:  
A>t b

Timer running. Elapsed: 003.988s.  
A>

Autor: Peter Sieg ([petersieg](#))  
Datum: 21.08.2010 09:34

---

[Leo C. schrieb:](#)

> Timer running. Elapsed: 003.988s.  
> A>

Ja, aber das ist 8-bit Dram ;-)  
Warte mal bis ich auch meine 8-bit Version aufgebaut habe und den  
direkt mitbestellten 40MHz Quarzosi draufgesteckt habe ;-)  
;-)

Peter

# Ladder Spiel

Autor: Leo C. ([rapid](#))

Datum: 23.08.2010 20:40

Angehängte Dateien:

- [LADDER.ZIP](#) (36,5 KB, 50 Downloads)

---

Hallo,

das avrcpm ist inzwischen ja wirklich schnell genug, um Ladder zu spielen. Vor ca. 25 Jahren war ich danach regelrecht süchtig. :-)  
Die Configuration über LADCONF.COM funktioniert bei meinem Terminal (minicom, VT-100/ANSI) nicht richtig. Deshalb habe ichs manuell angepaßt, und in der Initialisierung auch den störenden Cursor abgeschaltet. Mit dem "Programm" CURON.COM kann man den Cursor nach dem Spielen wieder einschalten. Ich habe mal alles zusammen gepackt und hier angehängt.

Ansonsten habe ich aus der 4-bit DRAM-Leseroutine nochmal 4 Taktzyklen rausgequetscht. Aus der Schreibroutine leider nur einen. Das geht, weil durch die Zusammenlegung mit der 8-Bit-Variante jetzt auch hier das \_255-Register vorhanden ist. Leider habe ich nicht ausprobiert, wie viel das gebracht hat.

Dafür habe ich aber mal wieder eine Tabelle von der 8-bit Version:

	--PUSH/POP--		--CALL/RET--		-----DEC-----		Flash
	[s]	[MHz]	[s]	[MHz]	[s]	[MHz]	[byte]
z80int	1.924	1.499	1.971	1.792	6.291	1.467	2426
8080int	1.732	1.665	1.831	1.861	5.911	1.561	2734
-t3	1.712	1.684	1.811	1.882	5.842	1.580	2564
-t3-jmp	1.367	2.109	1.466	2.325	4.707	1.961	2762
-jmp	1.206	2.391	1.281	2.660	3.988	2.314	3272

Unter DEC ist die Schleife von Peter. Ich habe dafür auch mal die Taktzyklen gezählt, damit man die auch mal in ein Z80-Äquivalent umrechnen kann. Unter MHz steht also jeweils die Frequenz, mit der ein echter Z80 laufen müßte, um das gleiche Ergebnis wie mein 20MHz [AVR](#) zu erzielen.

Unter Flash steht nur der Speicherverbrauch des Interpreters. Das restliche Programm ist bei allen Zeilen gleich.

Autor: Peter Sieg ([petersieg](#))

Datum: 24.08.2010 08:47

---

[Leo C. schrieb:](#)

> Unter DEC ist die Schleife von Peter. Ich habe dafür auch mal die  
> Taktzyklen gezählt, damit man die auch mal in ein Z80-Äquivalent  
> umrechnen kann.

Jup. Wie viele Zyklen hast du denn gezählt?

Na, Ladder werde ich doch dann mal probieren.. ;- ) Danke!

Gruß Peter

Autor: Leo C. ([rapid](#))

Datum: 24.08.2010 10:00

---

[Peter Sieg schrieb:](#)

> Jup. Wie viele Zyklen hast du denn gezählt?

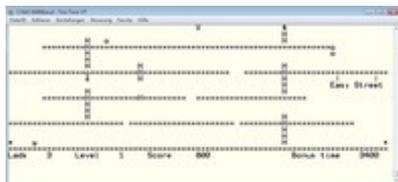
```
023E 3E 01      MVI  A,01
0240 D3 40      OUT  40
0242 0E 0A      MVI  C,0A    ;7                               7\
0244 06 00      MVI  B,00    ;7                               7\
0246 3E 00      MVI  A,00    ;7                               7\
0248 3D          DCR  A      ;4 \
0249 C2 0248    JNZ  0248  ;10 / 14*256 |3605*256 |922901*10 |
024C 05          DCR  B      ;4      4 |
024D C2 0246    JNZ  0246  ;10      10/
0250 0D          DCR  C      ;4                               4 |
0251 C2 0244    JNZ  0244  ;10                               10/
0254 3E 0F      MVI  A,0F    ;7                               7 |
0256 D3 40      OUT  40    ;11                               11/
0258 C9          RET                    ;                               11/
                                           Gesamtzahl Takte: 9229035
```

Ich hoffe, daß ich mich nicht verzählt habe.

Autor: Peter Sieg ([petersieg](#))

Datum: 24.08.2010 15:22

Angehängte Dateien:



[ladder.jpg](#)

58,5 KB, 306 Downloads

---

Soo.. mußte mir ersteinmal ein vernünftiges VT100 Terminalprogramm für Vista laden (TeraTerm) ;- )

Prima! Macht Spaß mit Ladder..

Peter

Autor: Horst S. ([h3aau](#))

Datum: 28.08.2010 11:56

---

Moin moin,

mal ein kleiner Zwischenbericht von der Fleißarbeit.

Alle Opcodes des Z80 aufgelöst, es sind etwas über 700.  
Splitten der Tabellen fertig.

Paging der IX,IY Opcodes fertig.  
CB Opcodes gehen alle.  
ED Opcodes am debuggen.

Der Z80 hat parallel zu den 8080 Opcodes noch einige weitere.  
Diese werden mittels Prefix , also voran gestelltem Opcode, eingeleitet.  
Der Prefix CB leitet z.B. BIT Opcodes ein.  
Der Prefix ED leitet z.B. IN/OUT und Block-Opcodes ein.  
Diese können wie die 8080 Opcodes über Tabellen abgearbeitet werden.

Der Prefix DD leitet alle IX Opcodes ein.  
Der Prefix FD leitet alle IY Opcodes ein.  
In diesen Fällen werden die Opcodes die das HL Register verwenden so umgebogen das nun das jeweilige I-Register verwendet wird.  
In diesen Fällen ist es einfacher die Routinen für FETCH und STORE zu ändern, den nur sie sind davon betroffen.  
Alle zusammen passen in eine 256 Byte Page. Das Highbyte dieser Page wird in main schon getrennt behandelt. Ich muss die geänderten Routinen also nur auf getrennten Pages ablegen und kann dann bei den 8080 Opcodes die gewünschte FETCH/STORE Page verwenden.

Ein Nachteil dieses Verfahrens ist das in diesen vier Fällen main doppelt durchlaufen wird.  
Der Vorteil ist das die Standart-Opcodes ohne zusätzlichen Zeitverlust abgearbeitet werden.

Gruß Horst

Autor: Peter Zabel ([flexopete](#))  
Datum: 31.08.2010 13:12

---

Moin,

habe mal angefangen nach der Artikelbeschreibung ein diskimage zu erstellen. Benutzt habe ich cpmttools für Windows. Die avrcpm habe ich mit Notepad erstellt.  
mkfs.cpm kommt mit einer Fehlermeldung:  
unknown format avrcpm.

Was mache ich falsch?

Gruß  
Peter

Autor: Peter Sieg ([petersieg](#))  
Datum: 31.08.2010 15:41

---

Peter Zabel schrieb:

```
> Die diskdefs liegt im gleichen Verzeichnis wie mkfs.cpm.exe  
> und ist um den Eintrag avrcpm erweitert. Trotzdem gibt  
> es die Fehlermeldung "unknown format avrcpm". Was kann es noch  
> sein?  
>  
> Peter
```

Jup.. da habe ich auch etwas gesucht ;-)) Die Antwort ist so simpel wie das Program doof ist.. Notepad fügt ein CR ans Ende der Zeile.. das können aber die cpmttools unter Windows gar nicht ab ;-))

Also:

1. Die cpmttools MÜSSEN unter X:\cpmttools liegen
2. Die diskdefs muß mit Unix Konvention als Zeilenende (nur LF) sein

Dann geht es.. garantiert..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 01.09.2010 15:11

---

Ich kann nur wiederholen; Es wird mal Zeit für eine ordentliche Dokumentation. Macht Mühe, aber erleichtert ungemein den Nachbau und verhindert sinnloses doppelt, dreifach,..., probieren. Mein Vorschlag: Einer nimmt sich dem Thema an, "Wie bringe ich ein CP/M System auf die SD-Card - eine Anleitung Step by Step". Da unter Windows alles recht bes\*\*eiden geht, gleich eine Anleitung für Linux (Live CD).

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 02.09.2010 10:10

Angehängte Dateien:

- [bdsc-games.zip](#) (39 KB, 43 Downloads)

---

Ich habe den Artikel mal mit entsprechenden Informationen angereichert.. In die grundsätzliche Bedienung eines Partitionierungstools muß man sich aber selber und unabhängig einarbeiten!!

Ich hänge hier auch mal ein BDS-C Sourcearchive an, in dem Spiele und unter anderem auch Pacman enthalten ist.. ich habe z.Z wenig Zeit.. evtl. hat ja mal jemand Zeit und Lust, das auch nach VT100 und unser System umzusetzen.. sollten nur 3-4 Routinen sein (ClrSCR, PosXY,..)..

Peter

Autor: Leo C. ([rapid](#))

Datum: 02.09.2010 20:19

Angehängte Dateien:

- [pacman-avrmpm.zip](#) (27,2 KB, 44 Downloads)

---

[Peter Sieg schrieb:](#)

> sollten nur 3-4 Routinen sein (ClrSCR, PosXY,..)..

Ich habe mal reingeschaut, weil ich auch dachte, daß das ganz schnell gemacht wäre. Leider gings nicht ganz so schnell, weil das Programm direkt auf den UART-Port zugreift. Das wurde bisher von unserem System noch nicht unterstützt. Wahrscheinlich hätte man den Pacman auch an die BIOS Console-I/O anpassen können, aber ich habe es jetzt mal anders herum gemacht.

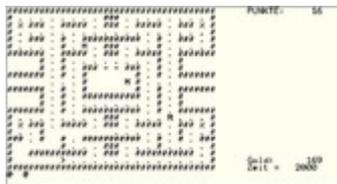
Die Cursor-Steuerung habe ich mal auf die Tasten 2,4,6,8 (Ziffernblock) gelegt.

Damit der Pacman aus dem Anhang läuft, muß man den ATmega updaten:  
<http://cloudbase.homelinux.net/viewvc/avr-cpm/bran...>

Autor: Peter Sieg ([petersieg](#))

Datum: 05.09.2010 17:55

Angehängte Dateien:



• [pacman.jpg](#)

77,9 KB, 276 Downloads

---

Jup. Pacman in aktion ;-)

Peter

## Anfänge der FAT16 Unterstützung

Autor: Frank Zoll ([fzoll](#))

Datum: 17.09.2010 10:07

---

Halloe.

Ich schlage mich leider immer noch mit der Partinionierung meiner SD-Card und dem aufbringen der Images rum. Die Idee, das ganze auf eigene Partinionen zu packen klingt zwar recht nett, macht aber einen riesenumstand für alle Windows-User.

Ich habe letzte nach extra einen "alten" PC platt gemacht um Linux drauf zu spielen. Nun kämpfe ich mit der Hardwareerkennung von Linux. Seitdem ich die SD-Card unter Linux neu partinioniert habe, wird sie beim einstecken nicht mehr gefunden. Ich komme praktisch nicht mehr drauf. Defekt ist die Karte wohl noch nicht, da Windows die eine FAT16 Partition die ich gelassen habe noch erkennt und nutzen kann.

Im moment überlege ich gerade, ob man das ganze nicht evtl. auf Dateien in einem "sticknormalen" FAT16 umstellen könnte. Ich denke darüber nach, die mit dem CPM-Tools angelegten images einfach mit einem bestimmten Namen in das Root-Verzeichniss einer FAT16-Partition zu packen. Sowas wie "diska.img", "diskb.img" .. usw. Und statt die Sektoren in den Partitionen direkt anzusprechen, könnte man dann auf den Dateien "rumrödeln".

Sollte ich die nächten Wochen mal Zeit dazu haben, werde ich mal versuchen die bekannten FAT16-Treiber mit in den Source des CP/M Projekts einzubinden. Leider ist meine Freizeit jedoch sehr beschränkt, so das das bei mir ewig dauern könnte.

Freundliche Grüße

Frank

Autor: Frank Zoll ([fzoll](#))  
Datum: 19.09.2010 14:46

---

Juhu, freu es tut zum ersten mal :-)

Als letzten Fehler war eine der Adressleitungen nicht richtig angeschlossen. Nun läuft's mit der neuen Platine auch einwandfrei. Als Prozessor habe ich den Atmega168 genommen. Sourcen stammen aus dem [SVN](#). Habe es nun erfolgreich mit dem [AVR Studio](#) Compiliert und installiert bekommen. Habe mir einfach ein kleines neues Projekt angelegt, die Sourcen reingeworfen und die Config auf 8bit und 168er Atmega angepasst. Als Betriebssystem auf der SD-Card hab ich die oben als letztes gelinkte Version genommen.  
CPM on an AVR, v1.0  
Testing RAM: fill...wait...reread...  
Initing mmc...  
CP/M partition at: 1757875, size: 9765KB.  
CP/M partition at: 1777406, size: 9765KB.  
CP/M partition at: 1796937, size: 73464KB.  
Partinit done.  
Ok, CPU is live!

ipl  
62k cp/m vers 2.2

```
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>
```

Vielen Dank für eure Hilfe.  
Euer Frank

Autor: Frank Zoll ([fzoll](#))  
Datum: 20.09.2010 11:06

---

[Leo C. schrieb:](#)

> Wir freuen uns jetzt auf den FAT16-Treiber. :-)

Hihi,

ich auch. Ich habe mir gestern Nacht schonmal ein wenig den Kopf darüber zerbrochen. Soweit ich das einschätze wird es möglich sein. Ich habe eine neue FAT16.asm Datei angelegt und schonmal angefangen die ersten FAT16 Implementierungen vorzubereiten.

Zum aktuellen Stand:

- Eine FAT16 Partition wird bereits erkannt.
- Der BootSektor der FAT16 Partition wird bereits erfolgreich gelesen.

Nächste Schritte:

- Finden und Auswerten des Root-Verzeichnisses.
- Zuordnung gefundener Images auf die bestehenden Laufwerke A: bis D:.
- Finderroutine für das Auffinden beliebiger Sektoren über die FAT-Einträge in den Image-Dateien
- Read/Write des gefundenen Sektors über die bestehenden MMC-Routinen anbinden.

Ich werde mit einem sehr stark eingeschränkten FAT16 Treiber die Sache wohl ans laufen bringen.

Folgende Eckdaten denke ich lassen sich erreichen:

- Es wird eine FAT16 Partition vorausgesetzt.
- In der FAT16 Partition wird es ein Paar Dateien geben, die die einzelnen CP/M Partitionen darstellen werden.
- Die Image- Dateien müssen im ROOT- Verzeichniss der FAT16 Partition angelegt werden.
- Die Image- Dateien werden sich nicht auf dem [AVR](#)-CP/M anlegen lassen. So spare ich mir die Implementierung von "create\_file"- Funktionen. (Da sehr Umständlich)
- Die Namen für die Image- Dateien werden wir im Code fest hinterlegen.
- Die Größe der Image-Dateien kann nicht dynamisch geändert werden. Die Images müssen praktisch von anfang an die volle Größe einer CP/M - Partition abbilden.
- Die Performance wird gegenüber den direkten CP/M Partionen stark leiden. Bei jedem Sektorzugriff muss dieser erst umständlich in der jeweiligen Image-Datei gesucht werden. ( Juhu Retro like nix schnelle :- ) )
- Schreiben und Lesen einzelner Sektoren der Image-Dateien wird funktionieren.
- Ich verwende den Pufferspeicherblock, der schon für das Lesen/Schreiben der CP/M Partitionen verwendet wird einfach wieder. Ich brauche ein paar zusätzliche Bytes, für die wichtigsten Variablen ausm FAT16 System. Ich werde mit den derzeit noch verbliebenen knapp 100 Bytes SRAM eines 168AVR's wohl hinkommen. ( Ein paar Bytes werde ich noch überlassen :- ) )
- Ich mache die FAT16 Implementierung speziell für dieses Projekt neu, um so viel Speicher und Ressourcen wie nur irgend geht einzusparen.
- Ich versuche das ganze "unauffällig" in den bestehenden Code einzufügen. Am Ende wäre das Ziel das man gleichzeitig mit FAT16 und CP/M Partitionen arbeiten kann.

Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**

**Datum: 20.09.2010 23:06**

---

[Leo C. schrieb:](#)

> [Frank Zoll schrieb:](#)

>> - Die Performance wird gegenüber den direkten CP/M Partionen stark  
>> leiden. Bei jedem Sektorzugriff muss dieser erst umständlich in der  
>> jeweiligen Image-Datei gesucht werden. ( Juhu Retro like nix schnelle  
>> :- ) )

>

> Ich glaube, so schlimm wirts garnicht. Die SD-Zugriffe sind im Vergleich  
> zum Interpreter und zu den RAM-Zugriffen nämlich sauschnell. Die  
> RAM-Disk ist auch eher langsamer als die SD-Disk(s).

Das Blöde ma FAT Dateisystem ist, das ich mich bei jedem Sektorzugriff erst die verlinkte Liste der Cluster in der FAT- Tabelle entlanghangeln muss, um raus zu bekommen in welchem Sektor der gesuchte Dateisektor

wirklich steht. Je weiter hinten in der Datei dieser Sektor steht, desto langsamer wird die Routine arbeiten. Ich nutzte praktisch kaum Puffer, so das ich da jedes mal vom anfang der Datei an alles lesen muss.

>> FAT16 System. Ich werde mit den derzeit noch verbliebenen knapp 100 Bytes SRAM eines 168AVR's wohl hinkommen. ( Ein paar Bytes werde ich noch überlassen :- )

>

> Und wenns doch nicht reicht, kann man die UART RX/TX-Buffer noch verkleinern. Gut wäre natürlich, wenn man einen 2. Sektorpuffer hätte. Das würde mit dem ATmega328 gehen.

Ich müsste schon einen ziemlich großen Block Speicher reservieren, um mir eine Tabelle aller Cluster aufzubauen aus denen die jeweilige Image-Datei besteht. Da komme ich mit ein paar mehr Bytes einfach nicht aus. Evtl. könnte man ja später mal drüber nachdenken diese Liste in einem Teil des DRAM's zu lagern. Im moment wäre ich aber auch schon froh, wenns überhaupt läuft.

Einen 2. Sektorpuffer brauche ich erstmal nicht denke ich.

Ach ja, zu eurem Leidwesen habe ich hier mal eben riesen Parts des Quellcodes umgeschaufelt und neu zusammen gestrickt. Ich habe die Datei "remainders.asm" ziemlich leer gemacht. Da ist kaum was über geblieben.

Bei mir siehts nun so aus:

virt\_ports.asm Hier sind die virtuellen Ports und die Hilfsroutinen für den UART- Zugriff über Ports.

dsk\_fsys.asm Hier liegen die Grundroutinen des virtuellen Dateisystems

dsk\_cpm.asm Hier liegen die Routinen für den Zugriff auf CP/M

Partitionen

dsk\_fat.asm Hier liegen die Routinen für den Zugriff auf FAT16

Partitionen

dsk\_ram.asm Hier liegen die Routinen für den Zugriff auf Ramdisk

Partitionen

dsk\_mgr.asm Hier liegen die Routinen für die

Partitionstabellenverwaltung

Um das ganze nett aussehen zu lassen habe ich die Tabelle hostparttbl um ein Byte erweitert. Jeder Eintrag dort wird nun angeführt von einem Byte, das den Typ der jeweiligen Partition widerspiegelt.

(0=None,1=CP/M,2=Fat,3=RAM). Im moment baue ich gerade alle Routinen so um, das sie dieses Byte erkennen können und dann jeweils die für die Partition richtigen lese/schreib routinen aufrufen.

Im moment läuft nix mehr, aber ich hoffe ich bekomme das noch wieder ans laufen :-)

Grüße

Frank

Autor: Frank Zoll ([fzoll](#))

Datum: 21.09.2010 14:38

---

[Leo C. schrieb:](#)

> [Frank Zoll schrieb:](#)

>> Es gibt

>> einige Cards bei denen ist das Timing besonders kritisch. Die Karte die

>> bei mir nicht wollte war eine 1GB-Platinum Card von [Reichelt](#).

>

> Würde es Dir sehr viel Mühe bereiten, die Platinum-Karte nochmal mit

> Softwarerevision 91 zu testen?

Ich habe mir heute morgen mal schnell die neuere Version gezogen und es ausprobiert. Habs aber auf Anhieb nicht gleich ans laufen bekommen und musste recht schnell aufgeben. Er hat einmal ganz kurz die Platinum Disk erkannt, sich dann aber irgendwie aufgehängt. Nach dem Aufhängen kam keine Verbindung mehr zum Terminal zustande. Erst nachdem ich wieder meine eigene Entwicklung aufgespielt habe, hat sich der Rechner wieder gemeldet.

Wie ich sehe scheint es nicht unbedingt nur an der Marke "Platinum" zu liegen. Zwei 1GB- Karten scheinen bei Hans Werner ja zu laufen. Ich denke einfach mal, das meine SD-Card bereits eine Macke hat. Wer ganz oben mitgelesen hat, wird ja schon sehen das die Karte auch unter SUSE-11.2 nicht ans laufen zu bringen war. Immer wenn ein HighSpeed zugriff statt fand, brach die Verbindung zur Karte ab. Ich hab die Karte mittlerweile mal versucht mit meinen HIVE ans laufen zu bekommen. Auf dem HIVE zieht sich da ein ähnlicher Fehler. Die Karte wird nur ab und zu mal erkannt.

Ich denke einfach mal das die Karte bereits einen kleinen Schaden weg hat. Aber seltsam ist's schon das sie unter Window noch ansprechbar ist.

Freundliche Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 21.09.2010 23:03**

---

Halloe.

Mal ein kurzer Status der FAT16 Implementierung. Ich kämpfe mich gerade durch die Strukturen von Bootblock, FAT und Directory. Habe schon die korrekten Positionen von Bootblock, FAT, und Root- Directory auf meiner Testpartition gefunden. Morgen schaue ich dann mal, das ich das Root- Verzeichniss nach den Image-Dateien durchsuche.

```
Initing mmc...
fat16 part found
fat16 scanning
free entrys in ptable ?
read fat bootblock.
fat16 bootblock check
Bytes per Sector 0200
Sectors per Cluster 20
Reserved Sectors__: 0001
Sectors per FAT__: 00D7
Ammount of FAT copies: 02
Max. entrys in Rootdir.: 0200
Begin of Volume at: 0000003F
Begin of FAT at___: 00000040
Begin of DIR at___: 000001EE

CP/M partition at: 1757875, size: 9728KB.
CP/M partition at: 1777406, size: 9728KB.
CP/M partition at: 1796937, size: 73344KB.
Partinit done.
Ok, CPU is live!
```

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

```
A>
```

Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 22.09.2010 23:29**

---

Halloe.

```
Wieder ein kleines Stück voran gekommen.  
CP/M partition at: 1757875, size: 9728KB.  
CP/M partition at: 1777406, size: 9728KB.  
CP/M partition at: 1796937, size: 73344KB.  
FAT16 File-Image at: 005, size: 256258Bytes.  
Partinit done.  
Ok, CPU is live!
```

Nun läßt sich der Dateianfang bereits erfolgreich lokalisieren. Als nächstes muss ich noch eine Methode bauen, die sich Cluster für Cluster zu einem beliebigen "Zielsektor" vorarbeiten kann. Dann kann ich das Laden und Speichern der Sektoren so anpassen, das Sektoren aus dem Image gelesen und geschrieben werden können. Vorher muss ich nur noch klären, wieso da oben 256258 Bytes steht und nicht 256256. Denn die Datei ist genau 256256 Bytes groß, irgendwie haben sie da 2 Zusatzbytes eingeschlichen :-). Die Zahl 005 gibt die Nummer des ersten Clusters der Datei an. Diese Zahl speichere ich nun zusammen mit dem Partitionstyp in der internen Partitionstabelle. Damit lassen sich später sowohl der Dateianfang als auch der erste Cluster in der FAT schnell wiederfinden.

Grüße  
Frank

**Autor: Peter Sieg ([petersieg](#))**  
**Datum: 23.09.2010 10:05**

---

@all: Mann! Schön, das sich hier so viel tut!! Wer hätte das mal gedacht.. das sich aus dem 'Proof of concept' mal ein (fast) vollwärtiges CP/M System entwickelt...

Ich bin zur Zeit mit CC 2010 (Verein zum Erhalt klassischer Computer) Messe/Treffen-Vorbereitungen etwas arg ausgelastet und komme daher zur Zeit zu fast nicht anderem mehr.. aber der nächste 'Winter' kommt bestimmt.. ;-)

Weiter so!

Peter

**Autor: Joe G. ([feinmechaniker](#))**   
**Datum: 24.09.2010 13:21**

---

@alle: Ich bin beeindruckt was hier so alles passiert, wenn man eine Woche nicht im Lande ist... Wichtig für noch alle weiteren potentiellen Nachbauer und natürlich auch für die aktuelle Dokumentation, wäre die Information ob Layout und Schaltung fehlerfrei sind (PWREN# habe ich schon). Da einige „neue“ Systeme schon arbeiten, gehe ich mal davon aus oder irre ich?  
Gruß Joe

Autor: Leo C. ([rapid](#))

Datum: 29.09.2010 01:58

---

Vor kurzem bin ich zufällig über ein CP/M-Plugin für den "Total Commander" gestolpert:

[http://hc-ddr.hucki.net/wiki/doku.php/cpm:disketten\\_xp2](http://hc-ddr.hucki.net/wiki/doku.php/cpm:disketten_xp2)

Vielleicht ist das ja für den einen oder anderen Windowsbenutzer interessant.

Auf der gleichen Seite habe ich auch was zu zxcc von John Elliott gesehen:

<http://hc-ddr.hucki.net/wiki/doku.php/cpm:windows>

Das benutze ich schon eine Weile, um unter Linux das BIOS mit M80 zu assemblieren. Dieses, weil ich die Macrofähigkeiten des M80 für die Disk-Parameter-Tabellen nutze.

Zitat von der zxcc Homepage:

<http://www.seasip.demon.co.uk/Unix/Zxcc/index.html>

ZXCC is a two-purpose CP/M 2/3 emulator allowing:

- \* Hi-Tech C for CP/M to be used as a cross-compiler under Unix.
- \* The CP/M build tools (MAC, RMAC, GENCOM, LINK) to be used under DOS or Unix.

Vielleicht mag das ja mal jemand unter Windows testen.

Autor: Peter Sieg (Gast)

Datum: 29.09.2010 08:29

---

Leo C. schrieb:

```
>> habe mit "DD if=diskimage of=\\.\f:" die Datei auf SD kopiert, benötige  
>> ich noch Parameter die eine Partion vom Type 52 anlegt ??  
>
```

```
> Ich kenne mich mit dem Windows-DD nicht genug aus, um sagen zu können,  
> ob "\\.\f:" in einer Partition liegt, oder am Anfang der Karte.
```

Damit liegt das Image am Anfang der Karte.. daher ist es keine Partition und damit ist Part.Type 52 auch irrelevant..

Ist denn der Artikel an dieser Stelle noch zu unverständlich..?:

[quote]

Das so erzeugete diskimage wird ROH (ohne Filesystem) mit:

```
dd if=diskimage of=<sd card identifier> bs=512
```

auf eine SD Karte geschrieben. Achtung! Alle anderen Dateien gehen dabei verloren!

Die aktuelle Version unterstützt ebenfalls bis zu 4 primäre CP/M = Typ 52 Partitionen. Diese lassen sich mit einer Linux Live-CD und dem Tool 'cfdisk' anlegen. Da z.Z die virtuellen Disketten nur ca. 250kb 'groß' sind, reicht es entsprechend große Partitionen anzulegen (Ich habe trotzdem 8MB große angelegt.. für später). Wichtig dabei ist es den Typ auf 52 = CP/M zu setzen.. und schreiben/speichern nicht vergessen. Die einzelnen Partitionen werden dann ohne zu formatieren wieder mittels 'dd if=diskimage of=[sd card partition] bs=512' gefüllt. Wird also die SD Karte unter Linux z.B als sdh angezeigt, so ist die erste Partition dann sdh1, die zweite sdh2 etc. of= ist dann also /dev/sdh1 /2 /3..

Ich habe übrigens die erste Partition als FAT eingerichtet und entsprechend formatiert. So kann man den großen Bereich weiter unter Windows nutzen. Und dann nur 3 CP/M Partitionen am Ende des Speicherbereiches angelegt..  
 [/quote]

Peter

Autor: Peter Sieg (Gast)  
 Datum: 29.09.2010 09:56  
 Angehängte Dateien:



- [4Mx4\\_SOJ.jpg](#)  
34,7 KB, 150 Downloads

```

64256, 44288
256Kx4 DRAM.

-----
D0 13 4444 201 SMD
D1 12 191 D0
/IDE 13 181 S0
/RA0 14 171 /CAS
15 44256 141 /IDE
A0 14 44288 131 A0
A1 17 141 A7
A2 18 131 A6
A3 19 121 A5
VCC 120 111 A4
-----
  
```

- [dram 4256 pinout.jpg](#)  
45,3 KB, 108 Downloads



The device signal pin assignments are listed in a table below. The table includes columns for Pin No., Pin Name, and Pin Description.

Pin No.	Pin Name	Pin Description
1	DRAM0	DRAM Data Bus (D0)
2	DRAM1	DRAM Data Bus (D1)
3	DRAM2	DRAM Data Bus (D2)
4	DRAM3	DRAM Data Bus (D3)
5	DRAM4	DRAM Data Bus (D4)
6	DRAM5	DRAM Data Bus (D5)
7	DRAM6	DRAM Data Bus (D6)
8	DRAM7	DRAM Data Bus (D7)
9	DRAM8	DRAM Data Bus (D8)
10	DRAM9	DRAM Data Bus (D9)
11	DRAM10	DRAM Data Bus (D10)
12	DRAM11	DRAM Data Bus (D11)
13	DRAM12	DRAM Data Bus (D12)
14	DRAM13	DRAM Data Bus (D13)
15	DRAM14	DRAM Data Bus (D14)
16	DRAM15	DRAM Data Bus (D15)
17	DRAM16	DRAM Data Bus (D16)
18	DRAM17	DRAM Data Bus (D17)
19	DRAM18	DRAM Data Bus (D18)
20	DRAM19	DRAM Data Bus (D19)
21	DRAM20	DRAM Data Bus (D20)
22	DRAM21	DRAM Data Bus (D21)
23	DRAM22	DRAM Data Bus (D22)
24	DRAM23	DRAM Data Bus (D23)
25	DRAM24	DRAM Data Bus (D24)
26	DRAM25	DRAM Data Bus (D25)
27	DRAM26	DRAM Data Bus (D26)
28	DRAM27	DRAM Data Bus (D27)
29	DRAM28	DRAM Data Bus (D28)
30	DRAM29	DRAM Data Bus (D29)
31	DRAM30	DRAM Data Bus (D30)
32	DRAM31	DRAM Data Bus (D31)
33	DRAM32	DRAM Data Bus (D32)
34	DRAM33	DRAM Data Bus (D33)
35	DRAM34	DRAM Data Bus (D34)
36	DRAM35	DRAM Data Bus (D35)
37	DRAM36	DRAM Data Bus (D36)
38	DRAM37	DRAM Data Bus (D37)
39	DRAM38	DRAM Data Bus (D38)
40	DRAM39	DRAM Data Bus (D39)
41	DRAM40	DRAM Data Bus (D40)
42	DRAM41	DRAM Data Bus (D41)
43	DRAM42	DRAM Data Bus (D42)
44	DRAM43	DRAM Data Bus (D43)
45	DRAM44	DRAM Data Bus (D44)
46	DRAM45	DRAM Data Bus (D45)
47	DRAM46	DRAM Data Bus (D46)
48	DRAM47	DRAM Data Bus (D47)
49	DRAM48	DRAM Data Bus (D48)
50	DRAM49	DRAM Data Bus (D49)
51	DRAM50	DRAM Data Bus (D50)
52	DRAM51	DRAM Data Bus (D51)
53	DRAM52	DRAM Data Bus (D52)
54	DRAM53	DRAM Data Bus (D53)
55	DRAM54	DRAM Data Bus (D54)
56	DRAM55	DRAM Data Bus (D55)
57	DRAM56	DRAM Data Bus (D56)
58	DRAM57	DRAM Data Bus (D57)
59	DRAM58	DRAM Data Bus (D58)
60	DRAM59	DRAM Data Bus (D59)
61	DRAM60	DRAM Data Bus (D60)
62	DRAM61	DRAM Data Bus (D61)
63	DRAM62	DRAM Data Bus (D62)
64	DRAM63	DRAM Data Bus (D63)

- [sd card pinout.jpg](#)  
34,6 KB, 115 Downloads

Pinout diagram for the ATmega8 microcontroller.

Pin	Function
1	RESET
2	PC0
3	PC1
4	PC2
5	PC3
6	PC4
7	PC5
8	PC6
9	PC7
10	PC8
11	PC9
12	PC10
13	PC11
14	PC12
15	PC13
16	PC14
17	PC15
18	PC16
19	PC17
20	PC18
21	PC19
22	PC20
23	PC21
24	PC22
25	PC23
26	PC24
27	PC25
28	PC26
29	PC27
30	PC28
31	PC29
32	PC30
33	PC31
34	PC32
35	PC33
36	PC34
37	PC35
38	PC36
39	PC37
40	PC38
41	PC39
42	PC40
43	PC41
44	PC42
45	PC43
46	PC44
47	PC45
48	PC46
49	PC47
50	PC48
51	PC49
52	PC50
53	PC51
54	PC52
55	PC53
56	PC54
57	PC55
58	PC56
59	PC57
60	PC58
61	PC59
62	PC60
63	PC61
64	PC62
65	PC63
66	PC64
67	PC65
68	PC66
69	PC67
70	PC68
71	PC69
72	PC70
73	PC71
74	PC72
75	PC73
76	PC74
77	PC75
78	PC76
79	PC77
80	PC78
81	PC79
82	PC80
83	PC81
84	PC82
85	PC83
86	PC84
87	PC85
88	PC86
89	PC87
90	PC88
91	PC89
92	PC90
93	PC91
94	PC92
95	PC93
96	PC94
97	PC95
98	PC96
99	PC97
100	PC98
101	PC99
102	PC100
103	PC101
104	PC102
105	PC103
106	PC104
107	PC105
108	PC106
109	PC107
110	PC108
111	PC109
112	PC110
113	PC111
114	PC112
115	PC113
116	PC114
117	PC115
118	PC116
119	PC117
120	PC118
121	PC119
122	PC120
123	PC121
124	PC122
125	PC123
126	PC124
127	PC125
128	PC126
129	PC127
130	PC128
131	PC129
132	PC130
133	PC131
134	PC132
135	PC133
136	PC134
137	PC135
138	PC136
139	PC137
140	PC138
141	PC139
142	PC140
143	PC141
144	PC142
145	PC143
146	PC144
147	PC145
148	PC146
149	PC147
150	PC148
151	PC149
152	PC150
153	PC151
154	PC152
155	PC153
156	PC154
157	PC155
158	PC156
159	PC157
160	PC158
161	PC159
162	PC160
163	PC161
164	PC162
165	PC163
166	PC164
167	PC165
168	PC166
169	PC167
170	PC168
171	PC169
172	PC170
173	PC171
174	PC172
175	PC173
176	PC174
177	PC175
178	PC176
179	PC177
180	PC178
181	PC179
182	PC180
183	PC181
184	PC182
185	PC183
186	PC184
187	PC185
188	PC186
189	PC187
190	PC188
191	PC189
192	PC190
193	PC191
194	PC192
195	PC193
196	PC194
197	PC195
198	PC196
199	PC197
200	PC198
201	PC199
202	PC200
203	PC201
204	PC202
205	PC203
206	PC204
207	PC205
208	PC206
209	PC207
210	PC208
211	PC209
212	PC210
213	PC211
214	PC212
215	PC213
216	PC214
217	PC215
218	PC216
219	PC217
220	PC218
221	PC219
222	PC220
223	PC221
224	PC222
225	PC223
226	PC224
227	PC225
228	PC226
229	PC227
230	PC228
231	PC229
232	PC230
233	PC231
234	PC232
235	PC233
236	PC234
237	PC235
238	PC236
239	PC237
240	PC238
241	PC239
242	PC240
243	PC241
244	PC242
245	PC243
246	PC244
247	PC245
248	PC246
249	PC247
250	PC248
251	PC249
252	PC250
253	PC251
254	PC252
255	PC253

- [atmega8-pinout.png](#)  
17,1 KB, 139 Downloads

Autor: Frank Zoll ([fzoll](#))  
 Datum: 29.09.2010 10:57

---

[Egmont schrieb:](#)  
 > [Frank Zoll schrieb:](#)  
 >> Halloe.  
 >> CP/M partition at: 1777406, size: 9728KB.  
 >> FAT16 File-Image at: 005, size: 256258Bytes.  
 >  
 > @Frank,  
 > was macht deine Software mit FAT16...

Och das ist eigentlich ganz einfach zu erklären. Ich hab mich einfach zu doll über Linux geärgert, weil ich immer wieder Probleme mit der Hardwareerkennung hatte. Daher suchte ich nach einer "einfacheren" Möglichkeit um unter Windows die "Partitionen" für das CP/M Projekt anzulegen.

Meine Aktuelle Lösung sieht so aus:

- Ich suche die 1. FAT16 Partition die ich auf der SD-Card finden kann.
- Ich schaue im ROOT- Verzeichniss dieser Partition nach Dateien die diesem Schema entsprechen "CPMDSK\_!.IMG" ( != Ein beliebiger Buchstabe A-Z )
- Diese Dateien müssen derzeit genau 256KB gross sein. Daher habe ich ein kleines Programm geschrieben das an die hier im Forum zu findenden Images einfach Nullen dranhängt, bis 256KB erreicht sind.
- Jede gefundene Datei wird als virtuelle Partition in das CP/M System eingebunden.
- Für das CP/M sieht der Zugriff über die Ports nacher genauso aus, wie bei den "echten" CP/M Partitionen.
- Ach ja. Da echte CP/M Partitionen einen schnelleren Zugriff erlauben, werden diese bevorzugt an den Anfang der Liste der "virtuellen" Laufwerke gesetzt. Eine Mischung auf CP/M und FAT16 Partitionen ist somit dann auch möglich.

Mit diesen Lösung spart man sich das Partitionieren der SD- Card unter Linux und den ständigen wechsel zwischen Windows und Linux zum kopieren der CP/M Images. Die per CPMTTOOLS angelegten Diskettenimages kann man nach dem "aufpusten" auf 256KB praktisch direkt verwenden.

Ich bin in den letzten Tagen nicht zum Coden gekommen. Aktuell muss ich "nur noch" die Funktion schreiben, die aus einer CP/M- Sektorangabe einen Sektor aus der Diskimage- Datei ermittelt. Schreiben und Lesen der Sektoren werde ich danach wieder über die bereits bestehenden MMC- Zugriffsrotuinen abwickeln. Ich denke das ich da noch 3-4 Abende rein stecken muss, bis es endlich richtig läuft. Sobald ich es am laufen habe werde ich hier mal ein Archiv mit meiner Version posten.

Mit dem zur Verfügung stehenden Speicher werden wir auskommen. Aktuell liegt der Speicherverbrauch bei meinem Atmega168 Version bei:  
Flash 9196 von 16384 genutzt.  
SRAM 930 von 1024 genutzt.  
EEPROM noch leer.

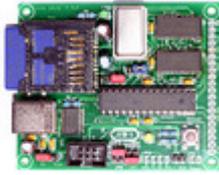
Grüße  
Frank

## Platine Version 2

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.09.2010 18:03

Angehängte Dateien:



[AVR CPM V2.0.png](#)

536,4 KB, 345 Downloads

---

So sieht sie aus, die Version 2.0  
Joe

## Weiter mit FAT16

Autor: Frank Zoll ([fzoll](#))

Datum: 01.10.2010 21:03

---

So,  
hat leider etwas länger gedauert als gedacht aber hier mal ein kurzes Update. Ich hab soeben das erste mal lesend auf ein Diskimage auf meiner FAT16 Partition zugreifen können. Das anzeigen des Directorys und der Programmstart funktionieren. Das schreiben hab ich vorsichtshalber erstmal abgestellt, bis ich sicher bin, das die Schreibopüerationen auch immer schön in die Datei gehen und nicht daneben. :-)

Ich habe Leo eben mal eine kleine Vorabversion zum testen geschickt. Sobald die letzten Macken beseitigt sind und der neue Code auch schön aufgeräumt ist, gibts hier ein kleines Release.

```
CP/M partition at: 1757875, size: 9765KB.  
CP/M partition at: 1777406, size: 9765KB.  
CP/M partition at: 1796937, size: 73464KB.  
FAT16 File-Image at: 005, size: 250KB.  
Partinit done.  
Ok, CPU is live!
```

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

```
A>d:  
D>dir  
D: ASM          COM : DDT          COM : DUMP          COM : ED          COM  
D: T            COM : TLOOP        COM : LOAD          COM : MBASIC       COM  
D: 23          BAS : BAGELS        BAS : CHECKERS     BAS : STARTREK   BAS  
D: TREKINST    BAS : MASTRMND     BAS : WEEKDAY      BAS : PIP         COM  
D: STAT        COM : SUBMIT      COM : XSUB         COM : ZORK1       COM  
D: ZORK1       DAT  
D>mbasic  
39417 Bytes free  
BASIC Rev. 4.51  
[CP/M Version]  
Copyright 1977 (C) by Microsoft  
Ok
```

Grüße  
Frank

Autor: Leo C. ([rapid](#))

Datum: 01.10.2010 22:44

---

[Frank Zoll schrieb:](#)

> So,  
> hat leider etwas länger gedauert als gedacht aber hier mal ein kurzes  
> Update. Ich hab soeben das erste mal lesend auf ein Diskimage auf meiner  
> FAT16 Partition zugreifen können. Das anzeigen des Directorys und der  
> Programmstart funktionieren. Das schreiben hab ich vorsichtshalber  
> erstmal abgestellt, bis ich sicher bin, das die Schreiboperationen auch  
> immer schön in die Datei gehen und nicht daneben. :-)  
>  
> Ich habe Leo eben mal eine kleine Vorabversion zum testen geschickt.  
> Sobald die letzten Macken beseitigt sind und der neue Code auch schön  
> aufgeräumt ist, gibts hier ein kleines Release.

Zum Ausprobieren ist jetzt alles auf:

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

Oder inzwischen auch besser mit [SVN-Client](#) (z.B. Tortoise [SVN](#)) auf:

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

Die Reste aus "remainders.asm" haben jetzt auch ihre eigenen Dateien bekommen. Außerdem habe ich für Jump und Call Makros gebaut, die je nach Prozessor und Sprungweite den passenden rjmp/jmp, bzw. rcall/call Befehl verwenden.

Bei mir assemblieren alle Prozessor-/RAM-Kombinationen mit und ohne FAT16-Unterstützung (Mega8 und 88 nur ohne FAT16).

Das ist alles nur zum Testen und noch nicht unbedingt stabil. Z.B. passt das BIOS aus trunk nicht mehr dazu. Bitte das BIOS aus dem stabilen Zweig nehmen.

Autor: Leo C. ([rapid](#))

Datum: 01.10.2010 23:12

---

Zur Zeit muß die FAT16-Partition den Typ 0E haben. Da wir nur LBA-Adressierung verwenden, wäre das eigentlich auch richtig. Gängig ist aber auch Typ 6, der nicht erkannt wird. Sicher könnte man das umstellen oder Typ 6 zusätzlich einbauen. Es stellt sich nur die Frage, was sinnvoll ist.

"Meine" Partitionstypenreferenz

([http://www.win.tue.nl/~aeb/partitions/partition\\_types-1.html](http://www.win.tue.nl/~aeb/partitions/partition_types-1.html)) meint dazu:

06 DOS 3.31+ 16-bit FAT (over 32M)

Partitions, or at least the FAT16 filesystems created on them, are at most 2 GB for DOS and Windows 95/98 (at most 65536 clusters, each at most 32 KB). Windows NT can create up to 4 GB FAT16 filesystems (using 64 KB clusters), but these cause problems for DOS and Windows 95/98. Note that VFAT is 16-bit FAT with long filenames; FAT32 is a different filesystem.

0e WIN95: DOS 16-bit FAT, LBA-mapped

0f WIN95: Extended partition, LBA-mapped

Windows 95 uses 0e and 0f as the extended-INT13 equivalents of 06 and 05. For the problems this causes, see Possible data loss with LBA and INT13 extensions. (Especially when going back and forth between MSDOS and Windows 95, strange things may happen with a type 0e or 0f partition.)

Windows NT does not recognize the four W95 types 0b, 0c, 0e, 0f ( Win95 Partition Types Not Recognized by Windows NT). DRDOS 7.03 does not support this type (but DRDOS 7.04 does).

Das Linux/Gnome Partitionierprogramm "gparted" ist übrigens besonders schlau ;-( Es hat einen Dialog, in dem man "Markierungen/Flags" einstellen kann. Schaltet man das LBA-Flag ein, wird der Partitionstyp von 6 auf E umgestellt.

Autor: Egmont (Gast)

Datum: 02.10.2010 09:32

Angehängte Dateien:

- [Bootloader\\_CP-M.zip](#) (20,7 KB, 34 Downloads)

---

Hallo zusammen,  
wollte mal was über den Bootloader schreiben den ich benutze:

Bootloader von Peter Danegger  
[http://www.mikrocontroller.net/articles/AVR\\_Bootlo...](http://www.mikrocontroller.net/articles/AVR_Bootlo...)

- Boot\_168.hex in ATMegal68 laden
- Fusebit auf "BOOTRST" und 512 Byte stellen
- FBoot.exe ins Verzeichniss kopieren wo auch die avrcpm.hex ist.
- FBoot ausführen mit folgenden Parameter  
oder Flash.bat

```
fboot /C2 /B9600 /Pavrcpm.hex
      ^   ^       ^
      |   |       |
      |   |       +- Hex-File
      |   +----- Baudrate
      +----- Comport (1-4)
```

fboot funktioniert bei mir nur bis Win XP.

Erlaubter Comport von 1 bis 4

Der Bootloader hat eine automatische Baudratenerkennung und man kann auch höhere Werte eingeben zb 38400

Tipp:

Mit dem [FT232RL-Tool](#) (MProg 3.5 Release) eine feste "USB Serial Number" vergeben, somit bekommt man immer den gleichen ComPort zugewiesen auch wenn sich der USB-Port ändert.

In der Systemsteuerung / Geräte-Manager den ComPort ändern auf z.B. 2

Gruß Egmont

Autor: Egmont (Gast)

Datum: 02.10.2010 13:50

Angehängte Dateien:

- [Bootloader\\_CP-M\\_Source.zip](#) (11 KB, 38 Downloads)

---

[Joe G. schrieb:](#)

> Für welche Taktfrequenz ist der Bootloader übersetzt?

Du fragst bestimmt wegen Baudrate... da ist das so das er sich das selbst errechnet. Es ist also kein Problem wenn man seine Taktfrequenz ändert. Bekommt von FBoot.exe Daten geschickt wo dann die Baudlänge ermittelt wird.

Habe ich schon selber ausprobiert. 12MHz vom [FT232](#) eingestellt CP-M-Source neu übersetzt mit der Taktfrequenz und mit dem Bootloader geladen.

In der FastLoad.H steht zwar bei mir 8Mhz drinn wird aber nur dafür verwendet um ein BootDelay zu errechnen.

```
.equ XTAL      = 8000000 ; 8MHz, not critical
.equ BootDelay = XTAL / 3 ; 0.33s
```

wichtig sind die Einstellungen in Bootload.asm

```
.equ STX_PORT      = PORTB
.equ STX           = PB1
```

```
.equ SRX_PORT      = PORTB
.equ SRX           = PB0
```

Was man noch beachten sollte:

- SD-Karte entfernen
- FBoot.exe starten
- CP/M Reset

Source vom Bootloader siehe Anhang

Ich hoffe ich habe deine Frage richtig verstanden bzw. beantwortet.  
Egmont

@Alle

für Linux gibt es auch Software dafür...

**Autor: Egmont (Gast)**

**Datum: 06.10.2010 07:18**

---

[Frank Zoll schrieb:](#)

> Ich hab soeben das erste mal lesend auf ein Diskimage auf meiner  
> FAT16 Partition zugreifen können. Das anzeigen des Directorys und der  
> Programmstart funktionieren.

Wie muss den die FAT16 Partition aussehen

Type (06) , Größe(960MB), IMG-Dateiname (cpm\_d.img), IMG-Dateigröße.

Bei mir erscheint nur "fat16 scanning" aber keine Laufwerk.

Der Werte in der Klammer sind die Einstellung so wie sie bei mir sind.

Die SD-Karte hat vier Partitionen:

1.	FAT16	06	960MB	
2.	CP/M	52	8MB	aktiv
3.	CP/M	52	8MB	
4.	CP/M	52	8MB	

Gruß Egmont

**Autor: Frank Zoll ([fzoll](#))**

**Datum: 06.10.2010 09:00**

---

[Egmont schrieb:](#)

> Wie muss den die FAT16 Partition aussehen  
> Type (06) , Größe(960MB), IMG-Dateiname (cpm\_d.img), IMG-Dateigröße.  
>  
> Bei mir erscheint nur "fat16 scanning" aber keine Laufwerk.  
> Der Werte in der Klammer sind die Einstellung so wie sie bei mir sind.  
>  
> Die SD-Karte hat vier Partitionen:  
> 1. FAT16 06 960MB

Der Typ der Partition sollte 0E sein (FAT16 / LBA-Modus). Ich weiss nicht genau, ob es auch mit 06er Partitionen funktionieren würde. Das habe ich noch nicht ausprobiert. Das Diskimage an sich war fast richtig. Es müsste 'CPMDSK\_D.IMG' heissen. Wobei die Suche nach den Images immer bei A beginnt und mit Z aufhört. Die CP/M Partitionen werden immer zuerst an den Anfang der internen Partitionstabelle gepackt. Und Diskimages landen immer dahinter. Die Imagedatei an sich sollte mindestens 256256 Bytes groß sein. Ist sie das noch nicht, so muss man sie erstmal mit Nullen auffüllen. Macht man das nicht, würde ein Schreibversuch hinter das Ende der Datei das Dateisystem evtl. beschädigen, wobei ich versuche diesen Fehler zu erkennen und abzufangen ! Ich habe es nur noch nicht getestet. Daher sind die Schreibroutinen derzeit in der [SVN](#) Version auch noch auskommentiert. Die Routinen können eine zu kleine Datei nicht vergrößern. Im moment sind Leo und ich dabei die Routinen zu testen und noch Fehler beim Schreibzugriff zu beseitigen. Hoffe es wird nicht mehr lange dauern, bis wir auch das im Griff haben.

Freundliche Grüße  
Frank

Autor: Frank Zoll ([fzoll](#))  
Datum: 06.10.2010 10:34

---

Ah, da fällt mir gerade noch was ein.

Das Booten von einer FAT16 Partition dürfte noch nicht funktionieren. Das ganz liegt an der Art wie das Startup in der Datei init.asm codiert wurde.

```
; Read first sector of first CP/M partition (ipl)
lds  xl,hostparttbl+1
lds  xh,hostparttbl+2
lds  yl,hostparttbl+3
lds  yh,hostparttbl+4
rcall mmcReadSect
```

Dieses Codestück muss noch angepasst werden, da bei einem Diskimage in der hoastparttbl an der Position 1-4 nicht der 1. Sektor der Datei abgelegt ist, sondern die Nummer des 1. Clusters der Datei. Hier greift der bestehende Lesezugriff an die falsche stelle der SD- Card. Ich werde die Stelle bei gelegenheit mal anpassen, damit man auch ganz ohne CP/M Partitionen ein System booten kann.

Grüße  
Frank

Autor: Leo C. ([rapid](#))  
Datum: 06.10.2010 17:05

---

[Frank Zoll schrieb:](#)

> Es müsste 'CPMDSK\_D.IMG' heißen. Wobei die Suche nach den Images immer  
> bei A beginnt und mit Z aufhört.

A - P würde genügen, da CP/M maximal 16 Laufwerke verwalten kann.  
Darüber hinaus vermute ich, daß gerade hier noch  
Optimierungsmöglichkeiten bestehen. Der Warmboot mit (nur) einem  
FAT16-Image dauert deutlich länger, als nur mit Partitionen. Mit  
eingeschalteten Debug-Messages sieht man eine deutliche Pause, nachdem  
alle FAT-Daten gelesen worden sind. Beim Lesen **und** Schreiben von  
CP/M-Dateien ist aber subjektiv kein Unterschied zum nackten  
Part-Treiber. Das FAT16-Dateisystem an sich ist also ausreichend  
schnell.

> Die Imagedatei an sich sollte  
> mindestens 256256 Bytes groß sein. Ist sie das noch nicht, so muss man  
> sie erstmal mit Nullen auffüllen. Macht man das nicht, würde ein  
> Schreibversuch hinter das Ende der Datei das Dateisystem evtl.  
> beschädigen, wobei ich versuche diesen Fehler zu erkennen und abzufangen  
> ! Ich habe es nur noch nicht getestet.

Aber ich jetzt. :-)

Der FAT16-Treiber weigert sich beharrlich, über das Dateiende hinaus zu  
lesen und zu schreiben.

> Daher sind die Schreibroutinen  
> derzeit in der [SVN](#) Version auch noch auskommentiert.

Jetzt nicht mehr. :-)

> Die Routinen können  
> eine zu kleine Datei nicht vergrößern. Im moment sind Leo und ich dabei  
> die Routinen zu testen und noch Fehler beim Schreibzugriff zu  
> beseitigen. Hoffe es wird nicht mehr lange dauern, bis wir auch das im  
> Griff haben.

Hier gibts die FAT16 Testversion:

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

oder

<http://cloudbase.homelinux.net/svn/avr-cpm/tags/2.0-fat16/>

Es fehlt noch etwas Feinschliff und die Tests waren bisher noch nicht  
sehr umfangreich. Mmnn sollte es aber mindesten so gut funktionieren, als  
der bisherige Code.

@Frank,  
tolles Stück Arbeit.

**Autor: Frank Zoll ([fzoll](#))**

**Datum: 07.10.2010 09:18**

---

Halloe.

Das Scannen nach den Image- Dateien ist wirklich nicht gerade schön. Bis  
alle 26 Buchstaben durchprobiert wurden müssten bei 512 Directory-  
Einträgen und 16 Einträgen pro Sektor insgesamt 832 Sektorlesezugriffe  
erfolgen. Pro Buchstabe der gesucht werden soll werden alle 32  
Direktory- Sektoren je einmal gelesen. Daher dauert es auch so lange.  
Dafür bekommt man halt die Laufwerke immer in die richtige Reihenfolge  
in der Partitionstabelle, egal in welcher Reihenfolge sie gerade auf der  
SD- Card im Verzeichniss liegen. Hier würde ich vorschlagen, das wir  
immer mit A anfangen und die Anzahl der Buchstaben auf die aktuelle

Größe der Partitionstabelle begrenzen. Mehr als diese Menge an Images können eh nicht verwaltet werden. Man ist dann halt dann gezwungen mit A anzufangen und die Images vorlaufend zu benennen, aber das sollte doch nichts ausmachen oder ?

@Leo  
Danke für die Hilfe bei den Tests.

Grüße  
Frank

Autor: Leo C. ([rapid](#))  
Datum: 07.10.2010 09:42

---

[Frank Zoll schrieb:](#)

> Man ist dann halt dann gezwungen mit A  
> anzufangen und die Images vorlaufend zu benennen, aber das sollte doch  
> nichts ausmachen oder ?

Sicher nicht (imho). Der größte Teil der Dateinamen ist ja sowieso hartcodiert.

Noch etwas anderes:  
Wie wichtig ist denn noch, daß das Programm noch auf [Atmega8](#) und [Atmega88](#) läuft?

Im Moment paßt nur noch die 4-Bit Version in den Speicher. Natürlich ohne FAT16. Die 8-Bit Version würde man noch mit etwas Geschwindigkeitseinbuße hinkriegen.

Autor: Leo C. ([rapid](#))  
Datum: 07.10.2010 21:11

---

[Frank Zoll schrieb:](#)

> Das Booten von einer FAT16 Partition dürfte noch nicht funktionieren.

@alle, die auf ihren SD-Karten nur FAT-Partitionen haben (wollen):

Das hat Frank inzwischen geändert. Man kann jetzt auch von einer Image-Datei im FAT16-Filesystem booten. Außerdem werden jetzt zwei FAT16-Partitionstypen erkannt. 0E und 06.

Das Update gibts hier:  
<http://cloudbase.homelinux.net/viewvc/avr-cpm/tags...>

## FAT16 fertig – VGA Teilprojekt

Autor: Frank Zoll ([fzoll](#))  
Datum: 08.10.2010 21:30

---

Halloe.

Da der FAT16- Treiber nun bereits recht gut läuft, hätte ich evtl. ein wenig Zeit über bei der Programmierung des VGA- Teils der Platine mit zu

helfen. Gibt es hier schon etwas, auf das man aufbauen könnte ? Bzw. ist schon jemand dabei etwas vorzubereiten und braucht ggf. noch Hilfe ?

Grüße  
Frank

p.s. Ich habe das TotalCommander Plugin das Leo gefunden hat mal kurz angetestet. Wenn man die diskdefs passend erweitert, kann man damit auch problemlos auf die Imagedateien zugreifen, die zu unserem netten kleinen Projekt hier gehören. Es muss also nicht unbedingt mit einem Laufwerk gearbeitet werden. Wer so wie ich GUI- Verwöhnt ist, dem kann ich nur den Tipp geben sich mal das Tutorial auf der Homepage anzuschauen. Damit klappts auf anhieb :-)

Autor: Leo C. ([rapid](#))

Datum: 09.10.2010 10:24

Angehängte Dateien:

- [cpm.zip](#) (181,8 KB, 46 Downloads)

---

Egmont schrieb:

```
> Bei mehr als 5 IMG-Dateien bekomme ich folgende Fehlermeldung  
> Version 136 vom SVN-Server.
```

```
> Bdos Err On E: Select
```

Die Anzahl Laufwerke ist an zwei Stellen begrenzt:

1.) dsk\_fsys.asm:

```
.equ MAXDISKS = 6 ;Max number of Disks (partitions)
```

2.) Z.Zt. noch im BIOS auf 4 Laufwerke.

Wenn Du MAXDISKS noch höher setzen willst, mußst Du den RAM-Verbrauch im Auge behalten. Ggf. kann man die UART-Buffer halbieren.

Im Anhang ist ein BIOS, das bis 8 Laufwerke gehen sollte. In dem Zip-Archiv sind auch die Quellen und die fertigen Binärdateien 'bios.bin', 'cpm.bin' und ein komplettes 'diskimage'. Leider habe ich es z.Zt. nicht besser.

Autor: Frank Zoll ([fzoll](#))

Datum: 09.10.2010 10:54

---

Halloe.

Ich habe mir in der letzten Nacht mal den VGA- Teil der aktuellen Platine angeschaut und das oben verlinkte Beispiel für das VGA-Betriebssystem genau unter die Lupe genommen.

Mitlerweile habe ich den Code der Vorlage aufgeräumt und so angepasst, das er sich auch für den 328er Übersetzen lässt. Nun ist das SRAM erstmal zu 197% gefüllt, wegen des zu großen Framebuffers. Als nächstes werde ich den Puffer so weit verkleinern, das er gerade noch so eben in das SRAM passt.

Findet sich hier evtl. jemand der lust hat einen spezeillen, neuen, Zeichensatz zu basteln ? Wir brauchen einen Zeichensatz von 6 Pixel breite verpackt in jeweils 1 Byte pro Zeile. Die höhe weiss ich noch nicht genau, weil ich die endgültige Größe des Framebuffers noch nicht ausprobiert habe. Bei den oben angedachten 25 Zeilen wäre die Höhe für den Zeichensatz 16 Pixel(Zeilen). Ich plane jedoch möglichst viele Zeilen aufs Display zu quetschen, da würden dann ggf. noch ein paar Zeilen im Zeichensatz wieder wegfallen.

Die 80 Zeichen pro Zeile scheinen machbar zu sein, unschön ist nur das zwischen jedem Zeichen 2 spalten dunkel bleiben, die wir nicht werden ansteuern können. Damit läßt sich dann wohl leider mit dem neuen Zeichensatz keine durchgehende horizontale Line erreichen. Umgehen läßt sich dieses Problem nicht so einfach, da der Port an dem das Schieberegister hängt nur 6 der eigentlich im Original vorgesehenen 8 Bits liefern kann. Die Routine, die die Zeihen ausgiebt, hat leider keine reserven mehr um da noch Befehle hinzunehmen zu können, die die fehlenden 2 Bits irgendwie ersetzen könnten. Vielleicht hat hier ja jemand noch eine super Idee ?

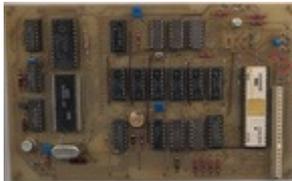
Kennt sich hier evtl. jemand mit den CP/M Terminals aus und kann mir da Helfen, heraus zu bekommen was so ein Terminal alles können muss außer bloß ein par Zeichen anzuzeigen ? Da scheint im Originalcode noch eine große Lücke zu klaffen, die wir noch füllen müssen.

Grüße  
Frank

Autor: Leo C. ([rapid](#))

Datum: 09.10.2010 11:43

Angehängte Dateien:



[Videokarte-800.jpg](#)

78,6 KB, 346 Downloads

---

[Frank Zoll schrieb:](#)

> Kennt sich hier evtl. jemand mit den CP/M Terminals aus und kann mir da  
> Helfen, heraus zu bekommen was so ein Terminal alles können muss außer  
> bloß ein par Zeichen anzuzeigen ? Da scheint im Originalcode noch eine  
> große Lücke zu klaffen, die wir noch füllen müssen.

Unbedingt:

- Clear screen
- Cursor positionieren (x,y)
- ein Attribut (eins aus:highlight, dim, inverse, underline, blink (igit))

Gut:

- Clear to end of line
- Delete line (nachfolgende zeilen rutschen hoch)
- Insert Line (Zeilen unterhalb der Curserposition eine Zeile nach unten)
- mehr Attribute

ANSI oder VT100 war zu CP/Ms Hochzeiten nicht verbreitet. Ist wahrscheinlich auch etwas aufwendig zu implementieren. Arbeitspferd war das Lear Siegler ADM-3A und viele Kompatible. Praktisch jedes CP/M-Programm mit Terminalsteuerung unterstützt das.

<http://www.tentacle.franken.de/adm3a/>

Das scheint aber wirklich nur die unbedingt notwendigen Funktionen zu haben.

Das nächst bessere, und auch sehr weit verbreitet, war dann Televideo 920 oder so.

Mein MC-Terminal, daß seit einigen Wochen wieder hier rumliegt und auf Inbetriebnahme wartet, ist auch Televideo-kompatibel.

Und weils so schön ist, in alten Erinnerungen zu schwelgen, im Anhang ein Bild von meiner ersten Terminal-Karte (Elektor, 64 Zeichen x 16 Zeilen, max 1200 Baud und 120ms für den clear screen)

Autor: Joe G. ([feinmechaniker](#))

Datum: 09.10.2010 11:46

---

[Frank Zoll schrieb:](#)

> Die 80 Zeichen pro Zeile scheinen machbar zu sein, unschön ist nur das  
> zwischen jedem Zeichen 2 spalten dunkel bleiben, die wir nicht werden  
> ansteuern können.

War leider mit dem ATmega328 nicht anders lösbar.

> Die Routine, die die Zeilen ausgibt, hat leider  
> keine reserven mehr um da noch Befehle hinzunehmen zu können, die die  
> fehlenden 2 Bits irgendwie ersetzen könnten. Vielleicht hat hier ja  
> jemand noch eine super Idee ?

Vorschlag zur Ausgabe einer durchgehenden Linie.

PB4 und PB5 an das Schieberegister und vor der Zeilenausgabe die beiden Bits gesetzt. Zwischen Backporch und Frontporch ist ja noch etwas Platz für den Code.

Joe

Autor: Leo C. ([rapid](#))

Datum: 09.10.2010 12:10

---

[Leo C. schrieb:](#)

Vergessen:

> Unbedingt:  
> - Clear screen  
  - Cursor links (back space)  
> - Cursor positionieren (x,y)  
> - ein Attribut (eins aus:highlight, dim, inverse, underline, blink)

> Gut:  
> - Clear to end of line  
> - Delete line (nachfolgende zeilen rutschen hoch)  
> - Insert Line (Zeilen unterhalb der Curserposition eine Zeile nach  
> unten)  
- Erase to end of screen

- Insert Character
- Delete Character
- Cursor right, up, down
- > - mehr Attribute

Autor: Frank Zoll ([fzoll](#))

Datum: 09.10.2010 17:43

---

Hm,

mitlerweile habe ich den VGA-Code ans laufen bekommen. Zufriedenheit sieht aber anders aus. Ich bekomme auf meinem alten Monitor einen klaren Sync bei 640\*400 @74Hz. Aber von meinem Beispieltext im Framebuffer ist nichts zu sehen. :-)

Grüße  
Frank

Autor: Frank Zoll ([fzoll](#))

Datum: 09.10.2010 18:31

---

Joe G. schrieb:

> Frank Zoll schrieb:

>> Aber von meinem Beispieltext im Framebuffer ist nichts zu sehen. :-)

>

> Hast du eines von den 74AC08 Gattern (IC7) auch auf High gesetzt?

> Ansonsten schiebt das Schieberegister nichts auf die VGA-Buchse.

Halloe.

Ich denke, das müsste gerade schön bunt aussehen. Hier das Setup für die Fordergrundfarbe.

Das Setup wird im moment unmittelbar vor der Ausgabeschleife aufgebaut. Alles was nicht unbedingt für das Terminal gerbaucht wurde hab ich mitlerweile raus geworfen. Im moment versuche ich gerade statt des Fonts ein einfaches 10101010 Streifenmuster zu erzeugen. Auch das tut leider nicht. Ich werd mir wohl mal die Hardware unter der Lupe anschauen müssen, evtl. hab ich da noch nen Bug drinn :-)

```
// farbe einstellen
```

```
// Pin 5-7 wieder auf Ausgang und andere im ursprünglichen Zustand belassen:
```

```
DDRD  |= ( 1 << DDD5 ) | ( 1 << DDD6 ) | ( 1 << DDD7 );
```

```
PORTD |= ( 1 << PD5 ) | ( 0 << PD6 ) | ( 1 << PD7 );
```

Schade das ich das ganze nicht nachmessen kann, ich habe weder ein Oszi noch einen Logikanalyser zur Hand.

Grüße  
Frank

Autor: Frank Zoll ([fzoll](#))

Datum: 09.10.2010 22:04

Angehängte Dateien:



• [Dscf5670.jpg](#)

104,2 KB, 430 Downloads



• [Dscf5673.jpg](#)

140,8 KB, 453 Downloads

---

Halloe.

Habe die letzten blödden Fehler endlich gefunden. Hier mal zwei Bilder des aktuellen Standes :-)

Die VGA-Ausgabe läuft. Es ist gerade so eben Platz für 80 Zeichen pro Zeile und insgesamt 22 Zeilen.

SRAM ist zu 96,4% gefüllt. Bissel was brauche ich noch für den Stack. Nun brauchen wir einen Zeichensatz wie oben beschrieben. (6 Pixel Breit, 18 Pixel Hoch) Je ein Byte pro Zeile. Als nächstes muss ich die Terminalverbindung reaktivieren und mich um einen Software UART für die Tastatur kümmern, die ich abschalten musste weil der 328er nur 1 UART zur Verfügung hat.

Grüße  
Frank

p.s. Mit Attributen (Blinken, Invers, Durchgestrichen etc.) pro Zeichen, das wird nix mehr, das RAM ist voll....

Autor: Frank Zoll ([fzoll](#))

Datum: 10.10.2010 01:17

---

Halloe.

Ich habs geschafft, 24 Zeilen ins Ram rein zu quetschen. Damit wären wir gerade noch konform mit den oben genannten Standard- Terminals. Ich habe nur noch keine Idee was wir mit den Attributen machen. Da die Terminals eigentlich alle mit 7 bit / Zeichen arbeiten, hätten wir noch 1 bit über. Dies Bit könnten wir zum Beispiel nutzen um sowas wie einen Inversen- Modus zu unterstützen. Bei 24 zeilen ist das Ram fast voll und ich habe noch nicht die Routinen für die Tastatur fertig, daher möchte ich mal nicht versprechen das ich evtl. wieder Zeilen rausstreichen muss um das ganze Lauffähig zu halten.

Grüße  
Frank

Autor: Leo C. ([rapid](#))

Datum: 10.10.2010 16:03

Angehängte Dateien:

- [myz80-img.zip](#) (8,1 KB, 44 Downloads)
- [avrcpm-log.txt](#) (6 KB, 186 Downloads)

---

Ich habe mal meinen aktuellen Softwarestand hochgeladen (trunk).

Achtung: Diese Version bitte nicht mit Disk Images testen, die wichtige Daten enthalten. Es könnte sein, daß unter bestimmten Bedingungen ein falscher Sektor geschrieben wird.

Ich habe die maximale FAT16-Image-Dateigröße und die maximale CP/M-Partitionsgröße auf 32MB begrenzt. Dh., CP/M-Dateisysteme können maximal 32MB groß werden. Sieht hier jemand ein Problem?

Das Programm kann jetzt verschiedene Diskformate erkennen und einbinden. Im Moment sind das die Images vom MyZ80 Emulator und von YAZE-(AG). In der YAZE-AG Distrib sind verschiedene Diskimages (\*.ydsk-Dateien), die fast alle unterschiedliche Größen haben.

<http://www.mathematik.uni-ulm.de/users/ag/yaze/>

Diese Formate können auch mit den cpmtools verarbeitet werden.

Optimal sind aber weder MyZ80 noch das YAZE-Format. Beide haben einen Header, der nicht der physikalischen Sektorgröße der Speicherkarten entspricht (MyZ80 256 Byte und YAZE 128 Byte). Das kostet Performance, weil die CP/M Datenblöcke dann immer mitten im Sektor beginnen und enden, und der Teil davor, bzw. danach auch immer gelesen, bzw. geschrieben werden muß.

Für unser System würde ich ein Format mit einem 1 Sektor (512 Byte) großen Header bevorzugen. Das ließe sich mit der jetzt vorhandenen Software verwenden.

Da aber Header unbeliebt zu sein scheinen ;), könnte man auch noch was mit fest vorgegebenen Formatgrößen realisieren. Die Software dazu ist im Prinzip auch vorhanden. Frank hat mir dazu schon den Vorschlag geschickt, das Format über den Dateinamen zu bestimmen. Das geht auch, allerdings nur mit FAT16 und natürlich nicht mit direkten Partitionen.

Eine andere (oder zusätzliche) Möglichkeit wäre, einige Laufwerksbuchstaben (z.B. M-P) für feste Formate zu reservieren. Dazu müßte man sich aber auf ein bis maximal 2 Formate einigen. Je größer die Disk ist, und vor allem, je mehr Directory-Einträge sie hat, um so langsamer wird der Zugriff. Große Disks mit kleinen Directories machen aber keinen Sinn.

In der angehängten Zip-Datei ist ein leeres MyZ80 Image. Die Datei wird beim Auspacken 8MB groß.

In den nächsten Tagen möchte ich das ganze nochmal überarbeiten (und das Chaos etwas aufräumen). Inzwischen ist mir noch eine deutliche Verbesserung zu der Schnittstelle zum 8080 BIOS-Teil eingefallen.

Eine andere Baustelle ist noch das Verhalten bei Diskwechsel mit und ohne Warmstart. Verschiedene Softwarestände verhalten sich inzwischen unterschiedlich, und mir ist noch nicht ganz klar wie es richtig geht. Im Moment muß nach Ziehen und Stecken der Karte ein Warmstart erfolgen, damit CP/M wieder auf die Karte zugreifen kann.

Wenn man allerdings die Karte immer automatisch neu initialisiert, wenn sie gewechselt wurde, besteht die Gefahr, das Daten auf die falsche

Karte geschrieben werden.

Bitte probiert das mal aus und macht Vorschläge.

Autor: Frank Zoll ([fzoll](#))

Datum: 10.10.2010 19:49

Angehängte Dateien:

- [avr\\_cpm\\_vga.zip](#) (69,1 KB, 65 Downloads)

---

Halloe

Die Hardware sowohl von der CP/M Emulatorseite als auch auf der VGA-Seite läuft gänzlich ohne Lötpatches stabil. Die VGA-Seite habe ich mit 25Mhz wie gefordert übertacktet. Die CP/M Seite läuft auf 20 Mhz, hier habe ich keinen Übertacknungsversuch gemacht. Auf VGA-Seite habe ich nur einen blöden Softwarefehler gehabt. Mittlerweile läuft auch schon die einfache Textausgabe über UART von der CP/M Seite aus. Das Keyboard habe ich noch nicht am laufen und der Font ist auch noch nicht fertig. Die Anzeige sieht wegen des falschen Fonts noch ziemlich kaputt aus, aber man kann die Zeichen mit gut will schon erkennen.

Grüße  
Frank

Ich habe mal hier meine aktuelle "Spielversion" angehängen. Das Hex-File ist für den 328er mit 25Mhz Quarz compiliert. Die UART-Geschwindigkeit ist 9600 Baud 8N1. Hier muss man auf der CP/M Seite von 36800 Baud noch eben runtergelen, sonst klappt die Anzeige nicht. :-)

Autor: Leo C. ([rapid](#))

Datum: 11.10.2010 10:47

---

Egmont schrieb:

```
> Hallo,  
> wo bekomme ich zum übersetzen von ipl.asm und bios.asm die Datei  
> z80asm.exe her. Googeln brachte bei mir kein Erfolg.  
> Ein Link oder eine Datei wäre nicht schlecht (für Windows).  
> Was ist mit bios.mac, für was wird die benötigt....anderer  
> Compiler/System ?
```

bios.asm ist zu alt. Aktuell ist bios.mac. Dazu gehören auch die beiden .lib Dateien.

bios.mac und ipl.mac können mit M80 (MACRO-80) von Microsoft auf Deinem CP/M-System übersetzt werden. :)

Wahrscheinlich gibts kompatible Crossassembler, die unter Windows laufen, aber ich habe noch keinen gefunden.

Es gibt auch CP/M-Emulatoren, die einen Befehl abarbeiten, der auf der Befehlszeile übergeben wird. "zxcc" ist so ein Ding, und funktioniert bei mir unter Linux wunderbar. Laut der Homepage sollte es aber auch unter DOS (Windows) laufen:

<http://www.seasip.demon.co.uk/Unix/Zxcc/>

Wenn zxcc richtig installiert ist, kann man das BIOS so assemblieren und linken:

```
$ zxcc m80 -=bios.mac
```

```
$ zxcc l80 -bios.rel,bios.bin/N/E
```

Das Makefile erledigt diese Schritte automatisch.

Für Windows habe ich noch eine eine Alternative gefunden: "m80l80pc.zip"  
Mit Google leicht zu finden.

Wenn Die Dateien aus dem Paket im Pfad zu finden sind, können M80 und L80 direkt auf der Befehlszeile gestartet werden. Das ganze ist offensichtlich mit dem CP/M-Emulator "22nice" gebaut, der ebenfalls leicht zu finden ist, und vielleicht auch sonst einen Versuch wert wäre.

Im m80l80pc.zip Paket ist auch das M80/L80 Handbuch als Textdatei. Sonst habe ich immer nur gescannte Handbücher als PDF gefunden.

Meine Bitte an die Windowsbenutzer wäre, daß mal jemand die 2 Varianten ausprobiert. Wenn m80l80pc die bessere Alternative ist, würde ich das mit ins Makefile aufnehmen. Das aktuelle Makefile mit zxcc sollte auch unter Windows ohne Änderung laufen.

Autor: Peter Zabel ([flexopete](#))

Datum: 11.10.2010 18:15

Angehängte Dateien:



• [VGA.JPG](#)

540 KB, 472 Downloads

---

Hallo,

habe mal versucht, ein Foto von der Bilschirmausgabe des VGA-Teils zu machen. Es ist ein 15" TFT.

Gruss  
Peter

Autor: Frank Zoll ([fzoll](#))

Datum: 11.10.2010 20:10

---

Hallo Peter.

Genau so siehts bei mir derzeit auch aus. Da der Font noch nicht angepasst ist, fehlen an den Zeichen rechts immer 2 Pixel. Und da die Höhe des Fonts nur 8 und nicht 16,667 Pixel ist, liest er am Ende über den Pufferspeicher hinaus. Als nächstes muss ich erstmal einen vernünftigen Font basteln und die Ausgaberroutine daran anpassen. Danach kann ich daran gehen, die fehlenden Funktionen des VT100 Terminals nachzubauen.

Grüße  
Frank

Autor: Peter Zabel ([flexopete](#))

Datum: 11.10.2010 20:32

---

Moin Frank,

fehlt nicht eher links was oder hängt das mit dem Schieberegister zusammen? Wenn Du mir kurz erklären, wie es aufgebaut ist, kann ich Dich beim Font unterstützen. Ist ja eher ne Fleissarbeit. Verstehe ich es richtig, dass in der font.c jede Spalte ein 8x8 Zeichen ist? 1.Spalte 0x00 bis 0xff. Sind dies schon die ASCII Adressen? Bei 16x6 werden die Zeichen rel. schmal, obwohl, es geht ja min. 1 linie als Zwischenraum weg.

Gruss  
Peter

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 11.10.2010 22:12**

---

Hallo Peter.

Ich habe schon einen einfachen Font angefangen. Im moment fehlen mir "nur" noch die Kleinbuchstaben. Und ich muss das File nacher noch passend aufbereiten. Ich spiele gerade mit einem ganz einfachen Fonteditor rum.

Der Font wie er in der Font.c abgelegt ist, ist praktisch Zeilenweise abgelegt. Pro Zeile eines Zeichens wird immer genau 1 Byte verwendet. Jedoch sind nicht etwa alle Zeichen einfach nacheinander Zeile für Zeile abgelegt.

Die Tabelle ist so angelegt, das jeweils eine Spalte von oben nach unten gesehen ein Zeichen bildet. Von Links nach Rechts sind die Zeichen von ASCII- Code 0 bis 255 aufsteigend abgebildet.

Das erleichtert der Bildaufbauroutine den Zugriff ganz enorm.

Mit der Aussage, es fehlt eher links was könnte stimmen. Wenn man die Bits eher so Zählt 76543210, dann würde er links 2 Bits wegwerfen. Auf jeden Fall werden nur die 6 niederwertigsten Bits auf den Port ausgegeben. Und das Abschneiden an der Linken Seite würde auch zu deinem Foto besser passen.

Grüße  
Frank

**Autor: Peter Zabel ([flexopete](#))**  
**Datum: 11.10.2010 23:31**  
**Angehängte Dateien:**



[FONT.jpg](#)

111,3 KB, 352 Downloads

---

Hallo Frank,

ich habe mal ein paar Buchstaben der 8x8 aufgemalt. Bei "B" und "D" sieht man den senkrechten Strich noch auf dem Monitor, weil sie eine Serife haben. Es fehlen also links 2 Bit. Die Ziffern rechts sind die cursor Positionen in der font.c im [AVR](#)-Studio.

Gruss  
Peter

## Simh hdisk Format für avrcpm

Autor: Leo C. ([rapid](#))

Datum: 11.10.2010 23:59

---

Neues Format.

Ich habe mir heute nochmal die Formate des genialen simh Altair 8800 Emulators angeschaut. Das Diskettenformat kann man vergessen. Das Harddiskformat ist aber ganz simpel und ich habe es schon eingebaut.

Kleiner Wermutstropfen: Das Format hat zwar 6 Systemspuren a 32 Sektoren reserviert, aber wenn man dort ein System zum booten einspielt, kann es nicht mehr erkannt werden. Das werde ich aber irgendwann mal ändern.

Zu dem Emulator gibt es eine ganze Menge Diskimages mit guter Software. Allerdings sind das alles Diskettenimages. Aber man kann sie ja im Emulator einfach auf "Harddisk" umkopieren. Hier kann man alles runterladen:

<http://www.schorn.ch/cpm/intro.php>

Hier ist eine Formatbeschreibung für die cpmtools:

```
# SIMH AltairZ80 Harddisk
diskdef simhd
  seclen 128
  tracks 2048
  sectrk 32
  blocksize 4096
  maxdir 1024
  skew 0
  boottrk 6
  os 2.2
end
```

## Weiter mit VGA-Teil

Autor: Frank Zoll ([fzoll](#))  
Datum: 12.10.2010 00:43  
Angehängte Dateien:

- [avr\\_cpm\\_vga.zip](#) (71,7 KB, 60 Downloads)

---

Halloe.

Es ist schon späht, aber ein kleines Update möchte ich euch noch zum anschauen überlassen. Im Anhang findet ihr eine erste Version der Routinen mit einem 6\*16 Font. Bitte nicht zu genau drauf schauen, ich habe den Font mal eben schnell komplett neu gezeichnet. Und ich bin bei weitem kein Mahlkünstler :-)

Leider habe ich keinen wirklich guten kostenlosen Editor für Windows gefunden. Man sieht beim Malen nicht, wie es nacher im gesamten wirklich aussieht. Ich werde den Font nochmal ein bisschen überarbeiten, aber man kann nun wenigstens schonmal was erkennen. :-) Etwa in der Art wird es nacher ausschauen.

Da ich noch ein "einfaches" Attribut umsetzen könnte frage ich einfach mal. Invertiert oder lieber Unterstrichen, was hättet ihr lieber ?

Grüße  
Frank

p.s. die 25. Zeile mit dem Müll darin werde ich in der nächsten Version ausblenden, so das sie nicht mehr stört.

## SVN Umstrukturierung

Autor: Leo C. ([rapid](#))  
Datum: 12.10.2010 13:10

---

Ich habe das [SVN](#)-Repository etwas umstrukturiert. Alte Links aus dem Forum werden wohl nicht mehr funktionieren, sorry. Vom Top-Level aus dürfte es aber kein Problem sein, alles wieder zu finden.

In avrcpm/trunk gibt es ein neues "tools" Verzeichnis, in dem jetzt Franks Programm zum Vergrößern von Images zu finden ist. Da das Programm jetzt mit E5 Hex statt mit Nullen füllt, kann man es auch zum Erzeugen neuer Images verwenden. Unter Linus geht das z.B. so:

```
$ ./makeimage /dev/null myz80.img $(expr 128 \* 65536 + 256)  
$ ./makeimage /dev/null simhd.img $(expr 128 \* 65536)
```

Unter Windows müßte es so ähnlich auch gehen. Falls es dort kein 'expr' oder ähnliches gibt, muß man die Größe selbst ausrechnen.

Das Resultat kann man in eine CP/M- oder unter passendem Namen, in eine FAT16- Partition kopieren, und man hat ein frisch formatiertes CP/M Dateisystem.

Außerdem gibt es noch ein Verzeichnis 'avrvga', das für die Terminalsoftware von Frank vorgesehen ist. (Änderungen vorbehalten)

# Weiter mit VGA-Teil - Tastatur

Autor: Frank Zoll ([fzoll](#))

Datum: 13.10.2010 23:09

---

Halloe.

Heute habe ich mich erstmal ein wenig um die Terminalemulation gekümmert. Am Bildschirmende wird nun nicht mehr einfach der ganze Bildschirm gelöscht. Statt dessen Scrollt der ganze Bildbereich nun eine Zeile aufwärts. Hier habe ich leider noch ein sehr eckliges Flackern beim Scrollen, das ich noch nicht weg bekommen habe.

Auch um die Tastaturanbindung habe ich mich schon ein wenig kümmern können. Bisher habe ich die Interruptroutine auch noch nicht ans laufen bekommen. Im moment plagt mich noch die befürchtung, das der Tastaturinterrput evtl. den gesammten Bildaufbau durcheinander bringen könnte bei jedem Bit, das die Tastatur an den Controller schickt. Leider habe ich hier weder ein Oszilloskop noch einen Logiganalyzer, so das ich im moment ein wenig im dunkeln stochere was das Problem angeht.

Grüße  
Frank

Autor: Sebastian B. ([sfreak](#)) 

Datum: 13.10.2010 23:25

---

[Frank Zoll schrieb:](#)

```
> Auch um die Tastaturanbindung habe ich mich schon ein wenig kümmern
> können. Bisher habe ich die Interruptroutine auch noch nicht ans laufen
> bekommen. Im moment plagt mich noch die befürchtung, das der
> Tastaturinterrput evtl. den gesammten Bildaufbau durcheinander bringen
> könnte bei jedem Bit, das die Tastatur an den Controller schickt. Leider
> habe ich hier weder ein Oszilloskop noch einen Logiganalyzer, so das ich
> im moment ein wenig im dunkeln stochere was das Problem angeht.
```

Du willst für die Tastatur einen interruptgesteuerten Software-UART benutzen? Ich weiß nicht wie es bei dir jetzt aussieht, aber in meinem Code war während einer Videozeile kein einziger Takt übrig. Deshalb habe ich die Arbeit einem Hardware-UART überlassen und nur in der horizontalen Austastlücke ggf. ein empfangenes Byte zwischengespeichert. Wenn du auf jedes Bit reagieren musst, wird das nicht reichen.

Sebastian

Autor: Frank Zoll ([fzoll](#))

Datum: 13.10.2010 23:45

---

Hallo Sebastian.

Ja, ich muss die Arbeit notgedrungener Massen selber machen. Der 168er [AVR](#) Controller hat leider nur einen UART. Und der ist bereits für die Verbindung zwischen dem VGA- Teil und dem CP/M Emulator belegt. Im moment gehe ich davon aus, das ggf. jedes einzelne Bit das von der Tastatur kommt alles durcheinander bringen könnte, wenn der Interupt

während der Pixelausgabe zuschlägt. In der vorliegenden Hardware ist der Clockausgang der Tastatur mit dem INT1- Eingang des Controllers verbunden. Im moment versuche ich gerade diesen Interrupt überhaupt erstmal ans laufen zu bekommen. Soweit meine Recherchen ergeben, sendet eine Standarttastatur mit einer Taktrate von 20 bis 30 Khz. Bei der derzeitigen Auflösung mit 400 aktiven Zeilen bei 74HZ, hätten wir rechnerisch 29,6 KHz abtastfrequenz, wenn wir es schaffen wären des horizontales Blanks jeweils einen Tatsaturtakt abzuarbeiten. Das könnte gerade so eben hinkommen würde ich sagen. Ich weiss nur noch nicht, wie ich den Intterrupt so lange heraus zögern kann, bis eine Grafikzeile vollständig abgearbeitet wurde. Gänzlich verschlucken darf ich dabei ja auch keinen einziegen Takt, sonst komme ich bei der Verarbeitung der Datenbits durcheinander. Bin mal gespannt, ob ich das ganze irgendetwas in den Griff bekomme.

Grüße  
Frank

Autor: Sebastian B. ([sfreak](#)) 

Datum: 13.10.2010 23:54

---

Die Verzögerung des INT1 passiert ganz von alleine:  
Die Videozeile wird in einer ISR ausgegeben. Tritt während diese ISR läuft der ext. Interrupt auf, wird nur das entsprechende INT1 Interrupt-Flag gesetzt. Die INT1 ISR wird erst ausgeführt wenn die Video ISR zurückkehrt.

Da Zeilen- und Tastaturtakt aber nicht synchron sind, wird früher oder später einen Takt verloren gehen.

(Verschachtelte Interrupts gehen auch, hab ich noch nie benutzt:  
[http://www.nongnu.org/avr-libc/user-manual/group\\_\\_\\_...](http://www.nongnu.org/avr-libc/user-manual/group___...) )

Autor: h3aau (Gast)

Datum: 14.10.2010 12:27

---

Nur mal so der Neugierde habler.....

640x400 Pixel heisst doch  $640 / 8 = 80$  Zeichen.

$400 / 8 = 50$  Zeilen.

Ein Zeichen besteht also aus einer 8x8 matrix.

Bei dieser werden nur die Bits xx543210 horizontal verwendet.

Bit 76xxxxxx sind die Zwischenräume der Zeichen.

In der vertikalen werden nur die Zeilen 0,1,2,3,4,5, und 6 verwendet.

Zeile 7 ist der Abstand zwischen den Zeilen.

Also besteht ein Zeichen eigentlich aus einer 6x7 Matrix.

z.B. xx543210xx543210

1111 1111 = 0x3C,0x3C

1 1 1 11 = 0x41,0x13

1 1 1 11 = 0x41,0x13

111111 11111 = 0x3F,0x3E

1 1 1 11 = 0x41,0x13

1 1 1 11 = 0x41,0x13

1 1 1111 = 0x41,0x3C

= 0x00,0x00

Bei 25Mhz Taktung gehen nur die 640x350, 640x400 und 640x480 Modis.

80x43 80x50 80x60

Wollt ihr die uVGA auf 80x25 umstellen?

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 14.10.2010 13:09

---

[h3aau schrieb:](#)

```
> Bei 25Mhz Taktung gehen nur die 640x350, 640x400 und 640x480 Modis.  
>                               80x43   80x50   80x60  
> Wollt ihr die uVGA auf 80x25 umstellen?
```

Geplant war 640x400 also 80x50 und dann jede zweite Zeile frei, also 80 Zeichen zu 25 Zeilen. Mehr geht eh nicht in den RAM.

Autor: fzell (Gast)

Datum: 14.10.2010 13:45

---

Halloe.

Wir haben auf 80\*24 Zeichen umgestellt mit einem neuen Font von 6(Breit) \* (16) Hoch Pixeln grÖÙe. Wobei nach jedem 6 Pixel breiten Zeichen 2 schwarze Pixel gesendet werden. 25 Zeilen passen nicht in den Speicher, weil dies mit den benötigten Variablen für den UART und den Keyboard-Puffer sowie die globalen Statusvariablen und den Stack kollidieren würde. Die Auflösung bleibt aber 640\*400 wobei die letzten 8 Zeilen einfach nur schwarz ausgegeben werden.

GrüÙe  
Frank

p.s. Für den Tastaturanschluss siehts ganz düster aus. Der Interrupt haut wenn er prio bekommen würde 2 mal pro Bidlschirmzeile rein und würde das ganze Bild durcheinander bringen. Und da die Clockerzeugung von der Tastatur vorgegeben wird, können wir darauf keinen Einfluss nehmen.

Autor: h3aau (Gast)

Datum: 14.10.2010 15:13

---

also:

```
z.B.  xx543210xx543210  
0      1111  1111   = 0x3C,0x3C  
  
1      1    1    1  11 = 0x41,0x13  
  
2      1    1    1  11 = 0x41,0x13  
  
3      111111 111111 = 0x3F,0x3E  
  
4      1    1    1  11 = 0x41,0x13  
  
5      1    1    1  11 = 0x41,0x13  
  
6      1    1  1111   = 0x41,0x3C  
  
7                                     = 0x00,0x00
```

geht das?

Font als 6x16, alle ausgeben und Unterlängen nutzen?

```
z.B.  xx543210xx543210  
0      1    1           = 0x21,0x00  
1                                     = 0x00,0x00  
2      1111           = 0x3C,0x00  
3      1    1           = 0x21,0x00
```

4	1	1		= 0x21,0x00
5	111111	1111		= 0x3F,0x3E
6	1	1	1 11	= 0x21,0x13
7	1	1	1 11	= 0x21,0x13
8	1	1	1111	= 0x21,0x3C
9			1	= 0x00,0x02
10			1	= 0x00,0x01
11		1	1	= 0x00,0x12
12			11	= 0x00,0x0C
13				= 0x00,0x00
14				= 0x00,0x00
15				= 0x00,0x00

**Autor: fzoll (Gast)**

**Datum: 14.10.2010 17:35**

---

Halloe.

Ja das mit den unterlängen geht Problemlos. Ich habe es bei meinem Font auch so gemacht. Die "Nulllinie" liegt dort auf der 12. Zeile. So sehen die Buchstaben nacher im Display recht gut aus. Nur so Buchstaben wie W und M die eigentlich besonders breit sind sehen bei 6 Punkten breite nicht berauschend aus. Aber Lesbar ists auf jeden Fall geworden würd ich sagen.

Ach ja, nochwas. Da CP/M Terminals nur 7bit ASCII- Codes verwendet haben, habe ich das 8. Bit dafür genutzt Inverse Zeichen möglich zu machen. Dazu habe ich alle Zeichen von 0-127 in den Bereich 128-255 kopiert und die Bits negiert. So kann ich später zumindest das Attribut "invers" mit unterstützen. Wahlweise hätte man auch "unterstrichen" nehmen können. Blinken z.B. würde nicht gehen, da wären der Ausgabe der Pixel kein einziger Tackt mehr für Sonderfunktionen über ist.

Grüße  
Frank

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 15.10.2010 09:04**

---

[fzoll schrieb:](#)

```
> Für den Tastaturanschluss siehts ganz düster aus. Der Interrupt
> haut wenn er prio bekommen würde 2 mal pro Bidschirmzeile rein und
> würde das ganze Bild durcheinander bringen. Und da die Clockerzeugung
> von der Tastatur vorgegeben wird, können wir darauf keinen Einfluss
> nehmen.
```

Einfach mal eine Idee.

A.)

Tastatur auf der VGA Platine kommt an die Hardware UART und die Verbindung zum CP/M System erfolgt über eine Soft-UART analog der schon realisierten.

B.)

Auf dem CP/M System sind ja noch zwei Leitungen frei. Über diese beiden Leitungen wird die Tastatur angeschlossen. Da Sie beide am Verbindungssteckverbinder anliegen, sollte das ohne großen Hardwareänderungsaufwand zu realisieren sein.

Joe

**Autor: fzoll (Gast)**

**Datum: 15.10.2010 10:12**

---

Hallo Egmont und Joe.

Leider ist die Idee von Egmont nicht brauchbar. Das Problem ist, das die Tastatur zu langsam wäre die gesamten 11 Byte eines Zeichens während des Vertical-Syncs zu senden. Und man kann die Übertragung nicht mittendrin anhalten. Sobald der Host (VGA-Controller) sagen würde, das er nicht mehr zum Empfang bereit ist, wird nach Standard die Übertragung komplett abgebrochen. Eine Pause ist im Protokoll nicht vorgesehen. Leider.

Joe's Ideen an den UARTS und der Platine etwas zu verändern würden wohl am meisten bringen.

Meine Idee zieht aber eher gerade in eine etwas andere Richtung. Ich wollte die schöne Platine nicht "kaputt" machen. Statt dessen denke ich gerade darüber nach mir einen ganz kleinen AVR-Controller auszuschauen und ihn einfach als "Adapter" zwischen den VGA-Controller und die Tastatur zu hängen. Dieser hätte dann genügend Zeit um alle Zeichen, die die Tastatur sendet, zwischen zu puffern. Und die beiden bestehenden Datenleitungen aus dem PS/2 Port der Platine würde ich dann nutzen um vom Adaptercontroller die Bits Stück für Stück abzuholen. Wobei ich dann den Takt während der horizontalen Sync-Zeiten auf Seiten des VGA-Controllers generieren würde.

Wenn wir es schaffen, da einen 4Pinner zu finden und zu programmieren, dann könnte man das Teil super einfach mit ein wenig Schrumpfschlack in ein kleines Adapterkabel einbauchen und wird müssten die schöne Platine nicht patchen.

Für eine Neuauflage würde ich das ganze dann gleich etwas anders designen. Ich würde für den VGA-Teil einen Propeller Chip nehmen, der hätte genügend Power um VGA/TV, samt Tastatur und Maus zu handeln. (DIP40 Gehäuse). Da wären dann auch die VT100-Attribute möglich. auch in hohen Auflösung....

Grüße

Frank

**Autor: fzoll (Gast)**

**Datum: 15.10.2010 12:14**

---

Halloe.

Leider kann der HOST den PS/2 Clocktakt nicht beeinflussen. Das Clock-Signal wird von der Tastatur generiert. Die Geschwindigkeit schwankt hier je nach Tastatur zwischen ca. 10kHz und 30kHz. Man könnte nun natürlich eine Suche starten und schauen ob man eine Tastatur finden kann, dessen Clock-Signal zu unserem VGA-Signal "compatibel" ist.

Man kann auch kaum voraus sagen, wann die Tastatur ein Bit senden/empfangen möchte. Im Grunde genommen startet der Tastaturclock in dem Moment in dem man eine Taste drückt. Kurze Verzögerungen innerhalb des Controllers mal ausser acht gelassen.

Diesen Start kann man unterbinden und hinauszögern, indem man dies per Busy-Signal meldet. Wenn der Start aber erstmal erfolgt ist, würde das Melden des Busy-Signals aber nur den Abbruch der aktuellen Übertragung bewirken. Einigen Tastaturen soll das Setzen des Busy-Signals sogar egal sein, sie senden dann einfach weiter. Die Folge wäre ein Datenverlust.

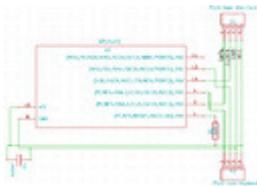
Ich habe mal versucht, meiner Tastatur ein Clock-Signal durch den Host generiert aufzuzwingen. Am Ende kam dabei aber nur Datenschrott zu stande, weil die Tastatur immer wieder ihre Übertragung abgerochen und neu gestartet hat und ich somit immer wieder die ersten Bits der 1. gedrückten Taste zu sehen bekam.

Grüße  
Frank

Autor: Frank Zoll ([fzoll](#))

Datum: 15.10.2010 20:13

Angehängte Dateien:



• [ps2\\_adapter.png](#)

10 KB, 260 Downloads

---

Halloe.

Ich spiele gerade mit dem Gedanken, das PS/2 Keyboardproblem durch den Einsatz eines weiteren Controllers zu erschlagen. Anbei ein kleines Schema von dem was ich mir vorstelle. Ich denke darüber nach, einen kleinen 8Pinner [AVR](#) Controller in die Leitung zwischen der PS/2 Buchse auf der Platine und der Tastatur zu einzubringen. Ich würde die Clockleitungen beider Seiten auf Interrupt- Eingänge des [AVR](#)-Controllers legen und die 2 Datenleitungen auf die übrigen Pins verteilen.

Nun könnte man den [AVR](#)- Controller so Programmieren, das er auf der einen Seite Datenbytes der Tastatur mit der geschwindigkeit, wie sie durch die Tastatur festgelegt wird empfangen kann. Diese Bytes könnte der Controller dann in seinem SRAM zwischenspeichern, bis sie durch den VGA- Controller auf unserer Platine abgeholt werden. Hierzu würde ich den VGA- Controller ebenfalls einen Tackt ausgeben lassen. Das Tacktsignal könnten wir dann relativ problemlos in der HSYNC- Phase erzeugen lassen. Wir könnten auf geraden Ausgabezeilen einen Tackimpuls erzeugen und auf ungeraden Zeilen die Daten entgegen nehmen. So könnten wir dann am Ende die Datenbytes stück für stück vom "Puffercontroller" abholen. Ein weiterer Vorteil wäre, das der Puffercontroller je nach größe bis zu 512 Zeichen Puffer bereit stellen könnte. Im moment denke ich an einen [ATtiny45](#) für knapp 1,82€, welcher uns einen 256 Byte Puffer schenken würde und leicht zu beschaffen ist.

Was ich daran nett finde ist, das wir dafür an der Platine keine Änderungen vornehmen müssten. Und auch der Geldliche Aufwand wäre ja minimal. (1 Stecker, 1 Buchse, der ATtiny Controller, ein Kerko und ein Widerstand). Das ganze sollte sich recht einfach "fliegend" aufbauen lassen.

Was haltet ihr von der Idee, wäre das eine nette Lösung für das Keyboard- Problem ?

Freundliche Grüße  
Frank

**Autor: Egmont (Gast)**  
**Datum: 15.10.2010 22:27**

---

Hallo Frank

[Egmont schrieb:](#)

> gibt es die Möglichkeit ein Bildaufbau auszulassen ??

Hier noch mal eine genaue Erklärung was ich damit meinte...

bei einer Vertical Frequency von ca. 70Hz kann man da nicht ein komplette Bildaufbau auslassen. Mit auslassen meine ich keine Daten ausgeben nur Hsync und Vsync.

Ein Bildaufbau dauert 12 ms (Active video time)soviel Zeit hat man die Tastatur auszulesen.

- der Tastatur durch setzen von Clock auf High signalisieren das man bereit ist Daten aufzunehmen.
- Daten einlesen (Tastatur kann 16 Byte im eigenem Buffer aufnehmen)
- Clock auf Low setzen.
- Bilddaten 69 mal ausgeben... und so weiter.

Ein Bildaufbau aussetzen (dunkel) sollte man eigentlich nicht merken.

Was meinst du, ist das machbar !!?

Gruß Egmont

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 15.10.2010 22:56**

---

Hallo Egmont.

Einen Bildaufbau bei Bedarf aus zu lassen wäre wohl Problemlos möglich. So lange die SYNC- Signale trotzdem erzeugt werden, sollte der Monitor nicht aus dem Tackt kommen. Damit hätte man eine Menge freier Prozessorzeit über, um Daten von der Tastatur einzulesen. Das ganze wird wohl ein bisschen Tricky, ich könnte mir aber vorstellen das das machbar wäre.

Freundliche Grüße  
Frank

**Autor: Hans- w. Schütz ([hschuetz](#))**  
**Datum: 15.10.2010 23:08**

---

@fzoll

eine Tastaturlösung mit eigenem Kontroller hab ich fertig hier liegen.. nur der Reiz wäre schon Tastatur und VGA auf einem Chip...

Allerding sollte es nicht funktionieren...

gruß

Hans- Werner

**Autor: Marc (Gast)**  
**Datum: 16.10.2010 10:06**

---

Wäre folgendes für den VGA-Kontroller möglich:

Tastatur über seriellen Port anschließen. Die Verbindung zum CPM-Prozessor mit zwei freien Portpins herstellen und eigenes, I2C

ähnliches ( synchrones Protokoll ) implementieren.

**Autor: Peter Sieg (Gast)**

**Datum: 16.10.2010 10:30**

---

Noch ein Vorschlag (habt ihr aber sicher auch schon dran gedacht..)  
Wenn der VGA-AVR das nicht mehr sauber packt.. dann den CP/M [AVR](#) nehmen  
und  
die 2 KBD Leitungen halt umlegen..

Peter

**Autor: Joe G. ([feinmechaniker](#))** 

**Datum: 16.10.2010 14:12**

---

[Frank Zoll schrieb:](#)

> Was haltet ihr von der Idee?

Wie im richtigen Leben gibt es für jede Lösung Pro und Contra. Ein ganz wichtiges Argument: „Die schöne Platine nicht kaputt zu machen“ Da geht eigentlich nichts drüber!

Doch nun weg von den emotionalen und hin zu den sachlichen Argumenten. Soll die VGA Karte als eigenständiges „Terminalmodul“ fungieren, so ist eine einfache universelle Schnittstelle immer von Vorteil. Das ist mit der derzeitigen V24 gegeben. Würde die Tastatur über die beiden I2C Leitungen der [AVR](#) CP/M Platine abgefragt, so wäre das ein Systembruch im Sinne eines eigenständigen Terminals. Die Lösung von Frank einen [Attiny45](#) zu nehmen hat auch ihren Charme. Damit könnten sogar solche Kunstwerke wie die Logitech diNovo Mini™ adaptiert werden.

Joe

**Autor: Frank Zoll ([fzoll](#))**

**Datum: 16.10.2010 20:38**

**Angehängte Dateien:**

- [avr\\_cpm\\_vga.zip](#) (71,9 KB, 80 Downloads)

---

Halloe.

Ich habe Egmont's Idee einmal zum Testen umgesetzt. VGA und Tastatur lassen sich nun zusammen mit dem Terminal betreiben. Und das ganz ohne Änderungen an der Platine oder zusätzliche Hardware.

Um das ganze recht einfach zu halten, habe ich es so umgesetzt, das jedes 32. Bild ausgelassen wird, um die Tastatur zum Zug kommen zu lassen. Auf meinem 17Zoll-Monitor ist das Auslassen eines Bildes deutlich als Flackern erkennbar. Ob ich damit Stundenlang arbeiten möchte, lasse ich einfach mal dahin gestellt.

Mal schauen, ob man es noch ein wenig weiter Verbessern kann. Anbei mal die aktuelle Version für euch zum Rumspielen.

Freundliche Grüße

Frank

p.s. Ich habe längst noch nicht alle VT100 - Kommandos umgesetzt, mir war erstmal wichtiger VGA und Tastatur ans laufen zu bekommen.

Autor: Frank Zoll ([fzoll](#))  
Datum: 17.10.2010 09:50

---

Halloe.

Da die Idee mit dem Auslassen eines ganzen Bildes recht gut funktioniert hat, habe ich mal versucht das ganze so umzustricken, das es komplett während der VSYNC- Zeit läuft. Dies hat mit meiner Tastatur leider nicht funktioniert. Es kam kein einziges Byte mehr rüber. Danach habe ich mal versucht, die Zeit der letzten 16 Zeilen, die eh nicht ausgegeben werden, mit dazu zu nehmen. Nun konnte ich zwar ein paar Zeichen empfangen, hatte aber immer wieder Übertragungsfehler. Mal schauen, was sich da noch machen läßt.

Habt ihr vielleicht noch Ideen ?

Ach ja, einen seltsamen Fehler habe ich auch noch beobachten können. Mit dem "BDOS" zusammen funktioniert es eigentlich super. Aber wenn man Programme lädt, gehts nicht mehr richtig. Konkret die RETURN- Taste scheint nicht mehr abgefragt zu werden. Bei ZORK und MBASIC kann man zwar munter etwas eintippen, aber wenn man RETURN drückt passiert nichts. Ich vermute mal, das es noch Fehler in der VT100 Emulation sind, die wir noch beseitigen müssen.

Grüße  
Frank

Autor: Egmont (Gast)  
Datum: 17.10.2010 12:23

---

Hallo Frank

[Frank Zoll schrieb:](#)

> Habt ihr vielleicht noch Ideen ?

Wenn in den letzten 16 Zeilen nichts mehr ausgegeben wird, sollte man da schon KB-Clock auf High setzen, wenn jetzt die Tastatur antwortet, also Daten hat, ein Bildaufbau komplett auslassen. Wenn in der Zeit nichts kommt, Bildaufbau normal ausgeben. Das heißt das Bild "flackert" nur wenn man eine Taste drückt.

16 Zeilen \* 25µs = 400µs -->  
Clock muss für min 50µs High sein bevor Tastatur sendet. Also schafft es die Tastatur in der Zeit von 350µs zu signalisieren das sie Daten hat. Das bedeutet auch das man 70 mal in der Sekunde versucht die Tastatur abzufragen ( ob was im KB-Buffer ist) ....  
ist besser als....  
2 mal in der Sekunde abzufragen und ein Bildaufbau auslassen (und dann hat der User noch nicht mal eine Taste gedrückt.... dumm gelaufen).

Egmont

Autor: Thomas Hommel (Gast)  
Datum: 17.10.2010 17:07

---

Hallo Frank,

Frank Zoll schrieb:  
> Habt ihr vielleicht noch Ideen ?

Alles in der VSYNC-Zeit zu schaffen, wäre schon ideal. Wäre es nicht denkbar, vom Modus 640x400@70Hz auf 640x480@60Hz zu gehen? Horizontal bleibt alles gleich, es kommen nur 80 schwarze Pixelzeilen dazu. Grob überschlagen hätte man dann pro Bildaufbau etwa 4.3ms Zeit für Keyboardroutinen, während es bei 640x400 nur ca. 2ms sind. Ich kann mir vorstellen, dass die derzeit 2ms zu wenig sind, um eine Tastaturantwort entgegenzunehmen. Teilweise sind die Make/Breakcodes auch 2 oder 3 bytes lang, welche man von der Tastatur ohne Unterbrechung entgegennehmen muß.

Viele Grüße, Thomas

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 19.10.2010 20:07**

---

Hallo Thomas.

Ich habe deinen Vorschlag ausprobiert. In der aktuellen [SVN](#)- Version ist die Auflösung nun 640\*480 Pixel bei 60Hz (62Hz zeigt meine Monitor an).

Die Keyboardroutinen laufen nun vollständig während der Zeit in der keine aktive Videoausgabe erfolgt. Von 480 Zeilen werden nur 384 Zeilen ausgegeben. Den rest der Zeit hat nun die Keyboardroutine um Zeichen zu empfangen. Damit konnten wir das Auslassen ganzer Bilder nun erfolgreich "wegoptimieren".

Das Bild bleibt nun recht stabil. Flakern tut's im Moment nur beim Scrollen, da werde ich als nächstes mal drauf schauen.

Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 21.10.2010 20:03**

---

Hallo Egmont.

Beide Versionen 640\*400 und 640\*480 habe ich auf einem alten Röhrenmonitor getestet. Der iiyama Visionmaster Pro410 hatte keine Probleme mit der Anzeige, obwohl der Pixeltakt nicht genau genug ist. (62 statt 60 bzw. 74 statt 70 Hz).

Ich habe die neue Version mal auf einem "neueren" TFT Monitor ausprobiert, um zu schauen ob evtl. neuere Monitore mit der krummen Auflösung Schwierigkeiten haben. Mein Samsung SyncMasterT220HD (BreitbildMonitor) hat keine Probleme, die Auflösung 640\*480 @ 62Hz wieder zu geben. Der Font mit 6(8)\*16 Pixeln sieht sogar recht nett aus, weil der Monitor das Bild netterweise in der Breite für mich gestreckt hat.

Wie siehts den sonst so bei den anderen aus? Habt ihr evtl. auch Monitore, die es stört das der Pixeltakt nicht genau genug ist für 640\*480@60Hz?

Grüße  
Frank

Autor: Egmont (Gast)  
Datum: 22.10.2010 00:01

---

Hallo

mit folgenden Werte habe ich ein besseres Bild bekommen  
und die Info sagt 640x480@60Hz:

```
----[ global.h ]-----  
#define H_FRONT      8  //2    // front porch  
#define H_SYNC       10 //12   // sync pulse -- NEGATIVE polarity  
#define H_BACK        6      // back porch  
#define H_ACTIVE     80    // active video  
#define H_TOTAL      (H_SYNC+H_BACK+H_ACTIVE+H_FRONT)  
#define H_CHARS      H_ACTIVE  
// unit: LINES  
#define V_FRONT      5  //11   // front porch  
#define V_SYNC        2    // sync pulse -- POSITIVE polarity  
#define V_BACK       31   // back porch  
#define V_ACTIVE     480  // active video  
#define V_TOTAL      (V_SYNC+V_BACK+V_ACTIVE+V_FRONT)  
#define V_CHARS      24
```

Dazu habe ich festgestellt,  
wenn man wie ich beide Atmel mit 25Mhz betreibt,  
ein besseres Bild bekommt wenn man nur ein Quarz benutzt  
(Jumper JP10 auf 1-2).

Gruss Egmont

Autor: Joe G. ([feinmechaniker](#))   
Datum: 22.10.2010 14:36

---

[Egmont schrieb:](#)

```
> Dazu habe ich festgestellt,  
> wenn man wie ich beide Atmel mit 25Mhz betreibt,  
> ein besseres Bild bekommt wenn man nur ein Quarz benutzt  
> (Jumper JP10 auf 1-2).
```

Dieser Effekt ist dem Entwicklungsboard geschuldet. Die [AVR](#)'s sind nicht besonders "EMV freundlich" In einem Industrieprodukt würde meine Stromversorgungsvariante der beiden [AVR](#)'s durchfallen. Ich hatte bewußt auf die Drosseln verzichtet. So schlägt der Takt recht stark auf die Stromversorgung durch. Bei nur einer Taktversorgung wenigstens nur mit der einen Frequenz. Asche auf mein Haupt! Das nächste Board bekommt ein EMV gerechteres Design.

Autor: Frank Zoll ([fzoll](#))  
Datum: 28.10.2010 00:45

---

Halloe.

Ich habe mal wieder ein kleines Update der Software für den VGA/Keyboard- Teil des aktuellen Boards auf den [SVN](#)- Server hochgeladen. Mit dem aktuellen Update sind nun die ersten VT100/VT52 Funktionen implementiert. Die Tasten RETURN und BACKSPACE tun nun was sie sollten. Damit läßt sich nun auch endliche eine Partie "ZORK1" auf dem Board spielen. Der Cursor wird nun auch endlich korrekt angezeigt. Die Tastaturroutinen laufen nun, ohne das ganze Bilder ausgelassen werden müssen. Durch auslassen der Darstellung der letzten 96 Bildzeilen

konnten die Tasterabfragen ans laufen gebracht werden. Das ganze ist etwas "schwammig" und sicher nicht für Actionspiele geeignet, aber immerhin tut's jetzt einigermaßen.

Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**  
**Datum: 04.11.2010 20:41**

---

Hallo Hans Werner.

Sorry, im moment komme ich hier zu fast gar nichts mehr. Daher hier erst verspätet meine Antwort.

Ich nutzte unter WindowsXP den [SVN](#)- Client Tortoise. Mit dem kann man eigentlich Problemlos auf den [SVN](#)- Server zugreifen und Updates ziehen geht dort ganz einfach über einen "rechtsklick" auf das Verzeichniss und die Auswahl des Punktes "SVN Update".

Ich kann die Zwischenstände auch hier im Thread Posten, habe es aber bisher vermieden, um den Server nicht unnötig mit Alpha & Betha-Versionen vollzumüllen. Mein Vorschlag wäre, erstmal noch bei dem [SVN](#)-Server zu bleiben und erst die fertige Version des VGA- Terminals hier für alle nochmal zu posten. Ich habe auch keine Probleme euch gezippte Dateien auf Anfrage per E-Mail zukommen zu lassen. Was meint ihr ?

Was aber das Problem mit dem Compilieren angeht, so habe ich es gerade nochmal ausprobiert. Die aktuelle [SVN](#)- Version läßt sich ohne Fehler übersetzen. Ich denke das Problem das Du da hast, ist evtl. das gleiche das ich auch beim ersten Compilieren der "Urversion" hatte. Ich hatte eine etwas ältere Version von [AVR-GCC](#) installiert. Diese Version kannte aber den 328er [AVR](#) noch nicht und hatte daher nicht die passenden Includes mit den Daten der Anschlußpins. Nach einem Update auf die Version "WinAVR-20100110" hat es dann bei mir aber funktioniert. Schau am besten einmal nach, welche Version dort installiert ist.

Zum "aktuellen" Stand kann ich sagen, das ich als nächstes vor habe die VT100 Emulation weiter aus zu bauen. Im moment habe ich hier lokal bereits eine Version mit fertiger aber noch ungetesteter VT52 Emulation. Meine Planung sieht vor, die Emulation zwischen VT52 und VT100/102 per F12- Taste umschaltbar zu machen. Im moment komme ich nur nich zum Programmieren.

Die VT52 Emulation ließ sich viel leichter Umsetzen als die VT100/102 Emulation, weil die Steuersequenzen deutlich kürzer und viel einfacher aufgebaut sind. Bei VT52 kann immer nur ein Commando mit fester Parameteranzahl nach dem ESCAPE kommen. Bei VT100/102 kann man praktisch beliebig viele "Variablen" nach dem ESCAPE senden und dann ganz am Ende sagen, welcher Befehl mit diesen Parametern auszuführen ist. Außerdem werden die Variablen bei VT52 praktisch "binär" als einzelnes ASCII- Zeichen gesendet, von dem man dann nur noch eine Constante abziehen muss um auf den richtigen Wert zu kommen. Bei VT100/102 kommen die Variablen als String und müssen erst noch interpretiert werden. Das ganze wird mich wohl noch eine ganze Zeit lang beschäftigen, bis es einigermaßen zufriedenstellend läuft.

Freundliche Grüße  
Frank

**Autor: Frank Zoll ([fzoll](#))**

Datum: 21.11.2010 21:56

---

In den letzten Tagen habe ich die Terminal- Emulation stark erweitert. Die aktuelle Version findet sich wie immer auf dem oben verlinkten [SVN-Server](http://cloudbase.homelinux.net/svn/avr-cpm). ( <http://cloudbase.homelinux.net/svn/avr-cpm> )

Unter anderem neu sind:

- Eine ADM-3A Emulation steht nun auch zur Auswahl.
- Umschaltung der Emualtionen VT52/VT100/ADM3-A per F11
- Umschaltung der Schriftfarbe per F12
- Fast alle ESCAPE- Codes des ADM-3A Terminals wurden implementiert. ( Ausnahmen: Auflösungswechsel, Testmodus und Baudratenänderung )
- Unterstützt die Font- Umschaltung auf einen 2. Font.  
( Derzeit wird hier nur zwischen 2 sehr ähnlichen Fonts umgeschaltet)

Was fehlt noch ?

- Jemand der Lust hat alle ESCAPE- Sequenzen des ADM-3A Terminals auszuprobieren und ggf. Fehler zu melden. Schön wäre ein MBASIC- Programm, mit dem man alle Codes ausprobieren kann, damit der Fehler leichter nachzustellen ist.
- Den beiden Fonts fehlen noch die Grafikzeichen, derzeit werden Graphiczeichen als Leerräume dargestellt. Mag hier wer die fehlenden Zeichen hinzu fügen ?
- Die VT52 und die VT100 ESCAPE- Sequenzen werden noch nicht vollständig beherrscht.

Freundliche Grüße  
Frank

**Autor: fzoll (Gast)**

**Datum: 25.11.2010 11:18**

---

Halloe.

Ich glaube ich konnte das Problem der verlorenen Zeichen vom CP/M Kern an das Terminal weiter einkreisen. Mir ist das ganze bisher im Test noch nicht aufgefallen. Das könnte evtl. daran liegen, das ich den CP/M Teil nur mit 20Mhz laufen lasse. Solltest Du den CP/M Kern übertacktet haben, so wäre der ja etwas schneller als meine Version, so das das unten Beschriebene Problem dann mit einer größeren Warscheinlichkeit auftreten könnte. Ich versuchs auf jeden Fall mal nachzustellen.

Hier meine Vermutung:

Es könnte daran liegen, das während der V-SYNC Phase keine Zeichen von der Seriellen Schnittstelle zum CP/M System abgeholt werden. Die V-SYNC-Phase ist jedoch ein klein wenig länger, als die Dauer der Übertragung eines einzelnen Zeichens. Das könnte dazu führen, das einzelne Zeichen verloren gehen, wenn sie genau in der VSYNC- Phase gesendet wurden. Ich hoffe ich habe mich da nicht verrechnet :-).

Ich werde heut Abend mal versuchen ob es evtl. besser wird, wenn auch während des VSYNC's auf eingehende Zeichen reagiert wird.

Was die nicht erkannten Tastendrucke angeht, so kann man da wohl nicht mehr viel ausrichten. Da ich nur während des VSYNC's mit der Tastatur interagieren kann, konnt es stark auf den Typ der Tastatur und die Tippgeschwindigkeit an. Während der Ausgabe der sichtbaren Bildteile muss ich die Tastatur ständig dazu zwingen ihre Kommunikation einzustellen und ggf. sogar abubrechen. Ich denke mal das das ganze einige der Tastaturen am Markt nicht all zu gern mitmachen. Meine eigene Tastatur gehört auch dazu. Sobald ich mal ein wenig schneller Tippe

gehen auch mir Tastendrücke verloren.

Eine mögliche Lösung hierfür sieht ein kleines ATINY- Puffer IC in der Tastaturleitung vor, das die Zeichen zwischenspeichert und dann von der VGA- Seite her per Pollingverfahren ausgelesen wird. Derzeit habe ich nur leider keinen ATINY hier rum liegen, so das ich das im moment nicht Ausprobieren könnte. Und das bestellen eines einzelnen ATINY's lohnt sich nicht. Sobald ich da etwas fertiges habe, werde ich das als option mit in den VGA- Kern übernehmen. Dann kann jeder nacher beim Compilieren per DEFINE's entscheiden, ob er mit oder ohne zusätzliches PufferIC arbeiten möchte.

Was mir noch aufgefallen ist, ist das auch das ADM-3A Terminal spezielle Steuerzeichen für Befehle wie Unterstrichen, Unsichtbar und Blinkend kennt. Diese kann ich leider mangels Ressourcen auf dem [AVR](#)-VGA Kern nicht unterstützen, sie werden daher derzeit auch noch nicht einmal erkannt. Hier baue ich noch eine Erkennungsroutine ein, die die Sequenzen erkennt und dann aber einfach ignoriert. Damit sollten die Steuerzeichen nicht mehr mit auf dem Bildschirm ausgegeben werden.

Vielen Dank fürs Testen,  
Frank

Autor: Peter Sieg (Gast)  
Datum: 08.01.2011 16:31

---

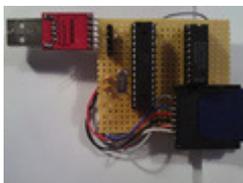
Hi. Ich bin immer noch bei der 4-bit Version und der alten Version mit tniasm...

Was ist eigentlich aus der Z80 Emulation geworden..?

Peter

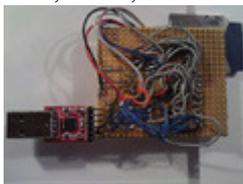
## Bilder Lochraster 4-bit + USB Dongle

Autor: Peter Sieg (Gast)  
Datum: 24.03.2011 18:22  
Angehängte Dateien:



[DSC00735.JPG](#)

126,8 KB, 269 Downloads



[DSC00734.JPG](#)

144,8 KB, 248 Downloads

---

Hier Bilder meines LR Aufbaus mit bei [ebay](#) erhältlichen USB<->Seriell Wandlern, die auch gleich +3,3V zur Verfügung stellen..

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 25.03.2011 09:08

---

Ja, irgendwie schade. Die Z80 Emulation hätte die Softwaremöglichkeiten noch gewaltig vergrößert.

Gruß Joe

Autor: Peter Sieg (Gast)

Datum: 25.03.2011 09:32

---

Hi Hans-Werner und Joe.

Das war der aller erste LR Aufbau, der zum aller ersten Post in diesem Thread führte. Damals noch mit 3.3V Netzteil und Handykabel zur seriellen Verbindung.. ;-)) Als ich den Adapter sah, mußte ich den einfach bestellen (3-4€) und wußte, das der hier wie die Faust aufs Auge passt..

Ist immer noch ein 4-bit Aufbau.. ich mag diese SMD Drams nicht :-(

Die 8080 Emulation und CP/M laufen sehr gut!

Wie Joe aber schon sagte habe ich von der Erweiterung auf Z80 Emulation nichts mehr gehört und das serielle Terminal ist wohl auch nur rudimentär fertig gestellt..

Die letzten Entwicklungen gingen mir zu schnell um noch mitzukommen und Richtung NUR Linux wollte ich auch nicht.. daher nutze ich immer noch das 'alte' Diskformat und den tniasm..

Evtl. baue ich die 8-bit Dramversion mal in DIL auf LR auf.. aber mit Platine wäre das schon einfacher..

Wie genau der letzte Stand ist und wo der nun liegt..??

---

Wenn man das mal wieder beleben wollte, würde ich (just my 2 cents!) eine neue Platine machen wollen: 8-bit aber DIL - kein SMD! Ohne Terminal, dafür aber obigen Adapter vorsehen, der gleich die 3,3V Versorgung und die PC Kommunikation über USB erledigt. Also quasi 1x ATmega328, 2xDIL Dram, Quarz+2xC, USB Adapter. Thats it.

-

Die darauf laufende Software auf den aktuellen Stand bringen und dokumentieren. Hier insbesondere das/die neuen Diskformate berücksichtigen.

-

Die Z80 Emulation voran bringen!

Peter

Autor: Peter Sieg (Gast)

Datum: 25.03.2011 09:36

---

Zu dem USB-Adapter:

<http://cgi.ebay.de/ws/eBayISAPI.dll?ViewItem&item=...>

USB To TTL / COM Converter Module buildin-in CP2102

Aktuell: 3,51€ inkl. Versand

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 25.03.2011 11:13

---

Peter Sieg schrieb:

> Wenn man das mal wieder beleben wollte, würde ich (just my 2 cents!)  
> eine neue Platine machen wollen: 8-bit aber DIL - kein SMD!

Das sollte sich machen lassen... wenn es wieder auf Interesse stößt.

> Die darauf laufende Software auf den aktuellen Stand bringen und  
> dokumentieren. Hier insbesondere das/die neuen Diskformate  
> berücksichtigen.

Hier sehe ich den größten Aufholbedarf. Das Projekt wird erst von Interessenten nachgenutzt, wenn die Doku auch eine Nachnutzung erlaubt.

> Die Z80 Emulation voran bringen!

JA

## Das Projekt schläft etwas ein

Autor: Peter Sieg (Gast)

Datum: 25.03.2011 15:17

---

Ich habe mal im Forum64, Forum des Vereins zum Erhalt klassischer Computer und Robotron Forum Werbung gemacht:

---

Hi. Mehr Info's hier:

[Beitrag "CP/M auf ATmega88"](#)  
[http://avr.cwsurf.de/?AVR\\_CP%2FM](http://avr.cwsurf.de/?AVR_CP%2FM)

Das Projekt ist etwas 'eingeschlafen', aber wir möchten es gerne etwas wiederbeleben.

Es soll - bei genügend Interesse - eine neue Platine geben nur in DIL! (kein SMD). Das wird dann ein ATmega328 und 2 x 4-bit Dram werden. Dazu noch ein SD Slot und den USB-TTL Wandler und fertig.

Was zur Zeit alles läuft: 8080 Emulation inkl. Microsoft Basic, BDS C-Compiler, Spiele etc. inkl. vollem CP/M System mit bis zu 4 Diskettenimages..

Was noch fehlt: Z80 Emulation

Falls ihr an einer Platine interesse habt und bei der Doku oder Programmentwicklung weiterarbeiten möchtet, postet es bitte im obigen Thread und hier.

Das Ding macht wirklich Spaß!

Peter

---

Mal sehen, ob wir mind. 10 Interessenten zusammenbekommen. Bei eine einfachen DIL Variante mit [AVR](#), 2xDram für 8-bit und dem oben genannten USB-TTL Wandler bin ich schon mit 2-3 Platinen dabei.

Peter

Autor: Frank P. ([mauz](#))

Datum: 30.03.2011 21:14

---

Hi Leute,

eins vornweg ich finde das Projekt herrlich bekloppt und werde erstmal ein Lochrasteraufbau fädeln. Da kommt der Tipp mit dem Modul von Peter wie gerufen, der FT ist ja nicht so gut Lochraster geeignet (ist ja nicht jeder Elm Chan :-). Da tut sich aber gleich eine Frage auf: Die erhältlichen Module scheinen alle mit CP2102 bestückt zu sein. Weiß jemand, wie gut die mit Linux laufen? Für eine professionell gefertigte Platine könnte ich mir noch eine "alles in SMD, so eng wie es handzubestücken geht, wenn irgendmöglich in USB-Flashstick-Format gequetscht"-Version vorstellen ;-)

frank

Autor: Nils Eilers ([yetanotheruser](#))

Datum: 30.03.2011 21:25

---

Ich finde die Idee mit der Stiftleiste auch gut, würde aber auf der Platine dann noch den [FT232](#) mit drauf nehmen - der braucht nicht viel Platz und wenn jemand was anderes will, kann er das über die Stiftleiste anschliessen ohne Leiterbahnen zu durchtrennen etc.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 31.03.2011 08:58

---

Es geht ja nicht darum prinzipiell eine Platine für das Projekt auf die Beine zu stellen, das haben wir ja schon in zwei Varianten getan, siehe hier:

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

Die Idee ist vielmehr mit einem ganz, ganz einfachen Aufbau Mitstreiter zu finden, die auch an der Software noch etwas tun. Stichwort Z80. Einfach soll auch heißen DIL auf Fassung um auch mal ein IC zu tauschen oder zu testen.

Joe

Autor: Peter Sieg (Gast)

Datum: 31.03.2011 11:30

---

Genau. Im Prinzip sehe ich die 4-bit Platine aus dem Artikel erweitert/umgebaut auf 8-bit (andere Portbelegungen) und der Pinheaderbereich links über dem SD Slot dann kompatibel zu dem CP2102 USB-TTL Adapter ausgeprägt als die hier jetzt angestrebte neue Platine. [FT232](#)+USB Buchse ist ja parallel noch drauf..

Der Vorteil der 8-bit Version ist halt die fast verdoppelte Geschwindigkeit der Emulation.. womit insbesondere Arcade-like Spiele (Ladder/Pacman) besser spielbar werden..

Und ja, schön wäre die Erweiterung auf Z80 Emulation, die dann die Tür öffnen würde für z.B Turbo Pascal, Sargon Schach, etc..

BTW: In dem Artikel ist die letzte 8-bit Version mit SMD Rams und VGA-Terminal Teil übrigens nicht dokumentiert..?

Peter

Autor: Frank P. ([mauz](#))

Datum: 31.03.2011 16:14

---

[getestet schrieb:](#)

> Ist kein Problem, Treiber ist seit langem im Kernel.  
Danke fürs testen ;-)

[Joe G. schrieb:](#)

> Es geht ja nicht darum prinzipiell eine Platine für das Projekt auf die  
> Beine zu stellen, das haben wir ja schon in zwei Varianten getan, ...  
Ich denke die Ansprache gilt meiner "so-klein-wie-möglich-Idee". Ich weiß, was Ihr schon alles auf die Beine gestellt habt. Dafür genießt Ihr meinen vollen Respekt. War ja auch nur so eine Idee, falls mal jemandem langweilig wird ;-)

> Die Idee ist vielmehr mit einem ganz, ganz einfachen Aufbau Mitstreiter  
> zu finden, die auch an der Software noch etwas tun. Stichwort Z80.  
> Einfach soll auch heißen DIL auf Fassung um auch mal ein IC zu tauschen  
> oder zu testen.

Kann ich nachvollziehen, und ich will Euch da auch nicht reinreden, Gott bewahre. Zumal ich Euch bei der Software wahrscheinlich nicht viel helfen kann. **Ich** finde für **mich** eine professionell gefertigte Platine für alles in DIL etwas übertrieben. Nichtsdestotrotz, wenn es neue Mitstreiter bringt und Ihr noch eine Mindeststückzahl braucht, würde ich auch eine LP nehmen. Es sollte dann aber schon eine 8-Bit Version sein. An Z80 wäre ich natürlich auch sehr interessiert.  
frank

Autor: Peter Sieg (Gast)

Datum: 04.04.2011 22:55

---

@alle: was ist denn eigentlich `der letzte Stand` der Software?

Im [SVN](#):

<http://cloudbase.homelinux.net/svn/avr-cpm/>

steht Revision 161.

Unter avrcpm - branches gibt es 2\ und fat16-test\

fat16-test war das lesen von Diskimages von fat16 formatierten SD Karten..  
..wie ist da der Status.. wer hats getestet..?

Und 2\ = Version 2? Was war doch gleich noch mal der Unterschied zu der ersten Version..?

Und dann gibts noch trunk\ ..?

Nun ich habe hier die Version 1, die für 4-bit und 8-bit assembliert werden kann.. und bis zu 4 CP/M Partitionen unterstützt.. damit werde ich ersteinmal die neuen 8-bit Platinen versorgen, solange mich niemand eines anderen überzeugt.. ;-)

Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 06.04.2011 12:22

---

[Peter Sieg schrieb:](#)

> @alle: was ist denn eigentlich `der letzte Stand` der Software?

Die 2.0 Version lief auf meiner 8-Bit Hardware. Problem dort war, die SD Karten konnten nur noch unter Linux mit den CP/M Partitionen erzeugt werden.

Für Win.. User nicht so schön.

Die fat16 Version konnte ich zwar übersetzten jedoch nicht von der SD Karte booten. Es war für mich nicht so richtig transparent, wie und in welchem Format das CP/M System auf die Karte kam. Für weitere Entwicklungen fürde ich diesen Zweig lieber weiterverfolgen.

Joe

Autor: Peter Sieg (Gast)

Datum: 09.04.2011 08:17

Angehängte Dateien:

- [fat16.zip](#) (145,1 KB, 34 Downloads)

---

Habe mir die fat16-test Version mal gezogen und für 4-bit; m168; 30MHz; fat16 Nutzung assembliert. Dann disk images als CPMSK\_A.IMG .. CPMSK\_D.IMG auf eine FAT16 formatierte SD Karte kopiert und es läuft!

8-bit kann ich mangels Hardware nicht testen. [ATmega88](#) scheint Fehler beim assemblieren zu geben (Funktionen werden genutzt, die erst ab 168 da sind).

Als disk images habe ich die originalen verwendet (ursprüngliches Format von ca. 243kb).

Sieht doch schon mal gut aus! Das komplette fat16 Verzeichnis inkl. [AVR Studio 4](#) Projektdatei und erzeugtem Hex hänge ich hier mal an.

Peter

# Bios Assemblieren

Autor: Andreas H. ([andreas\\_h35](#))

Datum: 09.04.2011 08:32

---

In froher Erwartung auf die neue [AVR](#)-cpm Leiterplatte habe ich mir das CPM-Image auch schon eingehender angeschaut, konnte aber nirgends die Quellen für das CPM.SYS finden. Wer hat das CPM.SYS generiert und kann dazu auch die Quellen abgeben ?!

Danke und Grüsse,  
Andreas

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 09.04.2011 09:59

---

Peter Sieg schrieb:

> Sieht doch schon mal gut aus!

Bestens Peter, werde ich heute oder morgen mal auf 8-Bit testen. Im nächsten Schritt sollten wir die Doku "Nachbausicher" machen. Auch im Hinblick auf solche Fragen wie von Andreas.

@Andreas

Das cpm.sys ist die Originaldatei, zu finden hier:

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

Den Quelltext dazu kenne ich auch nicht. Um das Bios zu übersetzen und die cpm.bin Datei zu erzeugen hatten wir auch für Win Makefiles erstellt. Ich suche mal alles zusammen und bemühe mich eine kleine Kurzanleitung dazu zu verfassen.

Joe

Autor: Peter Sieg (Gast)

Datum: 09.04.2011 12:28

Angehängte Dateien:

- [bios.zip](#) (72,9 KB, 26 Downloads)

---

Im Anhang mein bios Verzeichnis.

(Den Quelltext von cpm.sys habe ich aber auch nicht)

Darin ist bios.asm+ipl.asm

tniasm.exe zum assemblieren

make\_bios.bat zum übersetzen von ipl+bios

make\_cpm.bat zum bilden von cpm.bin (nutzt dd.exe, was auch enthalten ist)

Das Resultat cpm.bin wird dann zum bilden von diskimages mit den cpmttools

verwendet:

```
mkfs.cpm.exe -f avrcpm -b cpm.bin -L test CPMSK_A.IMG
```

und füllen mit:

```
cpmcp -f avrcpm CPMSK_A.IMG cpmsk/MBASIC.COM 0:MBASIC.COM
```

Peter

Autor: Andreas H. ([andreas\\_h35](#))

Datum: 09.04.2011 12:44

---

Wegen BDOS+CCP bin ich fündig geworden. Die Quellen, für Z80 angepasst, konnte ich bei Peter Schorn (<http://www.schorn.ch/cpm/intro.php>) finden.  
Gruss, Andreas

Autor: Peter Sieg (Gast)

Datum: 09.04.2011 17:29

---

@Andreas: Wäre super, wenn du die benötigten Teile zusammen puzzeln würdest, um cpm.sys zu bilden. Idealerweise auch mit einem schon verwendeten Z80 Assembler (tniasm; z80asm).

Peter

Autor: Andreas H. ([andreas\\_h35](#))

Datum: 09.04.2011 20:21

Angehängte Dateien:

- [cpmsurvey.zip](#) (6,6 KB, 31 Downloads)

---

Hallo Peter,

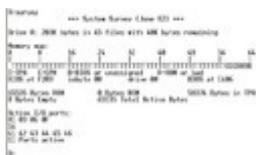
kannst Du mir bei Gelegenheit das SURVEY.COM (im Archiv) auf dem AVRcpm ausführen ?! und den Output schicken. Die wichtigen Adressen stehen zwar im BIOS.ASM, aber sicher ist sicher.

Danke, Andreas

Autor: Peter Sieg (Gast)

Datum: 10.04.2011 12:43

Angehängte Dateien:



- [survey.JPG](#)  
46,7 KB, 239 Downloads

---

Hier die Ausgabe von survey:

Peter

Autor: Peter Sieg (Gast)

Datum: 11.04.2011 13:18

---

Ich denke, wenn die fat16 Version auch für 8-bit ok ist, sollten wir diese als Basis für die weitere Entwicklung nehmen.

Evtl. mal ein Diff gegen V2/trunc fahren..

Meines Wissens hatte Leo C. das Diskettenformat vergrößert und evtl. ist der Z80 Emulationscode in dieser Versionen schon weiter fortgeschritten??

@Leo C: Liest du noch mit..?

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 11.04.2011 23:20**

---

Ja Peter, ich lese noch mit.

Inzwischen habe ich auch wieder etwas Zeit für das Projekt, aber wohl nicht mehr so viel wie letzten Sommer.

Am Samstag habe ich endlich mal meinen letzten Softwarestand in das [SVN](#)-Repository eingechekkt. Die Software funktioniert bei mir, ist aber nicht besonders gut getestet. Wegen der Features (FAT16, Disk Formate) bitte nach meinen letzten Beiträgen hier im Forum suchen. Danach hier fragen.

Die aktuelle Software ist für gewöhnlich hier zu finden:

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

Oder besser, mit einem [SVN](#)-Client:

<http://cloudbase.homelinux.net/svn/avr-cpm>

Als Client für Windows ist tortoissvn zu empfehlen:

<http://tortoissvn.tigris.org/>

Übrigens Peter, bei [SVN](#) steht trunk für Stamm, nicht etwa für Kofferraum oder Rüssel. :)

In trunk ist üblicherweise die Hauptentwicklungslinie.

Wegen Z80 Unterstützung könnte ja mal jemand bei "Horst S. (h3aa)" nachfragen. ([Beitrag "Re: CP/M auf ATmega88"](#))

Joe, wenn Du mit den SD-Karten-Problemen nicht weiterkommst, kann ich versuchen, zu helfen.

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 08:30**

---

@Joe: hast du mal probiert mit dem Debugschalter:

```
FAT16_DEBUG = 1
```

mehr Informationen zu bekommen?

Zur SD Card Initialisierung gibts ja einige Projekte zum vergleichen/kopieren: Z.B [www.sd2iec.de](http://www.sd2iec.de) und sicher auch hier im Forum..

[Beitrag "MMC/SD-Karte mit FAT16 an AVR"](#)

[Beitrag "MMC SD library FAT16 FAT32 read write"](#)

---

@Leo C: Prima! D.h. der trunk enthält jetzt die neueste Version?

Das Problem, was ich whrscheinlich damit habe ist, das es sich weit von Win\$ entfernt hat..? D.h mein ipl+bios etc. und Diskformate muß ich anpassen/ändern, oder kann ich die orig. Formate+Bios beibehalten

(vorerst)?

Ist den da neuerer Z80 code von Horst S. (h3aau) schon drin, oder noch nicht?

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 11:25**

---

Ich habe mir trunk gerade mal gezogen. Konnte ich für m168, 30MHz, 4-bit Dram und fat16 ohne Fehler übersetzen.

Frage: Kann ich das mit den 'alten' Bios+ipl Diskimages (243kb) nutzen?

Um die Assemblierung der bios.mac +ipl.mac möchte ich mich dann erst später kümmern...

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 12:02**

---

Bios assemblieren:

zxcc habe ich für Dos/Win\$ nicht gefunden.

m80180.zip habe ich probiert:

\*BIOS,TTY:=BIOS wirft 38 fatale Fehler aus.

Das ist so für mich natürlich keine akzepable 'Lösung'..

Ich brauche etwas, womit ich die unter Win\$ assemblieren kann..

@Leo C: Kannst du die \*.mac nicht wieder per awk? script in asm zurück wandeln?

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 12.04.2011 12:34**

Angehängte Dateien:

- [cpm.bin](#) (6,5 KB, 43 Downloads)

---

Im Anhang ist jetzt mal ein aktuelles System.

Zum Rest schreibe ich später nochmal mehr. Jetzt mal auf die Schnelle...

[Peter Sieg schrieb:](#)

> Bios assemblieren:

> zxcc habe ich für Dos/Win\$ nicht gefunden.

[Beitrag "Re: CP/M auf ATmega88"](#)

[Beitrag "Re: CP/M auf ATmega88"](#)

Darüber, und auch sonst, sollte es kein Problem sein,

<http://www.seasip.demon.co.uk/Unix/Zxcc/>

zu finden.

Das Archiv der 0.4 Version enthält auch Windows Binaries.

<http://www.seasip.demon.co.uk/Unix/Zxcc/zxcc-0.4.0.tar.gz>

```
> m80180.zip habe ich probiert:  
> *BIOS,TTY:=BIOS wirft 38 fatale Fehler aus.
```

Ich meine mich zu erinnern, daß ich das mal kurz ausprobiert hatte, in dem ich alle benötigten Dateien in ein Verzeichnis kopiert hatte. Um das alles richtig unter Windows zu installieren, fehlt mir die Zeit, vernünftiges Gerät und die Motivation. Hier wird es doch den ein oder anderen Windowsprogrammierer geben, der das mal richtig installieren und das Makefile ggf. anpassen kann. Mein Provisorium hat sicher funktioniert, sonst hätte ich diese Möglichkeit hier garnicht erwähnt.

```
> Das ist so für mich natürlich keine akzeptable 'Lösung'..
```

Das ist natürlich schlimm für Dich.

```
> Ich brauche etwas, womit ich die unter Win$ assemblieren kann..  
> @Leo C: Kannst du die *.mac nicht wieder per awk? script in asm zurück  
> wandeln?
```

nein.

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 12:59**

---

Ja nun habe ich auch gelesen, das in der 4.0 Version Dos/Win Programme enthalten sind. In der 5er aber nicht. Habe ich getestet (alles in 1 Verzeichnis), bekommen auch hier mit: zxcc m80 -=bios.mac 38 fatale Fehler??

---

Hier:

<http://hc-ddr.hucki.net/wiki/doku.php/cpm:windows>

habe ich auch den cpm player getestet.

Dort bekomme ich mit cpm m80.com =bios.mac 41 fatale Fehler?

bios.rel wird aber erzeugt.

G:\temp>cpm 180.com bios.rel,bios.bin/N/E

Link-80 3.44 09-Dec-81 Copyright (c) 1981 Microsoft

Data 0100 0328 < 552>

53266 Bytes Free

[0000 0328 3]

G:\temp>

---

@Leo C. Ja ich weiß.. mal sehen.. aber wenn alle Stricke reißen bleibe ich eben bei dem Code den ich aktuell habe.. das war auch ein Grund, damals ersteinmal nicht mehr weiter 'mitzumachen' - ist halt (für mich!) blöd, wenn das nach Linux Only abwandert und ich/Windows außen vor bleiben.

Nun, evtl. findet ja jemand noch woran es liegt bzw. eine funktionierende Lösung..

Kann ich denn den [AVR](#) Code aus trunk mit dem 'alten' Bios verwendet, oder ist zwingend auch das aktuelle Bios zu verwenden?

Peter

**Autor: Joe G. ([feinmechaniker](#))** ☰

**Datum: 12.04.2011 14:58**

---

[Peter Sieg schrieb:](#)

> Kann ich denn den [AVR](#) Code aus trunk mit dem 'alten' Bios verwendet,  
> oder ist zwingend auch das aktuelle Bios tz verwenden?

Nein geht nicht.

Joe

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 15:40**

---

Jup. Geht nicht.. habe ich gerade probiert..  
Leider noch eine schlechte Nachricht. FAT16 scheint noch Probleme  
zumindest beim schreiben zu haben.. BDSC (C-Compiler) meldet bei der  
trunk Version:  
BDos Err on C..  
Und die fat16-test meldet bei clink mmind - No main defined - aborted.  
Wenn ich mit Partitionen arbeite geht fat16-test einwandfrei.. trunk  
muss  
ich noch testen..

:-(

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 16:01**

---

arg. Muss mich mal anmelden, damit ich editieren kann..

---

fat16-test und trunk lassen sich nicht mehr für [atmega88](#) übersetzen  
- auch ohne fat16support!  
Das kann der [atmega88](#) nicht ab:  
call uartputc

Peter

**Autor: Andreas H. ([andreas\\_h35](#))**

**Datum: 12.04.2011 16:15**

**Angehängte Dateien:**

- [aliados.zip](#) (1,3 MB, 28 Downloads)

---

Hallo Peter und Leo,

ich habe hier mal das ALIADOS im Cygwin übersetzt. Es ist eine  
einigermaßen lauffähige CPM-Emulation - eigentlich für Linux - die aber  
auch unter Windows spielt (hier bei mir unter Win7)

Nach dem Start landet man auf Drive A: und das aktuelle Verzeichnis ist  
dann auch im CPM auf A: zu sehen. Wenn dort M80 und L80 liegen, kann man  
sogar mit:

```
aliados M80.COM =blafasel.asm ..
```

aus der DOS-Shell assemblieren.

Hier habe ich das gefunden: <http://www.arrakis.es/~ninsesabe/aliados/>

Im Ziparchiv liegt das aliados.exe und die notwendigen Cygwin-DLLs.  
Bitte die DLLs im gleichen Verzeichnis wie das .exe legen.

Wer es braucht, kann es ja mal checken, evtl. hilft das den  
Win\$-Freunden bei der BIOS Assemblierung.

Gruss,  
Andreas

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 16:57**

---

@andreas: Danke. Arbeitet aber genauso wie oben cpmplayer etc. m80  
meldet wieder 38 fatal errors..

Ich denke die haben ein Problem mit der Quelldatei..?

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 12.04.2011 17:34**

---

Ich habe gerade mal bios.mac in die AltairSIMH cpm2.2 disk geladen und  
dort m80 =bios.mac aufgerufen.. auch 38 fatal errors..??

Kann es sein, das ich nicht das richtige m80.com + l80.com habe..??

Bitte mal die richtigen Versionen hier anhängen.. dann solltes das ggf.  
mit

einige hier weiter oben aufgeführten Tools gehen.. denn die Meldung 38  
fatal errors hatte ich schon mehrere Male mit unterschiedlichen Tools..

Peter

**Autor: Andreas H. ([andreas\\_h35](#))**

**Datum: 12.04.2011 17:55**

**Angehängte Dateien:**

- [z80asm\\_linker.zip](#) (20,6 KB, 30 Downloads)

---

@Peter: ich konnte mit M80 und L80 das bios.mac aus dem [SVN](#)/trunk prima  
assemblieren und linken (habe das aliados unter macosx verwendet):

```
lunatix:cpm hein$ aliados m80.com =bios.mac
```

```
No Fatal error(s)
```

```
lunatix:cpm hein$ aliados l80.com bios,bios/n/e
```

```
Link-80  3.44  09-Dec-81  Copyright (c) 1981 Microsoft
```

```
Data    0100    0518    < 1048>
```

```
52514 Bytes Free
```

```
[0000  0518    5]
```

hier meine eingesetzten Tools.

Andreas

Autor: Leo C. ([rapid](#))

Datum: 12.04.2011 18:18

---

Peter Sieg schrieb:

```
> Ich habe gerade mal bios.mac in die AltairSIMH cpm2.2 disk geladen und  
> dort m80 =bios.mac aufgerufen.. auch 38 fatal errors..??
```

Kann es sein, daß Dein M80 die Includedateien 'avrcpm.lib' und 'cfgacpm.lib' nicht findet?

Autor: Peter Sieg (Gast)

Datum: 12.04.2011 20:02

---

Ja, so ist/war es ;-)

aliados geht!! Habe inzwischen das cpm.bin komplett neu erstellen können.

Nun auch neue diskimages erzeugt und unter Knoppix auf cp/m Partitionen mit dd geschrieben..

=> Bin jetzt auf neuestem trunk level ;-)

Trotzdem gibt es leider unter fat16 ein Schreibproblem!

@Andreas: kannst du bitte mal versuchen auch die cpmttools unter deiner cygwin Umgebung zu compilieren? Dann haben wir die cygwin dll's gemeinsam um

brauchen nicht ggf. 2 Versionen. Dazu haben die cpmttools unter Windows noch

2 Unschönheiten:

1. diskdefs wird FEST unter x.\cpmttools gesucht.

2. diskdefs muss nach Unix Konvention nur mit LF als Zeilenende versehen sein

(kein CR+LF wie unter Dos/Win üblich)

<http://www.moria.de/~michael/cpmttools/>

bei zxcc und cpm (cpmplayer) scheinen die lib's/includes von m80/l80 nicht

gefunden zu werden.. schaue ich mir noch mal an.. ob man das nicht noch hinbekommt..

Fazit: Man kann alles unter Windows assemblieren!

Peter

Autor: Peter Sieg (Gast)

Datum: 12.04.2011 20:17

Angehängte Dateien:

- [m80l80pc.zip](#) (103,7 KB, 29 Downloads)

---

Hmm.. m80l80pc geht nun auch.. vielleicht hatte ich wirklich nur die

includes nicht im Verzeichnis..

Ich hänge das m80180pc.zip inkl. aller benötigten Dateien hier an.

zxccc+cpmplayer schaue ich mir jetzt auch noch mal an..

Peter

Autor: Andreas H. ([andreas\\_h35](#))

Datum: 12.04.2011 21:07

Angehängte Dateien:

- [cpmtools-cygwin.zip](#) (1,4 MB, 26 Downloads)

---

@Peter:

hier das cpmtools als cygwin-binary, ohne MAN-Pages und Sourcecode.  
Das diskdefs File wird im aktuellen Verzeichnis gesucht.  
Andreas

Autor: Peter Sieg (Gast)

Datum: 12.04.2011 23:00

---

Super! Danke!

Damit läuft cpmtools+aliados mit denselben dll's.

Peter

Autor: Peter Sieg (Gast)

Datum: 13.04.2011 13:07

Angehängte Dateien:

- [avrcpm\\_fat16.zip](#) (153,3 KB, 31 Downloads)
- [avrcpm\\_aliados.zip](#) (2 MB, 30 Downloads)

---

Ich hänge jetzt hier mal an: avrcpm\_fat16.zip = neueste trunk Quellen  
inkl. AVR Studio 4 Projekt und avrcpm\_aliados.zip = alles zum bilden von  
Bios+Ipl, CPM.BIN und Disk Images mit cpmtools.

Das sind außer der Hardware alles was es unter Windows braucht..

Atfo#ebit läuft es prima mit Part

problem.

Bei Leo C. sollte es aber laufen (Part.?)

at16?)..

Autor: Leo C. ([rapid](#))

Datum: 13.04.2011 17:26

---

@Peter,

bitte vergrößere Deine FAT16-Imagedateien mal auf (mindestens) 256  
KByte.

Unser Treiber kann das leider nicht selbst.

Vergrößern geht z.B. durch anhängen von Bytes mit dd, oder mit makeimage von Frank Zoll, das im [SVN](#) in trunk/tools/makeimage zu finden ist. Wenn man bei makeimage die Größe wegläßt, wird die Datei "zufällig" auf 256 KB vergrößert.

Neue, frisch formatierte Images kann man auch folgendermaßen erzeugen, damit sie von Anfang an die richtige Größe haben:

```
$ makeimage /dev/null CPMSDK_x.IMG
```

mkfs.cpm kann man sich dann sparen.

Unter Windows muß man /dev/null evtl. durch etwas anderes ersetzen.

**Autor: Peter Sieg (Gast)**

**Datum: 13.04.2011 17:38**

---

Hmm. Was mir aufgefallen ist. Test 4 = V2.1, ab Test5 V2.0?

Alle Test sind mit fat16 Fileimages gemacht.

Kannst du bitte auch mal 8-bit testen, ohne fat16 Images auf die SD Karte zu tun, sondern anstatt dessen mal Partitionen zu verwenden?

Ich verwende übrigens auch 2 128Mb SD Karten: Noname + Kingston.

@Leo C: bei dir läuft aber die trunk in 8-bit?

Es sind doch mind. 10 Platinen mit der 8-bit 4MB + VGA Teil in Betrieb.. welche Version wird denn dort eingesetzt..?

Peter

**Autor: Peter Sieg (Gast)**

**Datum: 13.04.2011 17:48**

---

@Leo C: Du meinst/vermutest, das beim schreiben über die aktuelle Größe hinausgegangen wird und dann gibts Probleme.. weshalb diese Mindestgröße sein muß?

makeimage - liegt nur als c Source vor. Ich habe keinen Compiler installiert z.Z :-)

Und wenn sich jemand findet (@Andreas?) der das unter Windows / Cygwin kompiliert, dann bitte auch evtl. Unix Spezifika (/dev/null) entsprechen für uns Win\$ User händelbar machen..

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 13.04.2011 18:00**

---

[Peter Sieg schrieb:](#)

> @Leo C: bei dir läuft aber die trunk in 8-bit?

Logo. 4Bit ist (glaube ich) noch auf meinem Steckbrett, und habe ich schon lange nicht mehr getestet.

> Es sind doch mind. 10 Platinen mit der 8-bit 4MB + VGA Teil in Betrieb..

> welche Version wird denn dort eingesetzt..?

Keine Ahnung. Ich habe hier meinen eigenen Lochrasteraufbau.

Hier gibts ein Bild:

[Beitrag "Re: CP/M auf ATmega88"](#)

Autor: Leo C. ([rapid](#))

Datum: 13.04.2011 18:13

---

[Peter Sieg schrieb:](#)

> @Leo C: Du meinst/vermutest, das beim schreiben über die aktuelle Größe  
> hinausgegangen wird und dann gibts Probleme.. weshalb diese Mindestgröße  
> sein muß?

Nein, ich weiß, daß der Treiber auf das Dateiende testet, und nicht darüber hinaus schreibt. Stattdessen meldet er einen Fehler zurück. Der Treiber könnte an der Stelle auch die Datei vergrößern (sofern Platz auf dem Datenträger), aber der Aufwand dafür ist nach Franks und meiner Meinung nach unverhältnismäßig groß.

> makeimage - liegt nur als c Source vor. Ich habe keinen Compiler  
> installiert z.Z :-(

Das Programm ist in anspruchlosestem ANSI-C geschrieben, und kann mit jedem Compiler auf wahrscheinlich jeder Plattform übersetzt werden.

> Und wenn sich jemand findet (@Andreas?) der das unter Windows / Cygwin  
> kompiliert, dann bitte auch evtl. Unix Spezifika (/dev/null) entsprechen  
> für uns Win\$ User händelbar machen..

Andreas wird das sicher machen können. Cygwin braucht man dafür nicht, und das Programm hat keine Unix Spezifika.

Autor: Peter Sieg (Gast)

Datum: 13.04.2011 18:18

Angehängte Dateien:

- [make\\_disk.zip](#) (762 Bytes, 35 Downloads)

---

@Leo C. das Schreibproblem ist gelöst! Danke!

Ich habe die \_C.IMG Datei mit dd (in avrcpm\_aliados.zip) so auf 256k gebracht:

```
dd if=CPMDSK_C.IMG of=CPMDSK_D.IMG ibs=256k conv=sync
```

Dann auf SD Karte..

d:

```
D>cc mmind.c
```

```
BD Software C Compiler v1.60 (part I)
   35K elbowroom
```

```
BD Software C Compiler v1.60 (part II)
   32K to spare
```

```
D>clink mmind
```

```
BD Software C Linker v1.60
```

```
Last code address: 1B4D
```

```
Externals start at 1B4E, occupy 0000 bytes, last byte at 1B4E
```

```
Top of memory: E405
```

```
Stack space: C8B8
```

Writing output...  
42K link space remaining  
D>mmind  
Random seed generation.. wait.. hit a key.  
Guess my 4-digit number...(\*\*\*\*); 0000 to abort.  
  
Guess 1:0000  
  
You gave up..  
Press return to exit.  
D>

Funktioniert also!  
Ich habe das mal in meinem 'make\_disk.bat' eingebaut.. bitte entzippen  
und  
die alte Version damit ersetzen..

Peter

**Autor: Peter Sieg (Gast)**  
**Datum: 13.04.2011 18:27**

---

[Leo C. schrieb:](#)

> Nein, ich weiß, daß der Treiber auf das Dateiende testet, und nicht  
> darüber hinaus schreibt. Stattdessen meldet er einen Fehler zurück. Der  
> Treiber könnte an der Stelle auch die Datei vergrößern (sofern Platz auf  
> dem Datenträger), aber der Aufwand dafür ist nach Franks und meiner  
> Meinung nach unverhältnismäßig groß.

ok. Danke ich auch. Wenn man es weiß, ist es leichter die diskimage  
mind.  
so groß zu machen, wie diskdefs es definiert. Aktuell wären das ca. 243k  
gewesen.. richtig?

(Bei Gelegenheit muß ich dann mal schauen, was du da schon an  
Unterstützung von größeren Diskimagesgezauberst hast.. und wie ich das  
nutzen kann..)

Aber ersteinmal die 8-bit Version haben und am laufen haben..  
Dann Z80..

Peter

**Autor: Leo C. ([rapid](#))**  
**Datum: 13.04.2011 18:30**

---

Hallo Andreas,

[Andreas H. schrieb:](#)

> wo kommt das makeimage her ? (URL!! dann übersetzte ich das euch wieder  
> für cygwin

Die Software liegt auf meinem kleinen [SVN](#)-Server (Kopie von oben):

Die aktuelle Software ist für gewöhnlich hier zu finden:  
<http://cloudbase.homelinux.net/viewvc/avr-cpm/avrcpm/trunk/>

Oder besser, mit einem [SVN](#)-Client:  
<http://cloudbase.homelinux.net/svn/avr-cpm>

Als Client für Windows ist tortoisessvn zu empfehlen:

<http://tortoisesvn.tigris.org/>

Wer Verbesserungen und/oder Erweiterungen beisteuern will, kann von mir gerne einen Schreibzugriff bekommen.

Autor: Andreas H. ([andreas\\_h35](#))

Datum: 13.04.2011 18:41

Angehängte Dateien:

- [makeimage.exe](#) (18,6 KB, 62 Downloads)

---

hier das makeimage.exe, lauffähig mit den Cygwin.DLLs die Ihr schon habt. /dev/null geht wie unter Linux auch, da die Argumente ja in der Cygwin-Runtime ausgewertet werden.  
Andreas

Autor: Leo C. ([rapid](#))

Datum: 13.04.2011 18:58

---

Peter Sieg schrieb:

> so groß zu machen, wie diskdefs es definiert. Aktuell wären das ca. 243k  
> gewesen.. richtig?

Fast. Die System-Tracks müssen ja auch noch reinpassen.

>

> (Bei Gelegenheit muß ich dann mal schauen, was du da schon an  
> Unterstützung von größeren Diskimagesgezauberst hast.. und wie ich das  
> nutzen kann..)

Folgende sollten gehen:

simhd, simh altair 8800 hard disk format

MyZ80

YAZE

Siehe auch hier:

[Beitrag "Re: CP/M auf ATmega88"](#)

[Beitrag "Re: CP/M auf ATmega88"](#)

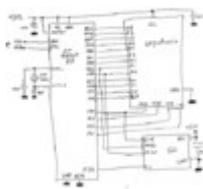
> Dann Z80..

Dazu habe ich vor 2 Tagen etwas geschrieben.

Autor: Peter Sieg (Gast)

Datum: 13.04.2011 20:15

Angehängte Dateien:



- [schema.png](#)  
40,2 KB, 225 Downloads

---

@Andreas: ich wollte mir das auch schon mal auf LR aufbauen. bin aber zeitlich nicht dazu gekommen.. Basis wäre die 4-bit Skizze. Aber Achtung! Völlig geänderte Pinzuordnungen bei 8-bit! D0..D3 des 2ten Drams ist dann D4..D7. Uart ändert sich ebenfalls (HW-Uart -> SW-Uart). Hi. Ich möchte eine 8-bit Version auf LR aufbauen. Wie genau muß ich was verdrahten? Ich gehe im Moment davon aus:

1. 2ten Dram Chip 1:1 mit dem ersten verbinden (Huckepack) - bis auf D0-D3!
2. Dann nach Port Declarations aus config.inc gehen:

```
-----  
; Port declarations  
  
#if DRAM_8BIT /* Implies software uart */  
  
;Port D  
.equ RAM_D0 = 0  
.equ RAM_D1 = 1  
.equ RAM_D2 = 2  
.equ RAM_D3 = 3  
.equ RAM_D4 = 4  
.equ RAM_D5 = 5  
.equ RAM_D6 = 6  
.equ RAM_D7 = 7  
.equ RAM_A0 = 0  
.equ RAM_A1 = 1  
.equ RAM_A2 = 2  
.equ RAM_A3 = 3  
.equ RAM_A4 = 4  
.equ RAM_A5 = 5  
.equ RAM_A6 = 6  
.equ RAM_A7 = 7  
  
;Port B  
.equ MMC_MOSI = 3  
.equ MMC_MISO = 4  
.equ MMC_SCK = 5  
.equ RAM_A8 = 3  
.equ RAM_A9 = 4  
.equ RAM_A10 = 5  
  
.equ RXD = 0  
.equ TXD = 1  
.equ MMC_CS = 2  
  
.equ P_MMC_CS = PORTB  
.equ P_A8 = PORTB  
.equ P_RXD = PORTB  
  
;Port C  
.equ RAM_RAS = 0  
.equ RAM_CAS = 1  
.equ RAM_OE = 2  
.equ RAM_W = 3  
  
.equ P_RAS = PORTC  
.equ P_CAS = PORTC  
.equ P_OE = PORTC  
.equ P_W = PORTC
```

```
#else /* 4 bit RAM, hardware uart */
```

Irgendwo hatte ich die anderen Pin's zum Dram auch mal in einer Zeichnung drin gehabt.. finde die aber so schnell nicht mehr..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 14.04.2011 09:38

Angehängte Dateien:



• [avrcp\\_m\\_sp.pdf](#) | [anzeigen](#)  
60,4 KB, 152 Downloads

---

[Andreas H. schrieb:](#)

> Hat jemand schon einen fertigen Schaltplan für die 8bit [AVR](#)-Maschine

Hier die 8-Bit Variante mit 4 MB x 4 RAM's.

@Peter

Ja, deine Portzuweisungen sind korrekt. Bei den Chips 4256 existieren dann A9 und A10 halt nicht, ansonsten alles so wie auf der Schaltung. Wenn alles mit 3.3V versorgt wird kann IC2 entfallen, dann aber unbedingt die MMC ziehen wenn über ISP und 5V programmiert wird.

Zwei Baustellen, Softwaretest und Hardwareüberarbeitung waren zu viel. Deshalb noch etws Geduld mit der neuen LP.

Joe

Autor: Peter Sieg (Gast)

Datum: 14.04.2011 18:03

---

@Joe: Ich denke wir haben für uns schon ganz schön was geschafft.. Ich bin jetzt endlich mal wieder auf dem aktuelle SW-Stand.. und fat16 sieht gut aus.. bleibt noch das mmc/8-bit Problem.. wenn wir mehr Hardware draußen haben.. gibts auch wieder etwas mehr Mitstreiter..

Also immer mit der Ruhe.. Gut Ding will Weile haben..

Peter

Autor: Hans- w. Schütz ([hschuetz](#))

Datum: 14.04.2011 20:27

---

Hallo Leute,  
meine Hardware läuft noch mit einer sehr alten 8bit Version. Interessant wäre die Z80 Implementation und die Ausnutzung der MMC/SD Karte, am besten alles mit dem Windows PC ladbar. Leider bin ich nicht der Softwarespezialist...CPM 2.2 behersche ich gerade ein bisschen...mit echtem Z80. Aber ich lese hier weiter mit... mal sehen was noch wird.

Gruß  
Hans-Werner

Autor: Leo (Gast)

Datum: 15.04.2011 09:01

---

[Hans- w. Schütz schrieb:](#)

> meine Hardware läuft noch mit einer sehr alten 8bit Version. Interessant  
> wäre die Z80 Implementation und die Ausnutzung der MMC/SD Karte, am besten  
> alles mit dem Windows PC ladbar.

Wenn Du mit "Ausnutzung der MMC/SD Karte" größere CP/M-Diskimages meinst, kannst Du auf die Schnelle mal folgendes ausprobieren:

- Software (avr und BIOS) auf neuesten Stand bringen  
<http://cloudbase.homelinux.net/viewvc/avr-cpm/avr-cpm/trunk/>
- Bootbare SD-Karte herstellen. Entweder mit System in CP/M-Partition oder als CPMDISK\_A.IMG auf einer FAT16-Partition
- Das MyZ80-Image aus dem folgenden Beitrag runterladen und als CPMDISK\_B.IMG auf die FAT16-Partition kopieren:  
[Beitrag "Re: CP/M auf ATmega88"](#)

Das Ergebnis sollte ein 8 MB großes Laufwerk B: unter CP/M sein.

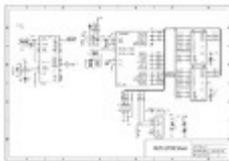
Wenn Deine cpmttools mit libdisk kompiliert sind, kann das Image auch auf dem PC mit Dateien gefüllt werden.

> cpmcp -f myz80 ...

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 15.04.2011 12:47

Angehängte Dateien:



• [avr-cp-m.pdf](#) | [anzeigen](#)  
25,4 KB, 145 Downloads

---

Hallo,

hier der Entwurf der 8-Bit Minimalvariante.

Optionen:

- FTDI durch Peters Platine ersetzen
- ISP Buchse einsparen (Programmierung über Bootloader)

Bitte Vorschläge, Änderungswünsche, Kritik....

Joe

Autor: Nils Eilers ([yetanotheruser](#))

Datum: 16.04.2011 00:26

---

Ich würde den FTDI nicht ersetzen, sondern zusätzlich eine Stiftleiste

vorsehen, damit man wahlweise den FTDI oder Peters Adapterplatine oder meinen [MAX232](#) verwenden kann.

Ich kenne jetzt die Belegung der Stiftleiste nicht, aber wenn dort TxD, RxD, GND und 3V3 drauf wären, wäre ich schon sehr zufrieden.

Auf die ISP-Buchse würde ich nicht verzichten. Das habe ich bei einem anderen Projekt (notgedrungen) und vermisse die Möglichkeit, direkt über ISP programmieren zu können, doch sehr.

Autor: Joe G. ([feinmechaniker](#))

Datum: 16.04.2011 11:11

---

[Peter Sieg schrieb:](#)

> GND / Pin 8 am [AVR](#) ist auch mit GND verbunden

Danke, die nicht mehr vorhandene Verbindung war dem Löschen geschuldet.

Ich würde den FTDI sozusagen an den Kopf der Platine setzen, dann eine Stiftleiste um mit Peters Adapter kompatibel zu sein und dann den Rest. Die bisherige USB Buchse würde ich gerne durch einen Stecker ersetzen, siehe hier:

<http://www.reichelt.de/USB-FireWire-Stecker/USB-AW...>

Die derzeitige Kabelverbindung ist immer so schwer, dass alles vom Schreibtisch fliegt :-)

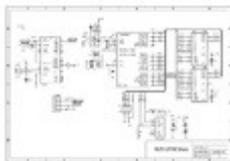
ISP lasse ich drauf.

Joe

Autor: Joe G. ([feinmechaniker](#))

Datum: 16.04.2011 16:07

Angehängte Dateien:



• [avrcp\\_m.pdf](#) | [anzeigen](#)  
25,9 KB, 141 Downloads

---

Hier die die Version mit USB-Type A Stecker und Adapter zur "Peterplatine"

Peter, bitte mal schauen, ob die Anschlußbelegung korrekt ist. Gerade bei TXD und RXD bin ich mir nicht sicher weil ich nicht weiß wo diese Pins am IC liegen.

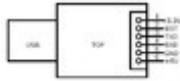
Joe

Autor: Peter Sieg (Gast)

Datum: 16.04.2011 18:42

Angehängte Dateien:

- Built on USB to UART1232 Transceiver chip
- Designed for the need for serial 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: 36 mm x 16 mm



• [usb ttl module pinout.jpg](#)

54,7 KB, 196 Downloads

@Joe: Sieht für mich ok aus.

Peter

Autor: Hans- w. Schütz ([hschuetz](#))

Datum: 16.04.2011 20:01

Hallo Peter,  
gibt es das Modul auch in Deutschland zu kaufen? Oder nur in Hongkong..  
habe da sehr schlechte Erinnerungen....  
Anmerkung zur Schaltung... Wo bleiben diejenigen welche noch eine  
serielle Schnittstelle haben... mir fehlt da ein Spannungsregler 3,3V...  
Gruß  
Hans-Werner

Autor: Peter Sieg (Gast)

Datum: 17.04.2011 11:08

Ob es das Modul noch - zu diesem Preis - woanders gibt könnte google  
klären.  
Ich habe da mit der Bestellung sehr gute Erfahrungen gemacht!  
Ansonsten bleibst ja noch [FT232RL](#) + USB Buchse. Der [FT232](#) stellt  
ebenfalls  
3,3V zur Verfügung. Und wenn das immer noch nicht reicht.. 3,3V  
Netzteile gibt es bei Pollin + ungebrautes Handydatenkabel = TTL<->USB  
Schnittstelle.

Peter

Autor: Peter Sieg (Gast)

Datum: 18.04.2011 14:38

@Joe: Wenn du dann bei Eagle und dem layouten bist.. kannst du bitte die  
4-bit Platine auch noch mal anpassen.. Da war ja Pin 24 - /WE und Pin 27  
- IO3 getauscht worden.. ich bekomme das irgendwie in eagle nicht hin..  
dann hat man, falls man noch mal die 4-bit Version fertigen will gleich  
die richtige Platine..

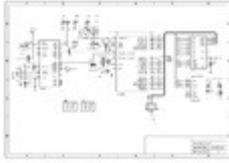
Danke, Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 18.04.2011 14:52

Angehängte Dateien:

- [V.01.01.0.zip](#) (85,7 KB, 28 Downloads)



- [avrcp\\_m\\_v1.pdf](#) | [anzeigen](#)  
25,2 KB, 133 Downloads

---

[Peter Sieg schrieb:](#)

> kannst du bitte die 4-bit Platine auch noch mal anpassen

Bitteschön  
Joe

Autor: Peter Sieg (Gast)  
Datum: 18.04.2011 19:32

---

Dankeschön! ;-)

Für die kommende Easy-8-bit Platine bitte hinter dem CP2102 Adapter vor der Versorgung der IC's+SD etc. in der +3.3V Leitung einen Jumper vorsehen für einen Ein-/Ausschalter.. so kann man schon einstecken und die USB Schnittstelle wird erreichbar.. Terminalprogramm kann gestartet werden und dann erst kann man dem Gerät seinen Strom geben und kann so die Bootmeldungen auch im Terminalprogramm verfolgen.. ohne diese Schaltmöglichkeit sieht man nach <Return> nur noch den A> Prompt..

Peter

Autor: Peter Sieg (Gast)  
Datum: 19.04.2011 09:02  
Angehängte Dateien:

- [CPMDSK\\_D.zip](#) (73,5 KB, 46 Downloads)



- [avrcp\\_m\\_v1.png](#)  
170,4 KB, 385 Downloads

---

@Andreas+Joe: Ihr habt Recht. An den Resetschalter hatte ich gar nicht mehr gedacht (fehlt auf meinem LR Aufbau). Das reicht..

Habe auch mal das myz80 8MB Image probiert. Klappt wunderbar.. 8MB große Disketten.. wie bekommen wir die nur gefüllt.. ;-)  
Muß mir jetzt mal die diskdefs Einträge für myz80 herausuchen..



> Prima! Da ich kein [SVN](#) inst. habe.. welche Datei(en) muß ich mir  
> herunterladen..?

Um alles herunter zu laden, begeben man sich z.B. hier her:  
<http://cloudbase.homelinux.net/viewvc/avr-cpm/avrcpm/trunk/>

Dann klicke man rechts oben auf "Download Tarball".

Wenn man nur einen Teil (z.B. den CP/M-Teil) haben will, steige man in das entsprechende Directory hinab, und klicke eben da auf "Download Tarball".

Wenn man nur eine einzelne Datei herunter laden will, klicke man auf den Namen der Datei oder auf die Revisionsnr. Anschließend findet man rechts oben "Download File" (Imho umständlich und nicht empfehlenswert).

Wenn man aber öfters mal nur die geänderte(n) Datei(en) laden will, installiert man (unter Windows) am Besten TortoiseSVN (<http://tortoisesvn.tigris.org/>).

Das ist einfach eine Erweiterung des Windows Filemanagers um die [SVN](#)-Client-Funktionen (Siehe Anhang).

Autor: Leo C. ([rapid](#))

Datum: 20.04.2011 17:32

Angehängte Dateien:

- [simh-imgs.zip](#) (567,9 KB, 47 Downloads)

---

[Peter Sieg schrieb:](#)

> Muß mir jatz mal die diskdefs Einträge für myz80 heraussuchen..

In meiner diskdefs ist folgender Eintrag

```
# MYZ80 hard drive (only works with libdisk, because it has a 256-byte header)
diskdef myz80
  seclen 1024
  tracks 64
  sectrk 128
  blocksize 4096
  maxdir 1024
  skew 1
  boottrk 0
  os 3
end
```

Der war wahrscheinlich in der Original-Datei, bin mir aber nicht sicher.

> Wer hat denn ggf. noch andere Anwendungen, die jetzt schon mit  
> emulierter 8080 CPU laufen..?

Ich hatte ja schonmal die die Diskimages vom simh Altair 8800 Emulator empfohlen:

[Beitrag "Re: CP/M auf ATmega88"](#)

Für die Spielkinder unter uns habe ich mal das Basic- und das Games-Image auf HD-Format umkopiert und als Zip hier angehängt. Viel Spaß damit.

Autor: Leo C. ([rapid](#))

Datum: 21.04.2011 17:11

Angehängte Dateien:

- [simhd-bootimage.zip](#) (57,9 KB, 46 Downloads)

---

Von simhd kann jetzt auch gebootet werden. Um ein solches Bootimage zu erkennen, muß im 1. Sektor (ipl) eine Marke vorhanden sein. Das cpm/Makefile kann diese Marke erzeugen. Z.B. so:

```
$ make clean
```

```
$ make diskimage IMGFORMAT=simhd
```

Oder das Makefile entsprechend anpassen.

Ein fertiges Image habe ich hier mal angehängt.

---

Autor: Peter Sieg ([petersieg](#))

Datum: 12.05.2011 09:43

---

Na.. ein bißchen ruhig geworden hier..?

@Joe: Stand 8-bit Simple Platinen?

Ich hatte hh3au (oder so ähnlich) mal per Mail kontaktiert, ob er die Z80 Emulation nicht noch abschließen könnte.. ohne jede Antwort.. :-(

Gruß Peter

---

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 12.05.2011 10:01

---

[Peter Sieg schrieb:](#)

> @Joe: Stand 8-bit Simple Platinen?

Ich mußte erst noch ein anderes Projekt abschließen. Der Schaltplan steht ja, das Layout auch in den Grundzügen, es muß nur noch "schön" gemacht werden.

Gruß Joe

---

Autor: Horst S. ([h3aau](#))

Datum: 14.05.2011 00:26

---

moin moin,

z80 liegt gerade aus gesundheits gründen auf eis.....  
will wer anders das beschleunigen?

g. Horst

---

Autor: Peter Sieg (Gast)

Datum: 14.05.2011 10:01

---

Ah. Hallo Horst. Schön wieder von Dir zu hören.  
Ich zumindest nicht aktuell.. Gut Ding will Weile haben.  
Freue mich das du noch mit im Boot bist!

Gesundheit geht vor! Alles Gute derweil!

Falls du aber irgendwann schon einen lauffähigen Zwischenstand hättest, würde wir den gerne schon mal im Code mit aufnehmen.

Gruß Peter

## AVR CP/M USB Stick

Autor: Joe G. ([feinmechaniker](#))

Datum: 07.07.2011 15:50

Angehängte Dateien:

- [avrcp\\_m.zip](#) (85 KB, 56 Downloads)



- [CP\\_M Stick.jpg](#)  
64,4 KB, 332 Downloads

---

Hat etwas gedauert...

Hier nun die 8-Bit Variante 3.0 als "CP/M-Stick". Bitte wie immer kritisch durchsehen.

Joe

Autor: Joe G. ([feinmechaniker](#))

Datum: 11.07.2011 16:51

---

@Hans-Werner

Diese Variante kann über USB mit 5V versorgt werden oder mit 3.3V über den Steckverbinder für Peters Adapter. Die interne 3.3V Versorgung wird bei der Bestückung mit dem [FT232](#) wieder über den internen Regler realisiert. Ansonsten entspricht die Schaltung fast der ursprünglichen Version bis auf die Datenbreite von 8-Bit.

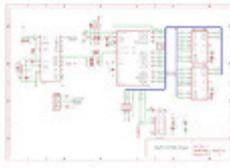
Der Datenaustausch unter Windows sollte mit der FAT-Version kein Problem sein. Die MMC-Karten können damit nur über Windows beschrieben werden. Linux ist nicht notwendig.

Joe

Autor: Joe G. ([feinmechaniker](#))

Datum: 16.07.2011 12:39

Angehängte Dateien:



- [avrcp\\_m.pdf](#) | [anzeigen](#)  
16,7 KB, 131 Downloads

---

[GPS'ler schrieb:](#)

> Für mich wäre wünschenswert, dass man die  
> 3,3V von extern einspeisen könnte.

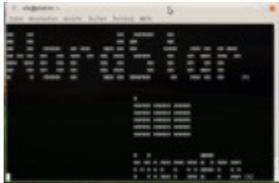
Wenn der [FT232](#) nicht bestückt wird, können die 3.3V auch über JP3 Pin 1 eingespeist werden. Das ist für alle Anwendungen, die einen anderen USB Adapter verwenden wollen. Wird der [FT232](#) selbst bestückt und man möchte die 3.3V über JP3 Pin 1 einspeisen muß die Verbindung zum Regler des [FT232](#) aufgetrennt werden. Das habe ich nun mit JP1 realisiert. Erfolgt die Stromversorgung über USB, kann statt JP1 eine Brücke eingelötet werden. Dank Leo C. haben wir ja immernoch die beiden I2C Leitungen des [AVR](#) frei. Auch wenn es noch keine Anwendung dazu gibt, habe ich sie nun auf JP5 gelegt. Paltz war ja noch auf der Leiterplatte.

Joe

**Autor:** Ole Reinhardt (Firma: Thermotemp GmbH) ([olereinhardt](#))

**Datum:** 21.07.2011 17:58

**Angehängte Dateien:**



- [cpm-ws-start.png](#)  
124,5 KB, 326 Downloads

---

Hi Leute,

Das Projekt ist einsame Spitze! Faszinierend, was man mit einem [AVR](#) alles anstellen kann.

Ich wollt Euch bei dieser Gelegenheit ein ähnliches Projekt vorstellen, welches ich gerade umgesetzt habe.

[Beitrag "Intel 8080 CP/M Emulator für Mikrocontroller \(ARM7, ARM9, CortexM3, AVR32\)"](#)

In diesem Fall laufen sogar zwei Betriebssysteme auf dem Mikrocontroller. Nut/OS (kleines Mikrocontroller Echtzeit Betriebssystem) und CP/M.

Nut/OS dient dabei lediglich als "Wirtsplattform" um z.B. Zugriffe auf die SD Karte zu ermöglichen.

Das ganze läuft auf allen von Nut/OS unterstützten Plattformen mit mind. 80 KByte RAM (ARM7, ARM9, Cortex M3, [AVR32](#), ...)

Viele Grüße,

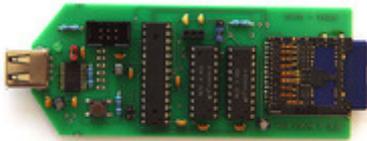
Ole

PS: Anbei ein screenshot des laufenden CP/M Systems mit WordStar Startscreen.  
(Ausgaben über serielle Schnittstelle im minicom dargestellt.)

Autor: Joe G. ([feinmechaniker](#))

Datum: 06.08.2011 15:49

Angehängte Dateien:



• [CP M Stick V3.jpg](#)

64,8 KB, 354 Downloads

---

Aufbauhinweise V3

Die Leiterplatten sollten bald bei euch eintreffen. Deshalb für alle, die neu dabei sind und natürlich auch für alle anderen, einige Aufbauhinweise.

1. Wenn die 5V aus der USB-Schnittstelle bezogen werden dann muss JP1 gesteckt sein. Dabei wird der interne 3.3V Regler des [FT232](#) genutzt.
2. Wenn die 5V über JP3 zugeführt werden, JP1 offen lassen.
3. Wird der [AVR](#) über einen externen Quarz getaktet, Pin2 und Pin3 bei JP2 brücken.
4. Erhält der [AVR](#) seinen Takt über den [FT232](#), dann Pin1 und Pin2 bei JP2 brücken. Der [FT232](#) muss jedoch vorher über ein Tool, welches von der FTDI Webseite bezogen werden kann, programmiert werden. CBUS0 muss dann auf Taktausgabe geschaltet werden.
5. ACHTUNG! Wer den [AVR](#) über die ISP Schnittstelle mit externen 5V programmiert MUSS vorher die SD-Card ziehen, sonst liegen die 5V an der SD-Card! Das übersteht sie nicht.

Viel Spaß beim Aufbau!  
Joe

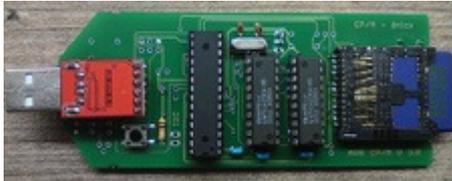
Autor: Peter Sieg ([petersieg](#))

Datum: 08.08.2011 19:04

Angehängte Dateien:



- [Bestueckung.png](#)  
6,6 KB, 272 Downloads



- [cpm\\_stick.jpg](#)  
172,3 KB, 300 Downloads



- [8-bit\\_Meldung.jpg](#)  
61,7 KB, 365 Downloads

So hier Bilder meiner ersten Platine.

Läuft alles!

Was ich festgestellt habe:

Das größte Problem sitzt meistens vor dem Rechner ;-)) Hatte TTL-USB Adapter falsch herum angeschlossen. ;-))

Die FAT16 Routinen sind wählerisch was die Karten angehen. 3 gehen nicht; 2 gehen (1GB Platinen; Noname). Hier gibt es einige Parallelprojekte (sd2iec) wo man ggf. schauen kann, ob man Code hierzu übernehmen kann..

Ach ja.. Speed = 2,623 bei 8-bit und 30MHz (5,293 bei 30MHz und 4-bit).

Peter

Autor: Nils Eilers ([yetanotheruser](#))

Datum: 19.08.2011 00:51

Angehängte Dateien:



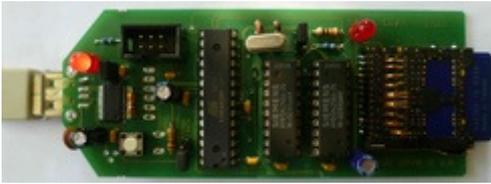
- [AVRCPMADM3A.jpg](#)  
39,3 KB, 414 Downloads

Screenshot vom ADM-3A

Autor: Günter Jung ([gjung](#))

Datum: 21.08.2011 17:16

Angehängte Dateien:



- [avr\\_cpm\\_bild1.jpg](#)  
170,2 KB, 373 Downloads



- [avr\\_cpm\\_screen.jpg](#)  
52 KB, 345 Downloads

---

So, hier die versprochenen Fotos.

Wie gesagt habe ich eine USB Buchse auf der Rückseite,  
die Kratzer am USB Anschluß kommen vom Entlöten.

Aktuell rennt hier ein ATmega168 bei 20MHz.

Gruß,  
Günter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 21.08.2011 17:37

---

Toll Günther!

Ich sehe irgendwann schon das Turbo Pascal laufen.

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 22.08.2011 22:38

---

Wer möchte, der 'Erfinder' des Ganzen = sprite tm hat einen PayPal  
Spendenbutton auf seiner Seite (Tip Jar):

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

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 24.08.2011 11:23

---

Unter: [http://www.mikrocontroller.net/articles/AVR\\_CP/M](http://www.mikrocontroller.net/articles/AVR_CP/M)  
stehen nun die aktuellen Layout Daten, eine Hexdatei zum Soforttest,  
CP/M Image zum Testen, sowie der Link zur aktuellen Software (Dank an  
Leo C.)

Offene Baustellen im Projekt:

1. Z80 Emulation
2. Breakfunktion (Wunsch von Nils)
3. SD-Card Funktion verbessern (es gibt einige Karten, die nicht korrekt funktionieren)

Überraschenderweise waren die knapp 30 Platinen sofort weg. Es gibt schon wieder Anfragen nach neuen Leiterplatten. Ich würde ab sofort wieder Interessenten sammeln. Natürlich sind jederzeit Wünsche, Änderungen, Erweiterungen willkommen (Peter Resettrennung ist vorgemerkt)

Joe

Autor: Günter Jung ([gjung](#))

Datum: 24.08.2011 14:57

---

Hallo,

Ich würde gerne dem Thema Z80 Emulation ein bisschen Leben einhauchen.

Horst hat, wenn ich es richtig mitbekommen habe, das Thema an den Edgar weitergegeben.

@Edgar, falls Du hier noch Hilfe benötigst würde ich mich gerne anbieten.

Gruß,  
Günter

Autor: Nils Eilers ([yetanotheruser](#))

Datum: 24.08.2011 18:48

Angehängte Dateien:



- [AVRCPM 3.jpg](#)  
56,8 KB, 366 Downloads



- [AVRCPM 6.jpg](#)  
84 KB, 369 Downloads

---

So, Hardware ist jetzt erst mal fertig und in ein preiswertes Gehäuse eingebaut ([Reichelt](#) SD 10 SW).

Die Pegelwandlung am [MAX232](#) von 5V nach 3V funktioniert prima mit einem simplen Spannungsteiler aus einem 1,8K und 3,3K Widerstand. Die Stromversorgung erfolgt über ein 5V-Steckernetzteil und [1N5818](#) Schottky-Diode als Verpolschutz. Neben der Reset-Taste gibt's noch eine [LED](#) für den SD-Kartenzugriff. Den Vorwiderstand habe ich auf 55 Ohm

verkleinert, damit sie gut hell leuchtet. Prima wäre noch, die Power-LED und die Zugriffs-LED in einer Dual-LED zu kombinieren, aber da ich nur eine Dual-LED mit gemeinsamer Kathode hatte, hätte das einen Transistor als Inverter erfordert - vielleicht flicke ich den ja mal noch ein ;-)

An den I2C-Bus habe ich noch eine batteriegepufferte Echtzeituhr [PCF8583](#) gehangen - mal sehen, ob und wann die Software das unterstützen kann. Das wäre eine weitere Baustelle: den I2C-Bus dem emulierten 8080 als I/O-Ports zur Verfügung stellen: vielleicht so, dass man in eine I/O-Zelle die Adresse des I2C-Busses schreibt und dann über eine andere den tatsächlichen Zugriff vornimmt.

**Autor: Joe G. ([feinmechaniker](#))** ☰

**Datum: 25.08.2011 16:35**

---

Thema Echtzeituhr:

Von CP/M 2.2 wohl nicht unterstützt, könnte man sich vielleicht trotzdem an die BDOS Funktionen 104 und 105 halten.

BDOS Funktion 104 - Datum und Uhrzeit stellen  
Aufruf mit C=68h, DE=Adresse des Zeitstempels.

Der Zeitstempel hat folgendes Format:

DW Tag ;Tag 1 ist 1.Januar 1978  
DB Stunde ;gepacktes BCD  
DB Minute ;gepacktes BCD

BDOS Funktion 105 - Datum und Uhrzeit holen  
Aufruf mit C=69h, DE=Adresse des Zeitstempels. Rückgabe A=Sekunden (gepacktes BCD).

Der Zeitstempel hat folgendes Format:

DW Tag ;Tag 1 ist 1.Januar 1978  
DB Stunde ;Gepacktes BCD  
DB Minute ;Gepacktes BCD

**Autor: Peter Sieg ([petersieg](#))**

**Datum: 12.09.2011 09:53**

---

@alle:

Ich habe mich nun durchgerungen nochmals Platinen in China zu bestellen. Um den Preis von 10€/Platine halten zu können, mußte ich allerdings einige mehr abnehmen, sodaß wir wahrscheinlich auch noch Weihnachten genug davon haben ;-). Reset zwischen USB TTL Wandler + [AVR](#) habe ich entfernt.

Als Abwechslung habe ich die Platinen mit weißem Lötstoplack und schwarzer Beschriftung gewählt. Das sollte ganz nett (nerd) mit den schwarzen IC's und dem roten TTL-USB Wandler aussehen ;-)

Sollten so in 14 Tagen da sein. SD Slots würde ich dann auch besorgen, sodaß 1 Platine mit Slot dann 12,50€ und Versand dann 2,50€ wären.

Wer welche möchte, schicke mir eine Mail mit Anzahl und Lieferadresse und derjenige bekommt dann meine Bankverbindung zur Vorkasse.

Peter

# Weiße Platinen

Autor: Peter Sieg ([petersieg](#))

Datum: 26.09.2011 20:32

Angehängte Dateien:



• [avrcpmstick\\_weiss.jpg](#)

347,2 KB, 399 Downloads

---

Platinen sind heute angekommen.. und erste läuft schon ;-)  
Bestellte Platinen+Slots gehen morgen zur Post..

Wer übrigens Lust hat.. am 1+2 Okt. findet die Classic Computing in  
37603 Holzminden statt (Infos: [www.classic-computing.de](http://www.classic-computing.de)). Ich werde mit  
dem Stick  
auch dort sein..

Peter

Datum: 10.10.2011 18:49

---

Hier scheint es auch noch Rams zu geben:

<http://www.demotronic.net/>

1,50€/Stück ist ok.

---

Ansonsten. Ist ein wenig ruhig geworden..?  
Was macht die Z80 Implementierfraktion..?

Aber auch ohne Programmierkenntnisse, kann man z.B weitere Programme aus  
CP/M Archiven ausprobieren und in einem neuen Diskimage zusammen  
stellen..

Peter

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 11.10.2011 01:15

---

Einen Z80 Emulator könnte ich demnächst zur Verfügung stellen.  
Allerdings braucht der mindestens ca. 17K Flash, so dass nur der Mega328  
in Frage käme. Dafür ist er aber recht flott. Für "Leos Schleife"  
braucht er im FAST Mode (ZX81 Emulator bei 20MHz [AVR](#)-Takt und 12 Takten  
für einen Speicherzugriff) ca. 0,8 Sekunden, was etwa 3,6 MHz Z80 Takt  
entsprechen würde. Eine auf Speed optimierte Variante (braucht ca. 24K)  
kommt sogar auf 0,7s was dann etwa 4,1MHz Z80-Takt wären (falls ich mich  
da jetzt nicht verrechnet habe).

Jörg

Autor: Peter Sieg ([petersieg](#))

Datum: 11.10.2011 11:46

---

@Jörg: Hört sich gut an! Bei den bisherigen Arbeiten zur Z80 Erweiterung hatte ich den Eindruck, das das aber erheblich kleiner geht mit dem Tabellenansatz.. so in Richtung +5-8kb.. arbeitet da z.Z eigentlich noch jemand hier daran..?

Ein 328 hatte ich mal versuchsweise geflasht.. der schien aber mit den 3,3V an Spannungsversorgung nicht zurecht zu kommen..?

BTW: Es sind noch Platinen+SD Slots zu haben! -> Mail.

Peter

Autor: Joerg Wolfram ([joergwolfram](#))

Datum: 11.10.2011 12:43

---

Über 5K belegen bei mir schon allein die Sprungtabellen:

Ohne Prefix:	512 Bytes
CB Prefix :	512 Bytes
ED Prefix:	512 Bytes
DD Prefix:	1024 Bytes
DD CB Prefix:	1024 Bytes
FD Prefix:	1024 Bytes
FD CB Prefix:	1024 Bytes

Was viel gegenüber meiner "alten Version" an Geschwindigkeit gebracht hat, ist zum einen die vollständige Ausdekodierung aller Befehle und die Nutzung des [AVR](#) Statusregisters. Außerdem habe ich es aufgegeben das originale Befehlstiming möglichst genau nachzubilden. Bei vielen Befehlen ist das Verhalten von [AVR](#) und Z80 gleich, da reicht nach der Operation ein

IN reg\_f, SREG

aus, für logische und Schiebeoperationen nutze ich Tabellen. Manchmal muß man das C Flag über die Operation mittels des T-Flags "retten". Und das N-Flag habe ich nach GPIOR0.0 verlegt. Bei bedingten Sprüngen etc. nutze ich dann die Positionen der [AVR](#) Flags. Bei den meisten Programmen sollte das auch keine Probleme bereiten.

Am [AVR](#)-CP/M Projekt selbst werde ich mich nicht direkt beteiligen, dafür habe ich zu viele andere "Baustellen". Aber das "AX81"-Projekt wird hoffentlich nächste Woche online gehen und damit auch der Sourcecode vom Z80 Emulator.

Gruß Jörg

Autor: Peter Sieg ([petersieg](#))

Datum: 11.10.2011 20:38

---

@Jörg: Ein ZX81 auf [AVR](#) Basis.. freue mich schon!

Peter

Autor: Peter Sieg (Gast)  
Datum: 02.11.2011 22:48

---

Der AX81 von Jörg ist da:

<http://www.jcwolfram.de/projekte/avr/ax81/main.php>

Peter

## Multiplan und Algol

Autor: Peter Sieg (Gast)  
Datum: 03.11.2011 16:32  
Angehängte Dateien:

- [CPMDSK\\_E\\_Multiplan.zip](#) (107,6 KB, 34 Downloads)

---

Na, damit sich auch mal wieder was zum Thema [AVR](#) CPM tut, hier ein Diskimage mit Microsoft Multiplan V 1.2. Entzippen und nach CPMDSK\_E.DSK umbenennen. (Anstatt E auch ein anderes virtuelles Laufwerk je nach Geschmack).

Vorher 1x Install.com ausführen und bei mir auf VT52 = Nr. 23 einstellen.

Peter

Autor: Peter Sieg ([petersieg](#))  
Datum: 06.11.2011 12:08  
Angehängte Dateien:

- [CPMDSK\\_F\\_ALGOL.zip](#) (26,7 KB, 26 Downloads)

---

Hier ALGOL Compiler (P-Code) und Interpreter.  
ALGOLM name.ALG -> Übersetzt nach name.AIN  
RUNALG name.AIN führt aus.  
2 Beispiele dabei.  
Hanoi läuft einwandfrei.  
Lunar läuft? Kommt nach Eingabe der Zahl nicht zurück..? unter Altairz80 ok.

---

Ich glaube, wir habe evtl. noch einen, kleinen Fehler in der 8080 Emulation..  
Mal Multiplan starten in in r1c1 12, in r2c1 24 und in r3c1 =r1c1+r2c1 eingeben (Summe aus 12+24). Dann r1c1 auf 17 ändern.. ;-)

---

Hier mal wie ich solche Diskimages zusammen stelle.  
Altairz80 herunterladen und CPU = 8080 einstellen!. cpm2 laden.  
Diskimage z.B von retroarchive herunterladen und testen.  
Wenn sie laufen, die Dateien mit w aus dem Image auf Filesystem schreiben.  
Mit cpmcp dann in ein leeres avrcpm Image kopieren.  
Image auf SD Karte kopieren und ausprobieren.

---

Arbeitet eigentlich jemand noch/wieder an der Z80 Unterstützung, jetzt da ein Z80 Emulator im Source (AX81) verfügbar ist..?

Peter

## VGA-Term mit P8X32A

Autor: Joe G. ([feinmechaniker](#))

Datum: 21.11.2011 18:41

Angehängte Dateien:



[terminal.jpg](#)

67,2 KB, 396 Downloads

---

[Joe G. schrieb:](#)

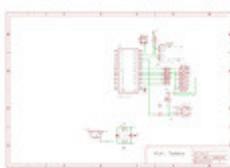
> Ich bin gerade an einer neuen VGA-Variante als VT100 Terminal.

Hier nun die ersten Ergebnisse. Das VT100 Terminal läuft vollständig ohne PC mit der [AVR](#) CP/M Platine. Für VGA und PS-2 werden genau zwei IC's verwendet (1 x P8X32A und 1 x AT24C512 als Speicher). Läuft alles auf 3.3V, somit sind auch keine Pegelwandler notwendig.

Autor: Joe G. ([feinmechaniker](#))

Datum: 22.11.2011 15:10

Angehängte Dateien:



[vga.pdf](#) | [anzeigen](#)

12,4 KB, 87 Downloads

---

Hier die Schaltung der VGA und Tastatursteuerung.

Autor: pollmull (Gast)

Datum: 22.11.2011 20:32

---

[Joe G. schrieb:](#)

> Hier die Schaltung der VGA und Tastatursteuerung.

Oder hier mit gleichem Chip Terminal gleich noch mit CP/M drin:

<http://forums.parallax.com/showthread.php?117996-D...>

Autor: Peter Sieg ([petersieg](#))

Datum: 25.11.2011 13:16

---

Oder [PIC32](#) mit z80sim:

<http://www.kenseglerdesigns.com/cms/node/8>

Peter

Autor: Peter Sieg ([petersieg](#))

Datum: 01.12.2011 18:32

---

Da nun bald Weihnachten wird, gebe ich einige, restliche Platinen des [AVR](#) CP/M USB Sticks unter SK ab für 6,50€/Stück. Habe noch 7 Stück da. Es gibt NUR die Platinen!

Bei Interesse per Mail melden bei: peter (dot) siegl (at) gmx.de

Versand per Post für 2€ in D egal wie viele Platinen.

Peter

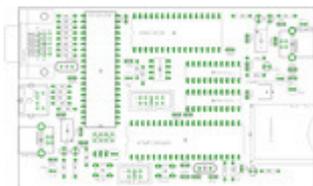
## Neue Platine V4

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 23.12.2011 13:23

Angehängte Dateien:

- [avrcp\\_m V4.zip](#) (114,2 KB, 52 Downloads)



- [AVR\\_CPM.png](#)  
12,7 KB, 267 Downloads

---

Hier etwas für die Feiertage.

Die Version 4.0 mit VGA auf einer Platine.

Änderungen gegenüber der alten Version:

- VGA mit VT100 Terminal auf einem Propellerchip
- SRAM
- ATMEGA 644
- zwei serielle Schnittstellen
- 8 Bit -E/A
- Stromversorgung über USB

USB1 ist für die Inbetriebnahme und Experimente gedacht. Über diese Schnittstelle kann der Propeller geladen werden und die CP/M Ausgaben über ein Terminal mitgelesen werden. Werden die Steckbrücken (JP1 und JP2) richtig gesteckt, dann ist das CP/M-System direkt mit dem VT100 Terminal verbunden.

USB2 ist für alle Anwendungen frei. Denkbar ist z.B. ein Bootloader für den [AVR](#). Damit muss bei der Softwareentwicklung der [AVR](#) nicht jedes Mal über die ISP-Schnittstelle geflasht werden.

Der DRAM wurde durch einen handelsüblichen SRAM ersetzt. Damit sind jedoch zwei zusätzliche Latch notwendig geworden.

Durch den Wechsel auf den ATMEGA 644 ist nicht nur noch eine weitere serielle Schnittstelle dazu gekommen, sondern auch noch ein komplett freier 8-Bit Port. Belegt habe ich PA0 bis PA7, damit können die Pins auch als 8-fach ADC verwendet werden.

Die Software muß natürlich noch angepasst werden. VGA und VT100 Terminal läuft schon.

Schöne Weihnachten!  
Joe

Autor: Frank P. ([mauz](#))

Datum: 23.12.2011 21:20

---

Ich weiß nicht, wurden am Beginn des Projektes nicht alle, die mit SRAM kamen, mit den Worten: "Das ist dann ganz was anderes, mach eine eigenes Projekt auf!", fortgeschickt? Ich meine der Reiz war ein CP/M auf zwei Chips. Das haben wir mit 2xRAM für 8-bit schon aufgeweicht. Noch ein Chip sind dann doch etwas über einer großzügig ausgelegten Definition von 2 Chips, meiner Meinung. Außerdem frage ich mich ob eine zusätzliche, inkompatible HW-Version Softwareentwickler anlocken kann. Da müssten dann langsam HW-Abstraktionsschichten eingezogen werden ;-)  
Das bindet doch nur Ressourcen, die sowieso nicht da zu sein scheinen. Ich hätte lieber Z80 als HW-V4.  
Nur meine bescheidene Meinung.  
frank

Autor: Hans- w. Schütz ([hschuetz](#))

Datum: 23.12.2011 21:33

---

Hallo Frank,  
endlich mal eine Meinung der ich mich anschliessen kann...  
Nur leider bin ich nicht der richtige für Software, aber Hardware haben wir genug. Ein CPM im Propeller mit VGA etc. gibt es schon und passt nicht zum [AVR](#). Ich finde die Kombination die du da gebaut hast, als Variante sehr gut. Aber leider lassen die Softwarefreaks hier auf sich warten...Z80 ist defenitiv überfällig.  
Gruß  
Hans-Werner

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 24.12.2011 10:08

---

@Frank&Hans-Werner

Sicher sind eure Argumente nicht ganz vom Tisch zu fegen. Für jedes Projekt gibt es Vor- und Nachteile. Der Propeller sollte auch kein CP/M emulieren sondern nur als Tastatur und VGA Ausgabe fungieren. Das kann er gut - das macht er gut.

Meine Motivation war die Folgende.

- nicht mehr so einfach beschaffbare DRAM's verhindern einen breiten Nutzerkreis
- keine weiteren freien Pins am [AVR](#) um sich die Welt nach außen zu öffnen
- keine serielle Schnittstelle wenn ein Terminal angeschlossen ist

Nun könnte man auch einen größeren [AVR](#) nehmen ohne gleich auf SRAM zu wechseln. Aber auch das hätte eine angepasste Softwareversion zu Folge. Da die RAM-Ansteuerung schon ausgelagert ist, kann sie auch für den statischen RAM angepasst werden.

Es war einfach ein Hardwarevorschlag von mir. Diesem Vorschlag braucht keiner zu folgen. Mir hat das Arbeiten daran Freude bereitet - einzig das zählt!

Schöne Weihnachten!

Joe

Autor: Frank P. ([mauz](#))

Datum: 24.12.2011 12:51

---

[Joe G. schrieb:](#)

> Es war einfach ein Hardwarevorschlag von mir. Diesem Vorschlag braucht  
> keiner zu folgen.

Das ist schon klar und wir wollen auch nur gepflegt Vor- und Nachteile diskutieren.

> Mir hat das Arbeiten daran Freude bereitet - einzig  
> das zählt!

Das sei Dir auch gegönnt.

Dir auch schöne Weihnachten!

frank

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 27.12.2011 10:57

---

[Frank P. schrieb:](#)

> Das ist schon klar und wir wollen auch nur gepflegt Vor- und Nachteile  
> diskutieren.

Wie wir sehen ist die Resonanz auch sehr gering. CP/M ist halt nicht AdroidGalaxis500. Da mein ursprünglicher Hardwarevorschlag auf eine vereinfachte Bauteilbeschaffung ausgerichtet war, werde ich diese Variante vorerst zurückstellen. Der ATMEGA 644 bekommt nun wieder seine beiden üblichen DRAM's. Die Idee der zwei seriellen Schnittstellen, eine für den Bootloader, werde ich beibehalten. So passiert es nicht mehr, dass die 5V ISP noch im [AVR](#)-Board stecken und die SD-Card nicht gezogen ist.

Joe

Autor: Peter Sieg (Gast)

Datum: 28.12.2011 13:27

---

@Joe: Ich sehe es (leider) auch so wie einige Vorredner.. Hardware haben wir z.Z genug.. ich habe immer noch 5? Platinen des USB Sticks hier.. Rams auch..

Auch gefällt mir! die Kombination aus [AVR](#)+Propeller nicht.. der Prop. kann dann schon alles alleine machen und Z80, CP/M gibts dafür auch schon.. (sogar in Streichholzschachtelgröße)

Was das Projekt wirklich voran bringen könnte wäre Z80 Emulation und dann die darauf aufbauenden Diskimages mit Turbo Pascal etc. etc. Die 8080 Emu scheint aber auch noch mind. einen Bug zu haben.. siehe Fehler im Multiplan..

Was auch schön wäre, eine altairz80 Emu, der direkt die Diskimages für dieses Projekt verarbeiten kann..

Und weitere Diskimages die jetzt schon unter 8080 laufen..

Was ist ansonsten ggf. noch 'niedlich' finden würde wäre ein Stick in SMD Bauweise mit Micro-SD Karte.. und Drams's die sich leicht beschaffen lassen in SMD.. ist aber nur so ein 'Hirngespinnst'..

Z80 Emu. in ASM gibt ja nun Dank Jörg Wolfram im AX81 Projekt.. Basis wäre also ja vorhanden..

Vielleicht sollten wir eine Bounty Prämie ausloben für eine funktionierende Z80 Emu für dieses Projekt..?

Euch allen die besten Wünsche für 2012!

Peter

## Endlich: Z80 Emulation – es geht weiter..

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.12.2011 16:48

Angehängte Dateien:

- [8080int-jmp.asm](#) (58,8 KB, 82 Downloads) | [Codeansicht](#)

---

[Peter Sieg schrieb:](#)

> Z80 Emu. in ASM gibt ja nun Dank Jörg Wolfram im AX81 Projekt.. Basis  
> wäre also ja vorhanden..

In der Quelle 8080int-jmp.asm ist ja auch schon alles vorbereitet. Es müssen (nur) noch die Z80 spezifischen Codes nachgetragen werden. Ich habe mal in der To\_Tabelle die Befehle mit (Z80) markiert. Dabei viel mir auch auf, dass der OpCode D9 falsch interpretiert wurde. Laut 8080 sollte das RET sein, ausgeführt wurde NOP und im Z80 ist es EXX. Ich habe es korrigiert (siehe Anhang).

Joe

Autor: Peter Sieg (Gast)

Datum: 29.12.2011 10:33

---

Prima! Danke für die Korrektur!

Hast du mal probiert, ob der ominöse Multiplanfehler nun weg ist..?

Gruß Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 29.12.2011 17:23

---

Peter Sieg schrieb:

> Hast du mal probiert, ob der ominöse Multiplanfehler nun weg ist..?  
Nein noch nicht.

@alle

Schraubt schon jemand an der Z80 Umsetzung? Wenn ja, bitte kurz melden damit wir nicht doppelt schrauben ;-)

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 30.12.2011 14:50

---

Der Fehler im Multiplan ist immer noch..

Einfach mal in Zelle r2c2 23 und in r3c2 4 und in r4c2 r2c2+r3c2 eintragen.

Ergibt ersteinmal richtig 27. Dann r3c2 von 4 auf 9 ändern .. :-(

Peter

Autor: Leo C. ([rapid](#))

Datum: 30.12.2011 18:06

---

Joe G. schrieb:

> In der Quelle 8080int-jmp.asm ist ja auch schon alles vorbereitet. Es  
> müssen (nur) noch die Z80 spezifischen Codes nachgetragen werden.

Dieses "nur" kann aber recht aufwendig werden, insbesondere bei den Indexregister-Befehlen (DD/FD-Prefixe). Ich wünsche Dir und uns viel Erfolg. Kennst Du diese Seite hier?

<http://z80.info/decoding.htm>

Da könnten ein paar nützliche Hinweise drin sein.

> habe mal in der To\_Tabelle die Befehle mit (Z80) markiert. Dabei viel  
> mir auch auf, dass der OpCode D9 falsch interpretiert wurde. Laut 8080  
> sollte das RET sein, ausgeführt wurde NOP und im Z80 ist es EXX. Ich  
> habe es korrigiert (siehe Anhang).

Dir Korrektur ist richtig (Danke), aber die Beschreibung nicht ganz. D9 ist bei 8080 ein undefinierter Opcode (RET ist C9). Meines Wissens hat der 8080 im Gegensatz zu den meisten anderen Prozessoren auch keine nennenswerten undokumentierten Befehle an solchen Stellen. Der Fehler dürfte sich daher in keinem Programm bemerkbar machen.

Joe G. schrieb:

> Bist du dir sicher, dass Multiplan unter dem 8080 läuft?

Ganz sicher. Damit habe ich Mitte der 80er die Telefonabrechnung für unser Stockwerk im Studentenwohnheim gemacht (10 Apartments und ein Telefon mit Einheitszähler auf dem Flur).

Nächstes Jahr werde ich mir den Fehler mal anschauen.

> Möglicherweise  
> ist die unterschiedliche Parity-Flag Behandlung die Ursache. Setze doch  
> mal EM\_Z80 auf 1, dann werden die Z80 Flags ausgeführt.

Ich kann mir nicht vorstellen, daß es ein Programm gibt, daß mit dem 8080-Befehlssatz auskommt, die Flags aber wie beim Z80 interpretiert.

**Autor:** Leo C. ([rapid](#))

**Datum:** 30.12.2011 19:01

---

[Peter Sieg schrieb:](#)

> Was das Projekt wirklich voran bringen könnte wäre Z80 Emulation und  
> dann die darauf aufbauenden Diskimages mit Turbo Pascal etc. etc.

Turbo Pascal wäre sicher nett, aber wichtiger finde ich, das das bestehende Minimalsystem zu verbessern, so das es richtig rund läuft. Dazu stehen auf meiner ToDo-Liste noch:

- Notwendige und nützliche Tools auf PC als Paket zusammenstellen ("Entwicklungsumgebung") und insbesondere dokumentieren. Möglichst einheitlich für Linux und Windows.
- SD-Kartentreiber verbessern.
- Break erkennen.

Mit dem ersten Punkt hatte ich vor einigen Monaten schon mal angefangen...

> Was auch schön wäre, eine altairz80 Emu, der direkt die Diskimages für  
> dieses Projekt verarbeiten kann..

Zu altairz80 habe ich da  
[Beitrag "Re: CP/M auf ATmega88"](#)  
schon was geschrieben.

**Autor:** Markus G. ([thechief](#))

**Datum:** 01.01.2012 15:42

---

Hallo Retro-Freak Kollegen, :-)

wünsche allen hier gut ins neue Jahr 2012 gekommen zu sein!

Dies ist mein erster Beitrag in diesem Thread (habe bisher nur mitgelesen).

Von Peter habe ich vor ein paar Tagen zwei Platinen erstanden, sind gestern auch angekommen, Danke Peter!

Nun muss ich noch die nötigen Bauteile besorgen und das Ganze aufbauen...

...und dann gehts erst richtig los! :-)

Ich würde Euch gerne bei diesem spannenden Projekt unterstützen!

Was ich beitragen kann:

Aus dem Bereich Software(-Entwicklung):

- Etwas angestaubte Kenntnisse in 8080/8085 Assembler von früher...
- Kenntnisse auch in anderen Assembler Dialekten ([6502/68000er](#)) wer weiß wofür es mal gut ist... :-)
- Kenntnisse in C / C++
- Allgemeine Kenntnisse im Bereich Softwareentwicklung (z.B. diverse IDE's, etc.)

Aus dem Bereich Hardware:

- Allgemeine Elektronik-Kenntnisse (bin ausgebildeter Kommunikationselektroniker)
- Kenntnisse im Bereich Mikroprozessor-/Mikrocontroller-Technik
- Messtechnik im eigenen Elektronik-Labor: 2 x Oszilloskop (analog u. digital), 34 Kanal-Logikanalyser, STK500 Entwicklungsboard, etc.

Ich möchte zu meinem Einstieg auch gleich einen Vorschlag machen:

Mir ist beim Lesen aller (ja es sind über 1000!) Beiträge aufgefallen, dass es viele wertvolle Hinweise / Links in diesem Threat gibt, welche aber nur teilweise in der Projektseite:

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

Zum Beispiel aktuell dieser: <http://z80.info/decoding.htm>

Ich würde als erstes die Links alle herausuchen und auf der Projektseite im Breich "Siehe auch" einfügen.

Des weiteren würde ich vorschlagen, auf der Projektseite auch einen Punkt "ToDo" einzuführen um dort anstehende Themen wie z.B.: Erweiterung um Z80 Emulation aufzuführen evtl. mit Hinweis wer an welchem Punkt bereits arbeitet!

Was haltet Ihr davon?

Viele Grüße

Markus

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 02.01.2012 08:38

---

@Markus

Willkommen bei den Retroinfizierten!

Wie du schon mitbekommen hast, lebt so ein Projekt vom Enthusiasmus der Beteiligten. Somit ist jede Mitarbeit willkommen. Sei es bei der Dokumentation der Hardware oder der Software. Dein Vorschlag, die Projektseite zu aktualisieren ist gut. Eine Linksammlung fehlte tatsächlich noch darauf. Auch eine ToDo-Sammlung ist eine gutes demokratisches Mittel „wirkliche“ Bedürfnisse zu erfassen sowie den aktuellen Stand zu dokumentieren.

Darf ich dir gleich einen konkreten Vorschlag unterbreiten? Da eine Aufbaudokumentation noch fehlt, ist es aus meiner Sicht ganz nützlich so was in Angriff zu nehmen. Wenn du bei deinem Aufbau die Aufbauschritte und auch die dabei auftretenden Hürden dokumentierst hätten wir gleich einen Anfang. Also einfach kurz alles notieren was bei der Hard- und Software schief gehen kann.

Gruß Joe

Autor: Markus G. ([thechief](#))

Datum: 03.01.2012 03:33

---

Hallo Joe,  
hallo Peter,

Danke für die herzliche Begrüßung!

Die Schlacht hat bereits gegonnen :-), seht hier:

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

Als nächstes werde ich dort noch jede Menge weitere Links einfügen, welche ich aus diesem Threat extrahiert habe!

Dann folgt der Todo Abschnitt, da dürft Ihr Euch dann verausgaben... ;-)

Und wenn ich, voraussichtlich nächste Woche, alle Bauteile zusammen habe, baue ich einen CP/M Stick auf und notiere mir dabei was mir dabei auffällt (wie von Joe vorgeschlagen)!

Viele Grüße

Markus

Autor: Markus G. ([thechief](#))

Datum: 10.02.2012 00:50

Angehängte Dateien:

- [Z80 spezifische Opcodes.ods](#) (16,2 KB, 36 Downloads)
- [Z80 spezifische Opcodes.xls](#) (49,5 KB, 60 Downloads)

---

[Joe G. schrieb:](#)

> [Markus G. schrieb:](#)

>> Die geänderte Assembler Quellcode-Datei habe ich diesem Beitrag  
>> angehängen.

>

> Super, läuft!

>

> Wenn du Lust hast auch am Z80 Code zu arbeiten....

>

> Die "JR Dinge" habe ich schon eingepflegt, sind aber noch nicht  
> vollständig getestet. CB, ED und FD sind noch offen (viel Fleißarbeit).

>

> Gruß Joe

Hallo Joe,

Danke, freut mich! :-)

Ja, ich habe Lust auch bei der Z80 Emulation zu unterstützen.  
Ich habe mir schon mal das z80int.asm Modul angesehen, kann es aber noch nicht komplett überblicken.

Mir sind dabei vermeintlich ein paar Z80 spezifische Dinge aufgefallen welche ich hier kurz aufzeigen möchte:

1. Wir müssen folgende Opcodes noch implementieren / testen:

```

.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 08      EX AF,AF' (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 10 oo    DJNZ o      (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 18 oo    JR o      (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 20 oo    JR NZ,o    (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 28 oo    JR Z,o    (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 30 oo    JR NC,o   (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; 38 oo    JR C,o   (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; CB      (Z80 specific)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; D9      EXX      (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; DD      (Z80)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; ED      (Z80 specific)
.dw (FETCH_NOP | OP_INV | STORE_NOP) ; FD      (Z80 specific)

```

Dahinter verbergen sich 456(!) einzelne Opcodes, soviel zum Thema Fleißarbeit... :-)

2. Unter den Z80 spezifischen Opcodes gibt es auch einige welche 4 Bytes lang sind.

Diese befinden sich alle in den Bereichen der DD, ED und FD Opcodes!  
Wie wirkt sich dies auf die Opcode-Decoder-Routinen aus? **Grübl**

Ich habe die von mir angefertigte Tabelle (im xls und ods Format) diesem Beitrag angehängt.

Diese Webseite diente mir als Quelle:

<http://nemesis.lonestar.org/computers/tandy/softwa...>

Viele Grüße

Markus

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 10.02.2012 09:49

Markus G. schrieb:

> Ja, ich habe Lust auch bei der Z80 Emulation zu unterstützen.

Der aktuelle Interpreter ist 8080int-jmp.asm

Für die 4 Byte Opcodes habe ich auch noch keinen Plan. Vielleicht fällt Leo C. was dazu ein ;-).

Gut ist auch immer ein Blick auf das Projekt von Jörg Wolfram

<http://www.jcwolfram.de/projekte/avr/ax81/main.php>

Dort hat er schon einen Z80 Interpreter für AVR implementiert. Er kann nicht 1:1 übernommen werden aber der Blick auf einzelne Befehle ist äußerst hilfreich (Danke Jörg!).

Wenn du mir deine PM senden könntest, dann würde ich dir meinen Arbeitsstand von 8080int-jmp.asm zukommen lassen.

Gruß Joe

Autor: Peter Sieg (Gast)

Datum: 10.02.2012 10:36

Sehr schön, das da (noch/wieder) wer dran arbeitet!

Gruß Peter

Autor: Markus G. ([thechief](#))

Datum: 10.02.2012 17:17

---

[Joe G. schrieb:](#)

> [Markus G. schrieb:](#)

>> Ja, ich habe Lust auch bei der Z80 Emulation zu unterstützen.

>

> Der aktuelle Interpreter ist 8080int-jmp.asm

> Für die 4 Byte Opcodes habe ich auch noch keinen Plan. Vielleicht fällt

> Leo C. was dazu ein ;-).

>

> Gut ist auch immer ein Blick auf das Projekt von Jörg Wolfram

> <http://www.jcwolfram.de/projekte/avr/ax81/main.php>

> Dort hat er schon einen Z80 Interpreter für [AVR](#) implementiert. Er kann

> nicht 1:1 übernommen werden aber der Blick auf einzelne Befehle ist

> äußerst hilfreich (Danke Jörg!).

>

> Wenn du mir deine PM senden könntest, dann würde ich dir meinen

> Arbeitsstand von 8080int-jmp.asm zukommen lassen.

>

> Gruß Joe

Hallo Joe,

okay, dann übernehme ich gerne Deinen Arbeitsstand.

Du hast soeben eine PM von mir bekommen mit meiner privaten  
Emailadresse! :-)

Das Projekt von Jörg ist mir auch bekannt, war auch gerade wieder auf  
der Seite und habe mir die Sourcen der V0.26 gezogen und kurz einen  
Blick in den Z80-Emu geworfen!

Was mir dort auf den ersten Blick gut gefallen hat ist die Unterteilung  
der "Opcode Familien" in einzelne Module (Include Dateien).

Ich finde dies macht das Ganze zumindest etwas übersichtlicher...

Viele Grüße

Markus

Autor: Markus G. ([thechief](#))

Datum: 12.02.2012 19:45

Angehängte Dateien:

- [init.asm](#) (7,1 KB, 66 Downloads) | [Codeansicht](#)

---

Hallo zusammen,

ich habe meine, im Beitrag #2541628 veröffentlichte,  
Speichertest-Routine nun noch einmal verfeinert:

Durch den Eintrag:

```
.equ MEMTEST = 3
```

(oder höher) in der Datei: config.inc

ist es nun möglich im Falle eines Speicherfehlers auf einem [AVR](#)  
CP/M-Stick V3.0 den/die defekten DRAM-Chips (IC5 / IC6) anzeigen zu

lassen!

Die geänderte Assembler Quellcode-Datei habe ich diesem Beitrag angehängen.

Um die geänderte Speichertest-Routine zu verwenden sind also nur zwei kleine Schritte nötig:

1. Austausch der vorhandenen Datei `init.asm` im Projektverzeichnis durch die im Anhang befindliche Datei `init.asm`.

2. Aktivieren der neuen Routine durch die Anweisung:

```
.equ MEMTEST = 3
```

(oder höher) in der Datei: `config.inc`

Auch die alten Speichertest-Routinen können weiterhin verwendet werden, hierzu ist wie bisher die Anweisung:

```
.equ MEMTEST = 1 oder 2
```

in der Datei: `config.inc` zu setzen!

Siehe auch folgenden Auszug aus der Datei `init.asm`:

```
-----  
; Info about MEMTEST:  
; 0 = no memtest at all  
; 1 = memtest shows defects but carries on  
; 2 = memtest shows defect and stops on first failure  
; 3 = memtest shows defect, stops on first failure and  
;     shows which RAM chip has to be replaced  
;     on an AVR CP/M-Stick V3.0  
-----
```

Viele Grüße

Markus

PS: @Leo: Du darfst auch die neue Version gerne (wieder) ins [SVN](#) einchecken!

## Der Multiplan Fehler

Autor: Leo C. ([rapid](#))

Datum: 24.02.2012 14:31

---

[Peter Sieg schrieb:](#)

```
> @Leo C.: Konntest du schon mal in das wirre Verhalten von Multiplan  
> schnuppern. Irgendwo muss da doch noch in der 8080 Implementierung ein  
> Wurm drin sein..?
```

Jetzt habe ich es mal mit der Version 1.06 von hier

<http://www.schorn.ch/cpm/intro.php>

ausprobiert. Da taucht der Fehler auch auf. Da der Fehler im `simh` mit 8080-Emulation nicht vorkommt, muß es wohl an unserer CPU-Emulation

liegen.

Ich tippe auf den DAA-Befehl, da Multiplan wahrscheinlich mit BCD-Zahlen rechnet. Vielleicht hat ja zufällig jemand CBASIC von Digital Research parat, das auch in obiger Quelle zu finden ist, und das auch mit BCD rechnet, und kann meine Vermutung bestätigen.

**Autor: Peter Sieg (Gast)**

**Datum: 24.02.2012 15:01**

---

@Leo C.: Habe leider nur MS Basic hier..

Evtl. zeigt ja schon ein Vergleich zw. was 8080-DAA machen soll und das was in unserer Implementierung passiert etwas auffälliges..

BTW: Das erinnert mich an einen Zeitungsartikel in dem es hieß:  
... der letzte Fehler in CP/M wurde auch erst nach 7 Jahren gefunden..

; -)

Haha.. derjenige war ganz sicher Gott oder er hat keine Ahnung von Software..

Peter

**Autor: Markus G. ([thechief](#))**

**Datum: 25.02.2012 13:34**

---

[Peter Sieg schrieb:](#)

```
> @alle: Wie ist denn der Stand bei der Implementierung der Z80
> Unterstützung?
> @Leo C.: Konntest du schon mal in das wirre Verhalten von Multiplan
> schnuppern. Irgendwo muss da doch noch in der 8080 Implementierung ein
> Wurm drin sein..?
>
> Ach Ja.. Platinen sind nun alle.. wenn noch wer hier Bedarf hat.. müßte
> ich dann nachbestellen und das müssen immer mind. 10 Stück sein..
>
> Peter
```

Hallo zusammen,

ich habe zwischenzeitlich von Joe seinen aktuellen Arbeitsstand der Datei: 8080int-jmp.asm für den Z80 Emulator bekommen, Danke Joe!

Bin aber leider noch nicht wirklich dazu gekommen mich dort etwas einzuarbeiten. :-)

Viele Grüße

Markus

**Autor: Markus G. ([thechief](#))**

**Datum: 25.02.2012 13:48**

---

[Leo C. schrieb:](#)

```
> Peter Sieg schrieb:
>> @Leo C.: Konntest du schon mal in das wirre Verhalten von Multiplan
>> schnuppern. Irgendwo muss da doch noch in der 8080 Implementierung ein
>> Wurm drin sein..?
```

>  
> Jetzt habe ich es mal mit der Version 1.06 von hier  
> <http://www.schorn.ch/cpm/intro.php>  
> ausprobiert. Da taucht der Fehler auch auf. Da der Fehler im simh mit  
> 8080-Emulation nicht vorkommt, muß es wohl an unserer CPU-Emulation  
> liegen.  
>  
> Ich tippe auf den DAA-Befehl, da Multiplan wahrscheinlich mit BCD-Zahlen  
> rechnet. Vielleicht hat ja zufällig jemand CBASIC von Digital Research  
> parat, das auch in obiger Quelle zu finden ist, und das auch mit BCD  
> rechnet, und kann meine Vermutung bestätigen.

Hallo zusammen,

ich nehme mich des "Multiplan Fehlers" an!  
@Leo C.: Danke für die Quelle zum CBASIC und den Hinweis auf den DAA  
Befehl, mal sehen was ich herausbekomme.

Diese Bausstelle bringt dann auch gleich mein Verständnis über den  
Prozessablauf des 8080 Befehlsinterpreters voran.  
Dies hilft mir anschließend für die Arbeit am Z80 Interpreter...

Alles ein Frage der richtigen Kombination(en)... ;-)

Viele Grüße

Markus

Autor: Leo C. ([rapid](#))  
Datum: 25.02.2012 14:03  
Angehängte Dateien:

- [8080int-jmp.asm](#) (58,6 KB, 73 Downloads) | [Codeansicht](#)

---

[Markus G. schrieb:](#)

> ich nehme mich des "Multiplan Fehlers" an!

Bin inzwischen einen Schritt weiter:  
Bei den Arithmetikbefehlen wird das H-Flag (half carry) nicht gesetzt.

Ich habe auch mal einen Patch angefangen (Anhang), aber z.Zt. habe ich  
nicht mal einen Assembler auf meinem Rechner installiert.  
Bis ich testen kann, dauert also etwas.

Autor: Peter Sieg (Gast)  
Datum: 25.02.2012 15:50

---

@Leo C / Markus: Prima!

Ich kann etwas mehr zu Rams sagen:  
SIEMENS HYB514256B-70 gehen bei mir definitiv nicht bei 3,3V, 20MHz und  
3 Waitstates. Mehr als 8 Stück getestet - alle liefen nicht.

Aber:  
V53C104P laufen nun bei 3,3V, 20MHz und 3 Waitstates!  
Diese liefen damals unter 30/25MHz, 3,3V und 1 Waitstates NICHT.

Peter

Autor: Leo C. ([rapid](#))

Datum: 25.02.2012 16:01

---

Meine Entwicklungsumgebung geht wieder.  
Der DAA-Code ist total buggy. 2 Fehler habe ich gefunden und dem  
nächsten bin ich auf der Spur. (Wer hat nur so einen Mist programmiert?  
;)

Autor: Peter Sieg (Gast)

Datum: 26.02.2012 12:38

---

Um noch mal ein paar 'Schwesterprojekte' zu zeigen:

Propeller:

<http://forums.parallax.com/showthread.php?116893-R...>

PIC32:

<http://www.kenseglerdesigns.com/cms/node/8>

Das Ramblade in Streichholzschachtelgröße hatte ich schon mal.. dann  
aber verkauft.. habe es mir mal wieder bestellt (40US\$+6US\$ shipping).

Bei dem [PIC32](#) habe ich mir mal ein Duinomite-Mini von Olimex für 30€  
bestellt. Das ist wohl z.Z das Stand-Alone Basic drauf.. aber das sollte  
sich ja ändern lassen..

Trotzdem ist und bleibt unser 'Baby' die [AVR](#) Lösung ;-)

Peter

Autor: Markus G. ([thechief](#))

Datum: 28.02.2012 17:43

---

[Leo C. schrieb:](#)

> [Markus G. schrieb:](#)

>> ich nehme mich des "Multiplan Fehlers" an!

>

> Bin inzwischen einen Schritt weiter:

> Bei den Arithmetikbefehlen wird das H-Flag (half carry) nicht gesetzt.

>

> Ich habe auch mal einen Patch angefangen (Anhang), aber z.Zt. habe ich

> nicht mal einen Assembler auf meinem Rechner installiert.

> Bis ich testen kann, dauert also etwas.

Hallo Leo,

ich sehe Du bist voll in Deinem Element... ;-)

Ich habe mir Deine neue 8080int-jmp.asm geladen, assembliert und damit  
Multiplan 1.06 geteset:

Ich kann Dir sagen, noch ist der Bug da! :-)

Er tritt z.B. bei Additionen mit einem Ergebnis größer gleich 30 auf, 29  
+ 1 = 130 und 29 + 2 = 131 :-)

Es gibt auch nette Subtraktionen: 1 - 8 = -1, 100 - 1 = 109 und 100 - 8  
= 102 :-)

Falls es wieder eine neue Version zum Testen gibt, nur her damit... :-)

Viele Grüße

Markus

Autor: Leo C. ([rapid](#))

Datum: 28.02.2012 18:04

Angehängte Dateien:



- [Bildschirmfoto - 28.02.2012 - 18 04 29.png](#)  
40,3 KB, 123 Downloads

---

[Markus G. schrieb:](#)

> Ich kann Dir sagen, noch ist der Bug da! :-)

Ja, der DAA-Code ist (bzw. war) ein einziger großer Bug. :)

Hast Du auch das hier gelesen?

[Beitrag "Re: CP/M auf ATmega88"](#)

> Falls es wieder eine neue Version zum Testen gibt, nur her damit... :-)

Ich mache DAA gerade komplett neu. Im Moment gibts besonders lustige Ergebnisse (Anhang). Aber vielleicht wirds heute noch fertig.

Autor: Leo C. ([rapid](#))

Datum: 28.02.2012 19:30

Angehängte Dateien:

- [8080int-jmp.asm](#) (58,3 KB, 67 Downloads) | [Codeansicht](#)
- [daa-test.mp](#) (896 Bytes, 25 Downloads)

---

So, für Multiplan reicht's jetzt.  
Ob die Emulation allerdings ganz korrekt ist, weiß ich nicht.  
Außerdem fehlt noch der Fall, wenn das N-Flag gesetzt ist. Das sollte aber für alle korrekten 8080-Programme ausreichen.

Im Anhang ist noch ein Multiplan Testprogramm.

Leider habe ich keine Quellen gefunden, die den DAA Befehl vollständig beschreiben. Für 8080 findet man noch weniger, als für den Z80.  
Auf der Suche habe ich aber noch folgende interessante Dokumente gefunden:

<http://anyplatform.net/cpus.html>  
(weiter unten 8080/Z80 Series)

Weitere habe ich schon wieder verloren...

Autor: Draco (Gast)

Datum: 28.02.2012 19:43

---

Leo C. schrieb:

> Leider habe ich keine Quellen gefunden, die den DAA Befehl vollständig  
> beschreiben.

Schau mal im Buch "Programmierung des Z80" ab Seite 226:

<http://www.wincpc.ch/docs/Programmierung%20des%20Z80.pdf>

Ich gehe mal davon aus, dass der Link legal ist, da Rodney Zaks an  
anderer Stelle der Bereitstellung des Buches zugestimmt hat:

<http://www.z80.info/zaks.html>

Autor: Leo C. ([rapid](#))

Datum: 28.02.2012 20:37

---

Hallo Draco,

danke für den Hinweis. Zaks ist schon eine der besten Referenzen für Z80  
überhaupt, aber leider fehlt auch bei ihm der Zustand des Half-Carry  
nach dem DAA-Befehl. Inzwischen habe ich aber noch ein gutes Dokument  
(wieder-) gefunden:

"The Undocumented Z80 Documented"

<http://www.myquest.nl/z80undocumented/>

(auch auf <http://www.z80.info>) zu finden)

Autor: Markus G. ([thechief](#))

Datum: 29.02.2012 02:41

---

Leo C. schrieb:

> So, für Multiplan reichts jetzt.  
> Ob die Emulation allerdings ganz korrekt ist, weiß ich nicht.  
> Außerdem fehlt noch der Fall, wenn das N-Flag gesetzt ist. Das sollte  
> aber für alle korrekten 8080-Programme ausreichen.  
>  
> Im Anhang ist noch ein Multiplan Testprogramm.  
>  
> Leider habe ich keine Quellen gefunden, die den DAA Befehl vollständig  
> beschreiben. Für 8080 findet man noch weniger, als für den Z80.  
> Auf der Suche habe ich aber noch folgende interessante Dokumente  
> gefunden:  
>  
> <http://anyplatform.net/cpus.html>  
> (weiter unten 8080/Z80 Series)  
>  
> Weitere habe ich schon wieder verloren...

Hallo Leo,

super! Danke!

Ich habe noch ein interessantes "Original-Dokument" gefunden.  
Über den folgenden Link kann man sich das "Intel 8080 Microcomputer  
System's Users Manual" ansehen:

<http://www.scribd.com/doc/19954501/Intel-8080-Manual-1>

Auf der Seite 29 von 40 ist der DAA Befehl beschrieben, die betroffenen

Flags (Fußnote 10) sind auf der Seite 33 von 40 beschrieben:

"If the value of the least significant 4-bits of the accumulator is greater than 9 or if the auxiliary carry bit is set, 6 is added to the accumulator. If the value of the most significant 4-bits of the accumulator is now greater than 9 or if the carry bit is set, 6 is added to the most significant 4-bits of the accumulator."

Viele Grüße

Markus

**Autor: Bernd B. ([mainhattan](#))**

**Datum: 29.02.2012 08:58**

---

Hallo zusammen,

der DAA des 8080 ist einfach, da er nur Additionen korrigiert. Der Wert des H-Flags nach der DAA-Operation ergibt sich aus der Oder-Verknüpfung des vorherigen Wertes von H und aus der Bedingung Lower Nibble > 9. Also  $H = H \text{ or } (\text{Akku and } 0x0f) > 9$ .

Seien a,b Element aus [0..9],  $s = a + b$ . Wenn  $s > 9$ , dann hat ein Überlauf von der Einerstelle auf die Zehnerstelle stattgefunden, also muss H gesetzt werden. Jetzt ist es aber so, dass wir nicht 10, sondern 16 als Basis für eine Stelle haben (Aufspaltung des Bytes in 2 Nibbles), daher wird für s aus [16..18] das H-Flag automatisch gesetzt. Für s aus [10..15] gibt es keinen Überlauf im 16er-System, aber im 10er-System, daher muss das H-Flag über die Bedingung lower nibble > 9 gesetzt werden.

Die gleiche Überlegung trifft auch auf das C-Flag und das Higher Nibble zu, aber man muss hier schon den Überlauf aus der Einerstelle mit einbeziehen. Das ergibt dann  $C = C \text{ or } (\text{Akku} > 0x99)$ .

Und der Korrekturwert ergibt sich aus den Werten der beiden Flags nach den beiden oben genannten Oder-Verknüpfungen:

Wenn  $H = 1$ , dann wird  $0x06$  zum Akku addiert.

Wenn  $C = 1$ , dann wird  $0x60$  zum Akku addiert.

Und wenn beide Flags gesetzt sind, dann ist der Korrekturwert  $0x66$ .

Entsprechende Überlegungen muss man bei der Korrektur für Subtraktionen beim Z80 machen.

**Autor: Gee\_Tee (Gast)**

**Datum: 29.02.2012 11:12**

---

Hallo,

Aus: Z80 Instruction Handbook (Your complete reference to the powerful Z80 instruction set) - SCELBI Publications

DECIMAL ADJUST THE ACCUMULATOR

DAA            047oct                    27hex

The decimal adjust accumulator instruction is a one byte instruction that adjusts the contents of the accumulator to two binary coded-decimal digits, one digit in the four least significant bits and one digit in the four most significant bits. This instruction is used following the addition or subtraction of two pair of BCD digits to adjust the result to the proper BCD code. It may also be used following an increment, decrement, or negate directive. Following an addition operation, this instruction operates in the following manner.

The least significant half of the accumulator is checked for a BCD value

of zero to nine, and the H flag is checked for a zero condition. If both of these conditions exist, this half is left as is.

However, if this half is greater than nine, or the H flag is one, the accumulator will be incremented by six, thereby adjusting the least significant half to the proper BCD code. The most significant half is then checked for a BCD value of zero through nine and the C flag is checked for a zero condition. If both conditions are true, the process is complete. Otherwise, if either condition is not met, the most significant half of the accumulator is incremented by six, with the carry flag being set to a one if an overflow from the most significant half occurs.

In the case of subtraction, the H and C flags are used to determine if the value six must be subtracted from each half of the byte (which occurs if the associated flag is set to a logic one state when the DAA directive is encountered). The Z, S, and P/V flags are affected by these operations (P/V reflects parity status). The N flag is not altered

Viele Grüße

Gerrit

**Autor: Peter Sieg (Gast)**

**Datum: 29.02.2012 18:00**

---

Super!!

Nun arbeitet MP so wie es soll ;-) )

Danke!

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 04.03.2012 16:49**

**Angehängte Dateien:**

- [8080int-jmp.asm](#) (59,7 KB, 64 Downloads) | [Codeansicht](#)
- [8080-vs-Z80-Flags-sort-8080.txt](#) (6,9 KB, 67 Downloads)

---

Vielen Dank für die weiteren Hinweise zum DAA-Befehl.

Wie die Flags gesetzt werden müssen, ist auch in den 8080 Datenbüchern, die ich auch auf meiner Festplatte gefunden habe, beschrieben.

Im Anhang ist mein aktueller Stand für den Interpreter.

DAA ist nur für den 8080 drin. Ich habe nochmal die Behandlung der Flags bei allen relevanten

Befehlen anhand der angehängten Tabelle überprüft, und einige kleine Fehler korrigiert.

Insbesondere INC und DEC sind dadurch leider größer und langsamer geworden.

Zum Ausgleich habe ich die Opcode-Tabellen nochmal etwas komprimiert.

Dies brachte

Platzeinsparungen und Geschwindigkeitssteigerungen um unteren Promillebereich. Insbesondere der

der NOP-Befehl konnte deutlich beschleunigt werden. :-)

Die anderen Interpretervarianten habe ich teilweise nachgezogen. Es stellt sich aber die Frage, ob diese überhaupt noch von Interesse sind (evtl. wg. ROM-Größe). Wenn ja, würde ich aber als nächstes die Dateien umstrukturieren, so daß die Action-Routinen (fetch/op/store) nur einmal vorhanden sind.

Als Übung, oder für Leute mit Zeit/Langeweile, hätte ich jetzt folgende Vorschläge:

1. Testen, testen, testen...
2. Überprüfung, ob die Flags überall richtig behandelt werden. (kann natürlich mit 1. verbunden werden (DDT))
3. INC/DEC sehen so aus, als könnten sie verbessert werden. Da die Befehle häufig vorkommen, würde sich das lohnen. Auch Peters Bench-Schleife ist jetzt langsamer geworden.

## Z80 – weiter geht's..

Autor: Leo C. ([rapid](#))

Datum: 07.03.2012 00:52

Angehängte Dateien:

- [8080int-jmp.asm](#) (67,6 KB, 63 Downloads) | [Codeansicht](#)

---

Ich habe mir mal ein paar Gedanken zum Z80 Interpreter gemacht, und angefangen, die Tabellenmakros entsprechend zu erweitern. Folgendermaßen könnte es klappen:

Für jeden Prefix wird eine neue Tabelle angelegt. Für DD und FD könnte man eine gemeinsame Tabelle verwenden, und beim Einstieg ein Flag entsprechend setzen.

Wir brauchen dann 4 zusätzliche Tabellen: ED, DD/FD, CB, und CBDD/FD. Da jede Tabelle 512 Byte belegt, und für die meisten Befehle nochmal Zwischensprungtabellen dazu kommen, könnte es für den ATmega168 knapp werden.

Für jeden Prefix gibts eine neue OP-Aktion:

### **do\_op\_prefixED:**

```
mem_read_ds z1,z_pc      ;z1 = memReadByte(z_pc)
adiw  z_pcl,1           ;++z_pc
ldi  zh,high(EDjmp)    ;
ijmp
```

### **do\_op\_prefixDD:**

;TODO: Flag für DD setzen

```
mem_read_ds z1,z_pc      ;z1 = memReadByte(z_pc)
adiw  z_pcl,1           ;++z_pc
ldi  zh,high(DDFDjmp)  ;
ijmp
```

### do\_op\_prefixFD:

;TODO: Flag für FD setzen

```
mem_read_ds z1,z_pc      ;z1 = memReadByte(z_pc)
adiw  z_pcl,1           ;++z_pc
ldi  zh,high(DDFDjmp)   ;
ijmp
```

Die Tabellen bekommen einen Namen über ein Makro zugeordnet. Zuerst die schon bestehende Tabelle:

```
opctable opcjmp
```

```
.
.
instr  fetch_nop,  op_prefixED,store_nop ;ED      ;(ED opcode prefix)
.
.
```

Danach die neuen Tabellen. z.B.:

```
opctable EDjmp
```

```
instr  fetch_nop,  op_nop,      store_nop ;00      ;NOP
.
.
instr  fetch_C,    op_IN,       store_B   ;40      ;IN B, (C)
instr  fetch_B,    op_OUT,      store_nop ;41      ;OUT (C),B
instr  fetch_BC,   op_SBCHL,    store_nop ;42      ;SBC HL,BC
instr  fetch_DIR16,op_STBC,     store_nop ;43      ;LD (nn),BC
instr  fetch_nop,  op_NEG,      store_nop ;44      ;NEG
.
.
```

Fetch/Op/Store-Aktionen müssen definiert sein, bevor sie in den Tabellen referenziert werden können. Damit die Sprünge möglichst kurz werden, sollten sie unmittelbar vor der Tabelle, in der sie das erste mal angesprochen werden, platziert werden.

Im Anhang ist ein ausbaufähiges Gerüst nach diesem Schema. Ich kann die Datei ohne Fehler assemblieren, und einen Befehl aus der ED-Tabelle habe ich erfolgreich getestet.

Autor: Konrad S. ([maybe](#))

Datum: 07.03.2012 08:44

---

### Leo C. schrieb:

```
> Für DD und FD könnte
> man eine gemeinsame Tabelle verwenden
```

Wenn mich meine Erinnerung nicht trügt, sind beim Z80 das DD und das FD nur Präfixe, die dafür sorgen, dass der nachfolgende Befehl auf das IX- bzw. IY-Register wirkt, statt auf das HL-Register. Daraus folgen einige undokumentierte Befehle, da alles, was mit HL geht auch mit IX und IY geht, nur ein bisschen langsamer und ein Byte mehr an Code.

Autor: Leo C. ([rapid](#))

Datum: 07.03.2012 09:26

---

Hallo Konrad,

so ist es.

Wenn man allerdings bei jedem Befehl mit H, L oder HL prüft, ob vorher ein DD- oder FD-Prefix war, dürfte das doch zu viel Geschwindigkeit kosten. Aber man kann es natürlich ausprobieren, vor allem, wenn der Platz nicht reicht.

Besonders hilfreich finde ich übrigens "The Undocumented Z80 Documented":

<http://www.myquest.nl/z80undocumented/>

Nicht nur wegen der undokumentierten Befehle.

**Autor: Bernd B. ([mainhattan](#))**

**Datum: 07.03.2012 14:05**

---

Brauchen wir wirklich für die DD/FD-Behandlung eigene, vollständige Sprungtabellen? Ich denke, dass geht besser, wenn man erweiterte fetch\_XX und store\_XX-Routinen benutzt. Also zusätzlich zu fetch\_HL auch noch fetch\_HLIXIY, welches ein Flag beachtet, dass durch die Präfix DD/FD gesetzt wird und nach jeder Ausführung wieder zurückgesetzt wird. Wenn wir z.B. INC HL betrachten, dann ist das fetch\_HL, op\_INCl6 und store\_HL für den i8080. Für Z80 wird daraus folgendes: fetch\_HLIXIY, op\_INCl6, store\_HLIXIY. Bei LD H, (HL) müssen wird daraus dann daraus fetch\_MHLIXIY, op\_nop, store\_H machen. INC L wäre dann fetch\_LXLYL, op\_INC, store\_LXLYL für das undokumentierte Feature auf die einzelnen Bytes der Indexregister zugreifen zu können.

Die Bitbefehle sind leider etwas komplizierter, da hier durch Vorsetzen von DD aus SET 7,A ein SET 7,(IX+d),A wird, also auf jeden Fall der Speicher über IX+d angesprochen wird.

Durch die Einführung von fetch\_HLIXIY dauert die Verarbeitung von Befehlen mit HL länger als die mit BC oder DE. Auch muss man bei der Behandlung des Displacement-Bytes bei fetch\_MHLIXIY und store\_MHLIXIY aufpassen, dass sie einzeln aber auch paarweise vorkommen können [Siehe INC (IX+d); LD (IX+d),A und LD A,(IX+d)].

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)  
**[Re: CP/M auf ATmega88](#)**

**Autor: Markus G. ([thechief](#))**

**Datum: 07.03.2012 23:47**

Angehängte Dateien:

- [Z80 spezifische Opcodes.ods](#) (16,7 KB, 29 Downloads)
- [Z80 spezifische Opcodes.xls](#) (50 KB, 52 Downloads)

---

[Markus G. schrieb:](#)

>  
> Ich habe die von mir angefertigte Tabelle (im xls und ods Format) diesem  
> Beitrag angehängen.  
>  
> Diese Webseite diente mir als Quelle:  
> <http://nemesis.lonestar.org/computers/tandy/softwa...>  
>  
>  
> Viele Grüße  
>  
> Markus

Hallo zusammen,

ich habe in den von mir im Beitrag #2542248 erstellten Tabellen drei versehentlich doppelt vorhandene Z80 Codes (CB01, DD22 und ED6B) gefunden und korrigiert.

Im Anhang die neuen Tabellen.

Viele Grüße

Markus

Autor: Markus G. ([thechief](#))

Datum: 08.03.2012 00:13

---

Hallo zusammen,

zum Thema Sprungtabellen und zusätzlicher Speicherbedarf:

Wie wäre es, wenn wir, um keine Geschwindigkeitsverluste zu haben und nicht von vorne herein zu viel Speicher zu verbrauchen, erst einmal nur alle "offiziellen" Befehlskombinationen implementieren würden?

Wenn wir anschließend wissen wie es mit dem Speicherbedarf (bezogen auf ATmega 168) aussieht, könnten wir dann ggf. die Befehls-Tabellen komplett (inkl. illegaler Opcodes) implementieren oder uns mit der Variante von Bernd B. (mainhattan) weiter beschäftigen?!

Just my 2 cents... :-)

Viele Grüße

Markus

Autor: Joe G. ([feinmechaniker](#))

Datum: 08.03.2012 16:07

---

Ich finde die Variante von Leo C. ganz gut. An Speicherplatz sollte es nicht mangeln. Der ATmega168 kann ja leicht durch den ATmega328 ersetzt werden.

Autor: Leo C. ([rapid](#))

Datum: 09.03.2012 18:02

Angehängte Dateien:

- [8080int-jmp.asm](#) (112,6 KB, 60 Downloads)

---

[Bernd B. schrieb:](#)

> Brauchen wir wirklich für die DD/FD-Behandlung eigene, vollständige  
> Sprungtabellen?

Dazu habe meine Meinung schon eins höher in der Antwort an Konrad geschrieben.

Markus G. schrieb:

> Wie wäre es, wenn wir, um keine Geschwindigkeitsverluste zu haben und  
> nicht von vorne herein zu viel Speicher zu verbrauchen, erst einmal nur  
> alle "offiziellen" Befehlskombinationen implementieren würden?

Natürlich sollte man (solltest Du :)) die dokumentierten Befehle zuerst implementieren. Ich glaube aber, daß die undokumentierten wenig zusätzlichen Platz belegen. Der Platz für die Sprungtabelle wird sowiso benötigt, und der Rest wird meistens aus bereits vorhanden Teilen zusammen gebaut.

Allgemein:

Ich habe einen (rudimentär getesteten) Vorschlag geliefert, wie man die bestehende Tabelle für die Z80-Mehrbyteopcodes erweitern kann, weil es dazu eine Anfrage gab

([Beitrag "Re: CP/M auf ATmega88"](#)).

Im Anhang ist noch mal Update.

Die ED-Tabelle ist jetzt fast vollständig. Die ED-Befehle sind aber noch nicht alle voll funktionsfähig (Bitte die "TODO:"-Stellen suchen).

Die anderen Tabellen (DDFD, CB und DDFDCB) sind bisher nur Platzhalter. Ich glaube, daß der Code dazu in 16K unterzubringen ist. Im Moment sind noch über 3,5 KByte frei.

**Autor: Horst.S (Gast)**

**Datum: 09.03.2012 21:17**

---

moin moin,  
nach langer zeit mal wieder zeit.....

- 1.) alle HL befehle auf eine 256byte page bringen.
- 2.) auf je eine page die IX und IY befehle. stratadresse im lowbyte identisch zu den HL befehlen.
- 3.) DD/FD schaltet nur das adress highbyte um. nach abarbeitung wieder auf HL high adresse.
- 4.) danach normale befehlsabarbeitung.

macht 512byte für die befehle bei gutem tempo.

**Autor: Horst.S (Gast)**

**Datum: 10.03.2012 16:06**

**Angehängte Dateien:**

- [Z80 Map Neu.xls](#) (324 KB, 54 Downloads)

---

Hier noch meine Tabelle dazu.....

**Autor: Leo C. ([rapid](#))**

**Datum: 10.03.2012 20:50**

**Angehängte Dateien:**

- [Z80 Map Neu-2.xls](#) (329 KB, 61 Downloads)
-

Hallo Horst,

ich habe mir erlaubt, Deine Tabelle etwas umzuformatieren. Kannst Du sie in dieser Form noch gebrauchen?

Wenn ja, die ED-Tabelle habe ich ergänzt. Es fehlen noch die Block-Befehle.

Autor: Leo C. ([rapid](#))

Datum: 10.03.2012 21:31

Angehängte Dateien:

- [init.asm](#) (4,5 KB, 64 Downloads) | [Codeansicht](#)

---

RAM-Test in init.asm

Markus G. hat Recht. Es macht keinen Sinn, das System bei einem RAM-Fehler weiter laufen zu lassen.

Aber

[Markus G. schrieb:](#)

> Durch den Eintrag:

> .equ MEMTEST = 3

>

> (oder höher) in der Datei: config.inc

>

> ist es nun möglich im Falle eines Speicherfehlers auf einem [AVR](#)

> CP/M-Stick V3.0 den/die defekten DRAM-Chips (IC5 / IC6) anzeigen zu

> lassen!

Dieses halte ich aber doch für etwas zu spezialisiert. Nicht jeder hat einen CP/M-Stick V3.0.

Was haltet Ihr denn von folgender Lösung (Anhang)?

CPM on an AVR, v2.1

Testing RAM: fill...wait...reread...

04<00@0000

05<01@0001

06<02@0002

07<03@0003

0C<08@0008

0D<09@0009

0E<0A@000A

0F<0B@000B

14<10@0010

15<11@0011

16<12@0012

17<13@0013

1C<18@0018

1D<19@0019

1E<1A@001A

1F<1B@001B System halted!

Die ersten 16 Fehler (knapper Bildschirm voll) werden ausgegeben und dann gestoppt. An den Mustern kann man u.U. erkennen, welche(s) Bit(s) defekt ist/sind.

Das Ausgabeformat könnte man noch verbessern. Man könnte auch zusätzliche Spalten ausgeben, mit 1-Bits, die 0 sein sollen und 0-Bits, die 1 sein sollen. Aber das halte ich auch wieder für übertrieben.

Autor: Peter Sieg (Gast)  
Datum: 11.03.2012 11:48

---

Wichtig wäre eine einfache Meldung Ram1 Und/Oder Ram2 defekt und gut ist und dann ggf. noch die Bits (für Lötfehler bei LR Aufbau).

Peter

Autor: Markus G. ([thechief](#))  
Datum: 11.03.2012 12:39

---

Markus G. schrieb:

```
> Hallo zusammen,  
>  
> ich habe meine, im Beitrag #2541628 veröffentlichte,  
> Speichertest-Routine nun noch einmal verfeinert:  
>  
> Durch den Eintrag:  
>  
>  
> .equ MEMTEST = 3  
>  
>  
> (oder höher) in der Datei: config.inc  
>  
> ist es nun möglich im Falle eines Speicherfehlers auf einem AVR  
> CP/M-Stick V3.0 den/die defekten DRAM-Chips (IC5 / IC6) anzeigen zu  
> lassen!  
>  
> Die geänderte Assembler Quellcode-Datei habe ich diesem Beitrag  
> angehängen.  
>  
> Um die geänderte Speichertest-Routine zu verwenden sind also nur zwei  
> kleine Schritte nötig:  
>  
> 1. Austausch der vorhandenen Datei init.asm im Projektverzeichnis durch  
> die im Anhang befindliche Datei init.asm.  
>  
> 2. Aktivieren der neuen Routine durch die Anweisung:  
>  
> .equ MEMTEST = 3  
>  
> (oder höher) in der Datei: config.inc  
>  
> Auch die alten Speichertest-Routinen können weiterhin verwendet werden,  
> hierzu ist wie bisher die Anweisung:  
>  
> .equ MEMTEST = 1 oder 2  
>  
> in der Datei: config.inc zu setzen!  
>  
> Siehe auch folgenden Auszug aus der Datei init.asm:  
>  
>  
> ;-----  
> ; Info about MEMTEST:  
> ; 0 = no memtest at all  
> ; 1 = memtest shows defects but carries on  
> ; 2 = memtest shows defect and stops on first failure
```

```
> ; 3 = memtest shows defect, stops on first failure and
> ;      shows which RAM chip has to be replaced
> ;      on an AVR CP/M-Stick V3.0
> ;-----
>
>
> Viele Grüße
>
> Markus
>
>
> PS: @Leo: Du darfst auch die neue Version gerne (wieder) ins SVN
> einchecken!
```

Hallo zusammen,

ja, ich verstehe Leo's Eiwand, dass meine Ramtest Variante 3 nur für den CPM-Stick V3.0 zu gebrauchen ist, das ist absolut richtig! Genau aus diesem Grund habe ich auch die von mir zuvor implementierte Variante 2 (MEMTEST = 2) im Code belassen. Diese ist allgemein gültig, da sie den / die defekten Ram-Chips nicht durch eine IC-Nummer / Position bestimmt und somit für jede Aufbauvariante (Lochraster, ältere Platinenversion, etc.) verwendet werden kann. Die Ergänzung der Ausgabe um die Bitmuster haltet ich auch für sinnvoll - Lötspitzer lassen grüßen... ;-)

Mein Vorschlag wäre also die Variante 2 um die Ausgabe der Bitmuster zu ergänzen.

Viele Grüße

Markus

Autor: Leo C. ([rapid](#))

Datum: 12.03.2012 12:51

Angehängte Dateien:

- [init.asm](#) (5,1 KB, 56 Downloads) | [Codeansicht](#)

---

[Peter Sieg schrieb:](#)

```
> Wichtig wäre eine einfache Meldung Ram1 Und/Oder Ram2 defekt und gut ist
```

```
Ist Ram1 jetzt der obere oder der untere Chip?
```

[Markus G. schrieb:](#)

```
>> ; Info about MEMTEST:
>> ; 0 = no memtest at all
>> ; 1 = memtest shows defects but carries on
>> ; 2 = memtest shows defect and stops on first failure
>> ; 3 = memtest shows defect, stops on first failure and
>> ;      shows which RAM chip has to be replaced
>> ;      on an AVR CP/M-Stick V3.0
```

Variante 1 hat sich offensichtlich nicht bewährt, da RAM-Fehler zu leicht übersehen werden können.

Wenn wir Variante 3 in den allgemeinen Code aufnehmen, werden wir über

kurz oder lang auch Schalterstellungen für die anderen Hardwarevarianten aufnehmen müssen.

Und mein Hauptpunkt: Ich befürchte ganz einfach, daß diejenigen, die nicht in Lage sind, die Bitmuster zu interpretieren, auch die größten Schwierigkeiten haben werden, das Programm selbst mit der richtigen Schalterstellung zu übersetzen.

Im Anhang ist mein aktuelles Verhandlungsangebot :-)  
Die Ausgabe könnte dann ungefähr so aussehen:

```
CPM on an AVR, v2.1
Testing RAM: fill...wait...reread...
Addr xx yy
0000 00 04 -----X--
0001 01 05 -----X--
0002 02 06 -----X--
0003 03 07 -----X--
0008 08 04 ----X----X--
0009 09 05 ----X----X--
000A 0A 06 ----X----X--
000B 0B 07 ----X----X--
000C 0C 04 ----X-----
000D 0D 05 ----X-----
000E 0E 06 ----X-----
000F 0F 07 ----X-----
0010 10 14 -----X--
0011 11 15 -----X--
0012 12 16 ----X----
0013 13 17 -----X-- System halted!
```

Verbesserungen am Format, eine ordentliche Überschriftenzeile, und die Erklärung, was das ganze überhaupt soll, überlasse ich gerne dem Leser. :-)

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)  
**[Re: CP/M auf ATmega88](#)**

Autor: Leo C. ([rapid](#))

Datum: 12.03.2012 13:27

Angehängte Dateien:

- [avrcpm-avr\\_breaktest.zip](#) (115,7 KB, 11 Downloads)

---

Im Anhang ist ein Vorschlag für BREAK-Erkennung und Verarbeitung.

Z. Zt. wird BREAK nur bei einem Character timeout erkannt. Ein BREAK, daß nur wenig länger als ein normales Zeichen dauert. funktioniert noch nicht.

BREAK setzt ein Flag in einem Interpreter-Register, in das auch die Flags für Interrupts (und vielleicht Trace, Einzelschritt...) kommen sollen.

Z. Zt. setzt BREAK den PC auf null. --> Warmstart.  
Denkbar wären auch Kaltstart, Interrupt/NMI, Sprung in Debugger...  
Vielleicht brauchen wir für so was noch ein schönes API.

Ich würde mich freuen, wenn viele Leute das mal mit vielen verschiedenen Terminals/-Emulatoren testen würden. Bei mir funktioniert mit minicom und Windows Hyperterminal, wobei ich für letzteres Google bemühen mußte, um die Break-Taste zu finden (ctrl-break, bzw. Strg-Untbr).  
Möglicherweise gibt es Terminals, die nur ein kurzes BREAK-Signal senden. Mit denen funktioniert dann (noch) nicht, und das wären dann

gute Testkandidaten für die Vervollständigung des Codes.

[Nils Eilers schrieb:](#)

> Was anderes: ich habe gesehen, wird ein BREAK auf der seriellen  
> Schnittstelle derzeit noch nicht behandelt. Das ADM-3A hat eine extra  
> Break-Taste und Terminalprogramme bieten üblicherweise die Funktion, ein  
> Break zu senden. Das würde die Möglichkeit eröffnen, bei Empfang eines  
> Breaks einen Kalt- oder Warmstart auszuführen, oder einen Debugger  
> aufzurufen.

Autor: Leo C. ([rapid](#))

Datum: 16.03.2012 19:19

Angehängte Dateien:

- [avrcpm-avr-178.zip](#) (118,8 KB, 13 Downloads)



- [Bildschirmfoto-ddtz.png](#)  
26,7 KB, 111 Downloads

---

Update

ddtz macht intensiv Gebrauch von Z80-Befehlen und kommt jetzt schon ziemlich weit. Er läuft aber noch nicht wirklich, weil noch eine Menge Befehle fehlen.

Z. Bsp. CB, und DD/FDCB komplett und in der ED-Tabelle fehlen bis auf LDDR noch alle Blocktransfer- und Suchbefehle. Ausserdem sind die implementierten Befehle kaum getestet.

Trace kann jetzt zur Laufzeit durch Schreiben auf einen Port ein- und ausgeschaltet werden. Praktisch zum Testen der neuen Befehle:

-l100,11a

```
0100 MVI A,01
0102 OUT 4F
0104 LXI B,77FF
0107 PUSH B
0108 POP PSW
0109 LXI B,1234
010C LXI D,5678
010F LXI H,9ABC
0112 ??= 08
0113 ??= D9
0114 ??= 08
0115 ??= D9
0116 MVI A,00
0118 OUT 4F
011A NOP
011B
```

-g100,11a

```
N A =01 BC =0000 DE =0000 HL =0000 SP=0100 PC=0104 01 FF 77
A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
N A =01 BC =77FF DE =0000 HL =0000 SP=0100 PC=0107 C5 F1 01
A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
```

```

      N   A =01 BC =77FF DE =0000 HL =0000 SP=00FE PC=0108           F1 01 34
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =77FF DE =0000 HL =0000 SP=0100 PC=0109           01 34 12
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =1234 DE =0000 HL =0000 SP=0100 PC=010C           11 78 56
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =1234 DE =5678 HL =0000 SP=0100 PC=010F           21 BC 9A
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =1234 DE =5678 HL =9ABC SP=0100 PC=0112           08 D9 08
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
          A =00 BC =1234 DE =5678 HL =9ABC SP=0100 PC=0113           D9 08 D9
SZHVNC  A'=77 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
          A =00 BC =0000 DE =0000 HL =0000 SP=0100 PC=0114           08 D9 3E
SZHVNC  A'=77 BC'=1234 DE'=5678 HL'=9ABC IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =0000 DE =0000 HL =0000 SP=0100 PC=0115           D9 3E 00
          A'=00 BC'=1234 DE'=5678 HL'=9ABC IX=0000 IY=0000 I=00
SZHVNC  A =77 BC =1234 DE =5678 HL =9ABC SP=0100 PC=0116           3E 00 D3
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00
SZHVNC  A =00 BC =1234 DE =5678 HL =9ABC SP=0100 PC=0118           D3 4F F7
          A'=00 BC'=0000 DE'=0000 HL'=0000 IX=0000 IY=0000 I=00           *011A
-

```

Da z. Zt. einiges in Bewegung ist, habe ich den alten Stand (8080 only, korrigierter DAA, alter RAM-Test, kein BREAK) als Version 2.2 eingefroren:

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

Die aktuelle Version ist wie üblich im trunk-Zweig.

**Autor: Peter Sieg (Gast)**

**Datum: 19.03.2012 15:20**

---

Super! Habe es gestern mal für 328P übersetzt und läuft ersteinmal soweit..

Gibts für ddtz eigentlich ein Manual?

Peter

**Autor: Leo C. ([rapid](#))**

**Datum: 19.03.2012 15:29**

**Angehängte Dateien:**

- [ddtz.zip](#) (9,9 KB, 26 Downloads)

---

[Peter Sieg schrieb:](#)

> Gibts für ddtz eigentlich ein Manual?

Im angehängten Zip ist sogar eins auf Deutsch drin.

(Ich weiß grad nicht mehr wo ich das her hab, deshalb kann ich keinen Link posten.)

**Autor: Leo C. ([rapid](#))**

**Datum: 19.03.2012 15:46**

**Angehängte Dateien:**

- [DDTZ.DOC.gz](#) (16,5 KB, 24 Downloads)

---

Sorry, das war kein Manual, sondern "nur" eine Art Referenz. Im Netz findet man aber viele Versionen von ddtz mit englischen und deutschen Beschreibungen.

## Turbo Pascal 3 Image – und läuft!

Autor: Leo C. ([rapid](#))

Datum: 20.03.2012 12:10

Angehängte Dateien:

- [avrcpm-avr-183.zip](#) (123,9 KB, 27 Downloads)
- [tp3-hd.dsk.zip](#) (76,6 KB, 33 Downloads)



- [Bildschirmfoto-turbo3.png](#)  
33,4 KB, 141 Downloads

---

'tp3-hd.dsk' in 'CPMDSK\_x.IMG' umbenennen.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 20.03.2012 12:21

---

Glückwunsch!  
Ich bin sprachlos.

Autor: Peter Sieg (Gast)  
Datum: 21.03.2012 08:33

---

@Leo C. Super! Herzlichen Glückwunsch!!

Muss ich die Tage gleich mal ausprobieren..

Vielen Dank!

Gruß Peter

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)  
**[Re: CP/M auf ATmega88](#)**

Autor: Peter Sieg (Gast)  
Datum: 21.03.2012 20:43

---

Hab ich ausprobiert.. Compile von hello.pas geht.. aber sollte da nicht ein hello.com gespeichert sein.. (man ist das lange her mit TP ;-)

Peter

Autor: Leo C. ([rapid](#))

Datum: 21.03.2012 20:52

---

Man müßte doch auch ein Handbuch im Netz finden.  
(Ich habe hier noch eine Kopie der gedruckten Fassung)

Aber Du wirst es sicher auch so noch finden. Sooo viele Punkte hat das Hauptmenü ja auch nicht.

Autor: Peter Sieg (Gast)

Datum: 21.03.2012 20:59

---

Arrg.. compiler (O)ptions - (C)om file ;-)

Ja, ja lang ist es her..

Super!

Peter

Autor: Leo C. ([rapid](#))

Datum: 21.03.2012 21:06

---

Im Z80-Interpreter dürften aber noch ein paar Fehler versteckt sein.

Der Turbo-Editor spinnt, und MC.PAS kann zwar als COM übersetzt werden, aber mit dem Demosheet verrechnet es sich, wenn man andere/größere Zahlen eingibt.

Letzte Nacht habe ich mal den Exerciser laufen lassen.  
Heute Morgen sah das dann so aus:

```
C>exz80doc
Z80 instruction exerciser
Undefined status bits NOT taken into account
<adc, sbc> hl, <bc, de, hl, sp>.... ( 71,680) cycles OK
add hl, <bc, de, hl, sp>..... ( 35,840) cycles OK
add ix, <bc, de, ix, sp>..... ( 35,840) cycles OK
add iy, <bc, de, iy, sp>..... ( 35,840) cycles OK
aluop a, nn..... ( 45,056) cycles OK
aluop a, <b, c, d, e, h, l, (hl), a>.. (753,664) cycles OK
aluop a, <ixh, ixl, iyh, iyl>..... (376,832) cycles ERROR **** crc
expected:a4026d5a found:e7318427
aluop a, (<ix, iy>+1)..... (229,376) cycles ERROR **** crc
expected:e849676e found:74a1545f
aluop a, (<ix, iy>-1)..... (229,376) cycles ERROR **** crc
expected:edd8136d found:7130205c
bit n, (<ix, iy>+1)..... ( 2,048) cycles OK
bit n, (<ix, iy>-1)..... ( 2,048) cycles OK
bit n, <b, c, d, e, h, l, (hl), a>.... ( 49,152) cycles OK
cpd<r>..... ( 6,144) cycles ERROR **** crc
expected:c8439345 found:a8554cee
cpi<r>..... ( 6,144) cycles ERROR **** crc
expected:e66fa7d2 found:478b76a1
<daa, cpl, scf, ccf>..... ( 65,536) cycles
VN A =8E BC =9079 DE =8D5B HL =A559 SP=299D PC=253B 27 00 00
A'=00 BC'=0000 DE'=0000 HL'=0000 IX=1D60 IY=09FA I=00 Invalid
opcode!
```

Diese Fehler sind inzwischen korrigiert. Der Abbruch mit "Invalid opcode!" kam beim DAA-Befehl, weil dort immer noch der Teil mit gesetztem N-Flag fehlt. Den Abbruch habe ich jetzt aber raus gemacht, damit ich sehen kann, wie weit das Testprogramm die nächste Nacht kommt.

Außerdem habe ich heute die noch fehlenden Block-I/O Befehle eingebaut. Update im [SVN](#).

Autor: Joe G. ([feinmechaniker](#))

Datum: 22.03.2012 17:27

Angehängte Dateien:



- [TP.jpg](#)  
71,6 KB, 105 Downloads

---

Klasse Leo!

TP 3.0 läuft bei mir nun auch. Der Editor bockt zwar noch etwas doch mit dem Exerciser sollten sich alle Fehler ausmerzen lassen. Es dauert....

Meiner rechnet immer noch hier:

```
<adc,sbc> hl,<bc,de,hl,sp>....( 71,680) cycles
```

Autor: Joe G. ([feinmechaniker](#))

Datum: 22.03.2012 17:47

---

[Leo C. schrieb:](#)

> Man müßte doch auch ein Handbuch im Netz finden.

<http://electrickery.xs4all.nl/comp/tp30/>

Autor: Leo C. ([rapid](#))

Datum: 22.03.2012 17:58

Angehängte Dateien:

- [avrcpm-avr-185b.zip](#) (124,6 KB, 27 Downloads)

---

Hallo Joe,

danke fürs testen.

Versuch's mal mit der Version im Anhang. Damit scheint jetzt alles zu laufen.

Der Turbopascal-Editor läuft damit, und das Microcalc Demo auch.

Der Exerciser sah bei mir zuletzt so aus:

```
C>ex1
Z80 instruction exerciser
Undefined status bits NOT taken into account
<inc,dec> a.....( 3,072) cycles   OK
<inc,dec> b.....( 3,072) cycles   OK
```

<inc,dec> bc.....	( 1,536)	cycles	OK
<inc,dec> c.....	( 3,072)	cycles	OK
<inc,dec> d.....	( 3,072)	cycles	OK
<inc,dec> de.....	( 1,536)	cycles	OK
<inc,dec> e.....	( 3,072)	cycles	OK
<inc,dec> h.....	( 3,072)	cycles	OK
<inc,dec> hl.....	( 1,536)	cycles	OK
<inc,dec> ix.....	( 1,536)	cycles	OK
<inc,dec> iy.....	( 1,536)	cycles	OK
<inc,dec> l.....	( 3,072)	cycles	OK
<inc,dec> (hl).....	( 3,072)	cycles	OK
<inc,dec> sp.....	( 1,536)	cycles	OK
<inc,dec> (<ix,iy>+1).....	( 6,144)	cycles	OK
<inc,dec> (<ix,iy>-1).....	( 6,144)	cycles	OK
<inc,dec> ixh.....	( 3,072)	cycles	OK
<inc,dec> ixl.....	( 3,072)	cycles	OK
<inc,dec> iyh.....	( 3,072)	cycles	OK
<inc,dec> iyl.....	( 3,072)	cycles	OK
ld <bc,de>, (nnnn).....	( 32)	cycles	OK
ld hl, (nnnn).....	( 16)	cycles	OK
ld sp, (nnnn).....	( 16)	cycles	OK
ld <ix,iy>, (nnnn).....	( 32)	cycles	OK
ld (nnnn), <bc,de>.....	( 64)	cycles	OK
ld (nnnn), hl.....	( 16)	cycles	OK
ld (nnnn), sp.....	( 16)	cycles	OK
ld (nnnn), <ix,iy>.....	( 64)	cycles	OK
ld <bc,de,hl,sp>, nnnn.....	( 64)	cycles	OK
ld <ix,iy>, nnnn.....	( 32)	cycles	OK
ld a, <(bc), (de)>.....	( 44)	cycles	OK
ld <b,c,d,e,h,l, (hl), a>, nn.....	( 64)	cycles	OK
ld (<ix,iy>+1), nn.....	( 32)	cycles	OK
ld (<ix,iy>-1), nn.....	( 32)	cycles	OK
ld <b,c,d,e>, (<ix,iy>+1).....	( 512)	cycles	OK
ld <b,c,d,e>, (<ix,iy>-1).....	( 512)	cycles	OK
ld <h,l>, (<ix,iy>+1).....	( 256)	cycles	OK
ld <h,l>, (<ix,iy>-1).....	( 256)	cycles	OK
ld a, (<ix,iy>+1).....	( 128)	cycles	OK
ld a, (<ix,iy>-1).....	( 128)	cycles	OK
ld <ixh,ixl,iyh,iyl>, nn.....	( 32)	cycles	OK
ld <bcdehla>, <bcdehla>.....	( 3,456)	cycles	OK
ld <bcdexya>, <bcdexya>.....	( 6,912)	cycles	OK
ld a, (nnnn) / ld (nnnn), a.....	( 44)	cycles	OK
ldd<r> (1).....	( 44)	cycles	OK
ldd<r> (2).....	( 44)	cycles	OK
ldi<r> (1).....	( 44)	cycles	OK
ldi<r> (2).....	( 44)	cycles	OK
neg.....	( 16,384)	cycles	OK
<rrd,rld>.....	( 7,168)	cycles	OK
<rlca,rrca,rla,rra>.....	( 6,144)	cycles	OK
shf/rot (<ix,iy>+1).....	( 416)	cycles	OK
shf/rot (<ix,iy>-1).....	( 416)	cycles	OK
shf/rot <b,c,d,e,h,l, (hl), a>..	( 6,784)	cycles	OK
<set,res> n, <bcdehl (hl) a>.....	( 6,912)	cycles	OK
<set,res> n, (<ix,iy>+1).....	( 448)	cycles	OK
<set,res> n, (<ix,iy>-1).....	( 448)	cycles	OK
ld (<ix,iy>+1), <b,c,d,e>.....	( 1,024)	cycles	OK
ld (<ix,iy>-1), <b,c,d,e>.....	( 1,024)	cycles	OK
ld (<ix,iy>+1), <h,l>.....	( 256)	cycles	OK
ld (<ix,iy>-1), <h,l>.....	( 256)	cycles	OK
ld (<ix,iy>+1), a.....	( 64)	cycles	OK
ld (<ix,iy>-1), a.....	( 64)	cycles	OK
ld (<bc,de>), a.....	( 96)	cycles	OK

All tests successful.

Ich habe die Source neu übersetzt und dabei alle Tests die schon positiv durchgelaufen sind, sowie DAA (bekannter Fehler) und die Block-I/O Befehle auskommentiert.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 22.03.2012 18:04

---

Prima, der Editor läuft nun auch!

Wenn du Lust hast dieses Stück zu übernehmen...

```
instr do_fetch_DIR8, op_DJNZ, do_store_nop ;10 nn ;DJNZ nn
(Z80 OK)
```

```
-----
;|Mnemonic |SZHPNC|Description |Notes |
;|-----+-----+-----+-----|
;|DJNZ e |-----|Dec., Jump Non-Zero |B=B-1 till B=0 |
;
;The b register is decremented, and if not zero, the signed value e is
added to pc.
;The jump is measured from the start of the instruction opcode.
;e = Relative addressing (PC=PC+2+offset)
do_op_DJNZ: ; decremt B, jump B=0
    lds temp, z_b ; B in temp
    dec temp ; temp decrementieren
    breq do_op_DJNZ_Z ; bei B=0
    sts z_b, temp ; temp in B
    subi opl, 0x80 ; z_pc + e im Zweierkomplement
    subi z_pcl, 0x80
    sbc z_pch, _0
    add z_pcl, opl
    adc z_pch, _0
do_op_DJNZ_Z:
    ret
```

Autor: Leo C. ([rapid](#))

Datum: 22.03.2012 19:38

---

[Joe G. schrieb:](#)

> Wenn du Lust hast dieses Stück zu übernehmen...

Nur wenn Du den Fehler behältst. ;-)

Die trickreiche Displacement-Berechnung braucht einen Takt (und ein Register) weniger. Super!

Die Abkürzung ("Gehe nicht über store\_...") bringt aber mehr.

Autor: Leo C. ([rapid](#))

Datum: 23.03.2012 11:27

Angehängte Dateien:

- [Z80int-jmp.asm](#) (135,9 KB, 64 Downloads)

---

Hallo Joe,

Die angehängte geht noch einen Tick besser. Bis auf DAA werden damit alle Tests von EXZ80DOC bestanden.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 23.03.2012 11:43

---

Hallo Leo,

super läuft. Ich lasse gerade EXZ80DOC laufen und gehe erst mal was essen...

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 23.03.2012 15:16

Angehängte Dateien:



• [Cursor Befehle.pdf](#) | [anzeigen](#)

15,2 KB, 75 Downloads

---

Hier die Befehle für den TP Editor.

Autor: Leo C. ([rapid](#))

Datum: 24.03.2012 09:28

---

Da der Z80-Interpreter nun komplett ist, und gut läuft, habe ich mir erlaubt, die Versionsnummer auf 3.0 zu erhöhen. Diese Version ist auch wieder dauerhaft im tags-Verzeichnis abgelegt:

<http://cloudbase.homelinux.net/viewvc/avr-cpm/avr-cpm/branches/3.0/>

- Z80 ist jetzt der Defaultprozessor. Das Programm belegt damit z.Zt. aber mehr als 16 KB Flash. --> Atmega328 erforderlich.
- Für 8080 kann übersetzt werden, wenn "EM\_Z80" in config.inc auf 0 gesetzt wird.
- Trace kann jetzt (auch) zur Laufzeit ein- und ausgeschaltet werden.  
Einschalten: 1 auf Port 0x4F ausgeben.  
Ausschalten: 0 auf Port 0x4F ausgeben.  
(ddtz: "0 1 4f")
- Um Trace von der CP/M-Befehlszeile aus zu schalten, wurde das Timer-Steuerprogramm (im Verzeichnis cpm/utills) erweitert.  
(Der Name TIMER.COM passt nicht mehr, Vorschläge?)
- BREAK: Ein laufendes Programm kann jederzeit durch senden von BREAK abgebrochen werden. Zur Zeit wird dann ein Warmstart ausgeführt,

was natürlich nicht hilft, wenn BDOS und/oder BIOS durch ein amoklaufendes Programm überschrieben wurden.

Bitte gründlich testen. Kommentare, Änderungs- und Verbesserungsvorschläge wie immer...

Schönes Wochenende

Autor: Joe G. ([feinmechaniker](#))

Datum: 24.03.2012 10:12

---

Perfekt!

Trace läuft...

Break läuft...

TIMER.COM darf auch ACT.COM heißen. (AVR CP/M Tool) ;-)

Autor: Peter Sieg (Gast)

Datum: 24.03.2012 10:38

---

Hi.

Habe gerade Ramblade (ZiCOG - der Welt kleinste CP/M Computer ;-)) vor mir liegen: PCB is 1.9"x1.2" (48x30.5mm) SMT (surface mount)

<http://forums.parallax.com/showthread.php?116893-R...>

Und dachte mir so, das wäre doch mal eine Herausforderung @Joe?

Atmega328p in SMD

MicroSD Card Slot

2x4256 Drams in SMD (sind leider sehr groß - Alternative SRam?)

Bisschen Hühnerfutter

Nur Anschluß eines CP2102 Dongles = +3.3V und TTL ser. IO

Platine von beiden Seiten bestückt (sonst wird das nichts)

Peter

Autor: Joe G. ([feinmechaniker](#))

Datum: 24.03.2012 11:00

---

[Peter Sieg schrieb:](#)

> Und dachte mir so, das wäre doch mal eine Herausforderung @Joe?

Das Projekt liegt schon in der Schublade. Als RAM habe ich 16 Stück HM51W17805J-6 rumliegen (16M EDO DRAM 3.3V). Es würde nur ein IC benötigt werden da er als 2M x 8 organisiert ist. Mit dem Refresh bin ich mir nicht ganz sicher, sollte jedoch gehen (vielleicht schaut Leo auch mal aufs Datenblatt).

Es könnte dann eine CP/M 3.0 Jubiläumsversion geben ;-)

Autor: Joe G. ([feinmechaniker](#))

Datum: 24.03.2012 16:44

---

Eine Frage zur RAMDISK.

Wenn ich #define RAMDISKCNT auf 1 setze wird eine Ramdisk angelegt. I: z.B.

Da ja RAMSIZE nicht ausgewertet wird und in der aktuellen Stickversion nur 2x256Kx4 verbaut sind, wo liegt dann der Ram und wie groß ist er? STAT zeigt mir 1M für I: an, was ja nicht sein kann. In diesem Zustand werden auch alle Programme fehlerhaft beendet. Zum Glück gibt es ja nun Break ;-)

Autor: Leo C. ([rapid](#))

Datum: 24.03.2012 19:31

---

Ohje Ramdisk,

die habe ich außer Betrieb genommen, als ich die Erweiterung für mehrere Diskformate eingebaut hatte. Mit etwas Glück, könnte sie laufen, wenn man die Parameter in config.inc (RAMSIZE) anpasst. Bin aber nicht optimistisch.

Der Hauptgrund, warum ich mich nicht mehr dafür interessiert habe ist, daß die RAM-Zugriffe langsamer sind, als die Zugriffe auf die SD-Card.

Zum EDO-Ram: Refresh ist sicher kein Problem, und auch sonst müßte es genauso funktionieren, wie die normalen D-Rams. EDO ist ja nur eine weitere "Betriebsart".

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 25.03.2012 10:27

---

EDO-Ram: Ich setzte mal einen Testweise auf die Stickversion.

Ramdisk: Ist nur schade wenn man 2MB verschenkt.

Mal eine ganz andere Frage in den Raum geworfen.

Ob man die **Jugend** begeistern kann, wenn man ein "CP/M - Shield" für Arduino ins Leben ruft. Das könnte in beliebigen Ausbaustufen entstehen. Arduino Nano - Shield mit RAM und SD-Card - kleinstes CP/M System der Welt ;-)

Arduino Uno - CP/MLino Shield mit VGA und Tastatur

...

Joe

Autor: Peter Sieg (Gast)

Datum: 25.03.2012 14:07

---

@Joe: Die Idee das als Arduino Nano Addon zu machen gefällt mir! Läuft allerdings mit 5V = Levelshifter zur SD Karte erforderlich. Aber das lässt sich ja mit 3k3 und 1k8 (6 Stück) SMD Widerständen machen.

Leider sind die Arduino nicht wirklich 'billig'.. hatte gerade was um die 30 US\$ gesehen.. aber naja.. auf dem addon wären dann nur Ram und (Micro-) SD Karte inkl. Levelshiftern..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 25.03.2012 16:31

---

Den Mini gibt es als Arduino Pro Mini Enhancement (3.3V/5V adjustable)/16MHz MEGA328P-AU für 7,84€ inc. Versand.

Autor: Peter Sieg (Gast)

Datum: 25.03.2012 19:28

---

Jup:

Ebay-Artikel Nr. [160762063172](#)

Da fehlt dann aber ein USB Dongle..

Peter

Autor: Leo C. ([rapid](#))

Datum: 25.03.2012 23:36

---

[Joe G. schrieb:](#)

> Ob man die **Jugend** begeistern kann, wenn man ein "CP/M - Shield" für

...

> Arduino Uno - CP/MLino Shield mit VGA und Tastatur

Arduino kenne ich ja noch. Aber "Shield" ... mußte ich erst mal nachschlagen.

Ja. Shield muß sein. Und Lino muß unbedingt in den Namen. Aber ob das für CP/M. daß ja zu seinen Lebzeiten schon nicht wirklich cool war, bei der "Jugend" reicht? Also wenn es nicht mindestens Gezwitscher blubbern kann [1], und einen "Gefällt mir" Knopf hat, sicher nicht.

[1] <http://bubblino.com/>

Autor: Peter Sieg (Gast)

Datum: 26.03.2012 16:14

---

@Leo C: Haha.. Wink mit dem Zaunpfahl..

Nein ich sehe es folgendermaßen:

Wenn wir ein 'Shield' für diese Plattform anbieten haben wir:

- autom. eine fertige Platine für [AVR](#), Quarz, Vin, ggf. USB?
- Ein 'Shield' mit Micro-SD, ggf. Levelshiftern und DRam.

Vorteile:

- [AVR](#) Teil kann man fertig kaufen, ist getestet, läßt sich woanders für verwenden.
- Nur Zusatzplatine muss selbst gebaut werden.
- Evtl. läßt sich Zusatzplatine auch für anderes verwenden!

Nachteile:

- evtl. höherer Preis für beides zusammen anstatt auf einer Platine?

Und an China kommt man solange man sehr günstige Preise haben will nicht umher (sorry, aber ist so! alles andere ist leider realitätsfern..).

Just my 2 cents

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.03.2012 17:27

---

Die Bastlergemeinde ist sowieso nicht wirklich zu durchschauen. Der Raspberry-Pi hat einen waren Hype ausgelöst. Mehr als 10.000 verkaufte Boards innerhalb von Stunden. Ja und was kann man damit tun? Auch nicht

mehr als mit dem CP/M Teil. Na gut Doom spielen aber Ladder und Chess sind auch ganz nett. An Programmiersprachen gibt es unter CP/M alles als das Herz begehrt - sogar ohne dot.net Geraffel. Ich überlege mir mal eine Shield Variante und stelle sie dann zur Diskussion. Über I2C darf dann auch bubblino angeschlossen werden ;-)

Autor: Leo C. ([rapid](#))

Datum: 27.03.2012 13:09

Angehängte Dateien:

- [avrcpm-20120327-ramdisk192k.zip](#) (61,9 KB, 31 Downloads)

[Joe G. schrieb:](#)

> Eine Frage zur RAMDISK.

Im aktuellen [SVN](#) war nur Code für 4-bit Hardware.

Im Anhang ist jetzt mal eine 8-Bit 192 KB Ramdisk. Getestet habe ich nur mit 4 MBit Chips, aber es sollte auch mit kleineren funktionieren. Da die Ramdisk-Größe über Tabellen im BIOS statisch festgelegt ist, muß dieses eben auch angepaßt werden. Deshalb ist im Anhang auch gleich ein Diskimage.

Hier mal der Versuch (ausbaufähig), die Speicherorganisation darzustellen:

Linear	21  20  19  18  17  16  15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
DRAM	R C R C R C R C R C  Row   Col
	10   9   8   7   6   5   4   3   2   1   0   7   6   5   4   3   2   1   0
Ramdisk	DSK 0-3  Track 0-255   Sector 0-31   Byte 0-127
Memory	1 1 1 1 1 1   RAM-Adr. 0-65536

Wenn der Z80-Arbeitsspeicher adressiert wird, sind die Adressleitungen A8-A10 immer High, egal ob entsprechend große RAMs vorhanden oder nicht. Der Arbeitsspeicher liegt also immer am oberen Ende des physikalischen RAMs.

Die RAM-Disks füllen den Speicher von unten. Maximal sind 4 Disks a 1 MByte möglich. Zur Zeit wird nicht geprüft, ob eine Ramdisk-Adresse (Disk/Track/Sector) überhaupt in den vorhandenen Speicher hinein paßt. Es ist also möglich, durch Schreiben in die RAM-Disk, den Arbeitsspeicher zu überschreiben.

Eine 192K Disk wird im BIOS (BIOS.MAC) folgendermaßen definiert:

```
; Name spt bls dks dir cks off
dpb rd192, 32, 1024, 192, 32, 0, 0
dpb: Macro, daß aus den folgenden Daten den "disk parameter block"
generiert.
spt: sectors per track. Die Zahl muß zum AVR Ramdisk-Treiber passen.
bls: block size (KByte).
dks: disk size. Anzahl Blöcke der Größe bls.
dir: Anzahl Directory Einträge
cks: check size. Anzahl zu prüfender Directory Einträge.
Bei Wechselmedien = cks, sonst 0.
off: offset. Anzahl reservierter Tracks. Wenn off bei der ersten
Ramdisk > 0 ist, wird beim Kaltstart das System hier her kopiert,
und beim Warmstart von hier, statt von SD-Karte geladen. Bringt im
```

Moment fast nix, da Original BDOS beim Warmstart ohne Disk A:  
hängen bleibt.  
(s.a. [Beitrag "Re: CP/M auf ATmega88"](#))

Die Ramdisk wird beim Starten nicht gelöscht (formatiert), so daß am  
Anfang meistens Müll drin steht, den man auch durch "ERA \*.\*" in allen  
Userbereichen nicht weg bekommt. Deshalb gibt es das Programm WIPE.COM  
auf dem angehängten Diskimage.

TODO:

- RAMDISK-Zugriff auf für Ramdisk tatsächlich vorhandenes RAM  
einschränken. ("#define RAMSIZE" in config.inc)
- Ramdisk(s) abhängig von RAMSIZE dynamisch beim Start einrichten.  
Analog zu SD-Karten-Images. BIOS-Anpassung dann nicht mehr notwendig.
- Bei Power-On-Reset die Ramdisk initialisieren.  
Sonstige Resets übersteht der Ramdiskinhalt in der Regel, auch wenn  
dabei der Refresh für längere Zeit ausfällt.
- Ramdisk-Treiber für 4-bit-RAM aktualisieren.
- MMU: "Ge-paged-tes" RAM für CP/M3

Freiwillige vor.

**Autor: Holger (Gast)**

**Datum: 27.03.2012 15:36**

**Angehängte Dateien:**

- [RAMFLOP4.MAC](#) (3,5 KB, 41 Downloads)

---

Die Ramdisk haben wir damals initialisiert, indem wir den ersten  
versteckten DIR-Eintrag getestet haben. War es unser Eintrag, war die  
Initialisierung fertig. Waren fremde Bytes drin, so wurde unser Eintrag  
geschrieben, der Rest vom DIR gelöscht und in den Speicherbereich, auf  
den der DIR-Eintrag zeigt, das CCP kopiert.

**Autor: Peter Sieg (Gast)**

**Datum: 30.03.2012 18:39**

---

Noch mal was abgefahrenes:

<http://www.dmitry.co/index.php?p=../04.Thoughts/07....>

; -)

Peter

[Beitrag melden](#) [Bearbeiten](#) [Löschen](#) [Markierten Text zitieren](#) [Antwort](#) [Antwort mit Zitat](#)  
**[Re: CP/M auf ATmega88](#)**

Autor: Konrad S. ([maybe](#))

Datum: 30.03.2012 22:40

---

Peter Sieg schrieb:

> Noch mal was abgefahrenes:

>

> <http://www.dmitry.co/index.php?p=../04.Thoughts/07....>

Abgefahren? Ja, aber auf eine Art, vor der ich den Hut ziehe!

Jetzt fehlt nur noch, dass jemand mehrere ATmegas nimmt und damit ein Multi-Core-System aufbaut.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 09.04.2012 13:26

---

[Leo C. schrieb:](#)

> Im Anhang ist jetzt mal eine 8-Bit 192 KB Ramdisk.

Danke Leo! Deine Version geht nun auch mit 256 x 8. Das gibt genau eine Ramdisk von 192 KB. Unter Verwendung von Wipe läßt sie sich auch sauber löschen und dann beschreiben. Die Variante zunächst das BIOS zu ändern und dan ein neues CP/M zu bauen ist akzeptabel. Dabei viel mir auf, das in unsrer Version SUBMIT nicht richtig ausgeführt wird. M80 und L80 werden einfach nicht gestartet. Auf AltairSIMH läuft es jedoch problemlos

Joe

Autor: Leo C. ([rapid](#))

Datum: 14.04.2012 14:54

---

[Joe G. schrieb:](#)

> Dabei viel mir auf, das  
> in unsrer Version SUBMIT nicht richtig ausgeführt wird. M80 und L80  
> werden einfach nicht gestartet. Auf AltairSIMH läuft es jedoch  
> problemlos

Das habe jetzt endlich mal getestet und ist hier leider genauso. Aber früher ging das mal...

Vorläufig kann man DO.COM (bzw. SuperSUB), z.B. aus obigem Emulator-Paket nehmen, das sowieso besser ist.

Noch was anderes.

Da wir ja inzwischen auch Z80 können, wäre es auch interessant, einen der verbesserten BDOS-Clones zu installieren. Diese wiederum würden von einem Uhrenbaustein profitieren.

Da ich demnächst sowieso einige Bauteile bestellen will, möchte ich einen RTC mitbestellen. Welcher Chip ist denn hier so gängig/beliebt? Beim [Reichelt](#) käme wohl nur der [DS1307](#) in Frage.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 14.04.2012 15:08

---

Der [DS1307](#) ist sehr weit verbreitet und sicherlich nicht die schlechteste Wahl.

Ich hatte mir die Tage auch schon mal Gedanken darüber gemacht wie I2C zu benutzen ist. Es könnten ja dann dort ein 8 Bit I/O Port oder A/D oder oder angeschlossen werden. Die BIOS Funktionen 05 - Stanzer und 06 - Lesereingabe werden doch eigentlich nie mehr benötigt. Diese Funktionen könnten doch eigentlich genutzt werden um auf den I2C zu schreiben bzw. vom I2C zu lesen. Wir müßten uns nur noch ein ordentliches Protokoll einfallen lassen.

An andere BDOS-Clones habe ich auch schon gedacht CP/M 3.0 ...

Autor: Leo C. ([rapid](#))

Datum: 14.04.2012 16:29

---

[Joe G. schrieb:](#)

> Der [DS1307](#) ist sehr weit verbreitet und sicherlich nicht die  
> schlechteste Wahl.

Na, dann kommen mal  $\geq 2$  [1] auf die Liste.

> oder oder angeschlossen werden. Die BIOS Funktionen 05 - Stanzer und 06  
> - Lesereingabe werden doch eigentlich nie mehr benötigt. Diese Funktionen  
> könnten doch eigentlich genutzt werden um auf den I2C zu schreiben bzw.

Ich habe noch einen Stapel Lochkarten. Aber leider keine Stanze. ;-)

Imho greifen alle gängigen CP/M Kommunikationsprogramme direkt auf die  
Ports der seriellen Schnittstelle zu, weil die CP/M-Kanäle kaum zu  
gebrauchen sind.

Man könnte z.B. im [AVR](#)-Teil (damals) gängige UARTs emulieren.

> An andere BDOS-Clones habe ich auch schon gedacht CP/M 3.0 ...

CP/M 3 ging vorher auch schon, da es mit 8080 auskommt. Macht meiner  
Meinung nach ohne Banking wenig Sinn. Die Alternativen leisten dann  
mehr, bei weniger RAM-Verbrauch.

[1] Da ich z.Zt. dabei bin, meine gute alte HD64180 Karte [2]  
wiederzubeleben, soll die natürlich nicht zurück stehen müssen.  
Allerdings wäre dafür ein RTC mit SPI besser.

[2] [Beitrag "Re: z80 system"](#)

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 19.04.2012 17:44

Angehängte Dateien:



• [CPM Lino Brd.jpg](#)

60,4 KB, 162 Downloads

---

CP/M Lino

CP/M Shield für Arduino Pro Mini

- Größe 20 mm x 50 mm
- Mini USB B
- Micro SD
- 2 MB RAM

Wollen wir?

Autor: Peter Sieg (Gast)

Datum: 20.04.2012 10:07

---

Sieht schick aus!

USB Anbindung wie? [FT232RL](#)?

Welcher Ramchip (1 oder 2)? Bezugsquelle?

Gibts passende Arduino Pro Mini über [ebay](#) oder wo?

Ansonsten wäre ich mit 2 Stück dabei (je nach Preis auch 3-5).

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 20.04.2012 10:20

---

USB über [FT232RL](#), die Variante über den [MCP2200](#) würde noch einen externen Quarz benötigen.

DRAM ist ein HM51W17805 (16M EDO RAM 2-Mword x 8 Bit)

Als Arduino Pro Mini würde auch ein DFRobot Pro Mini V1.2 5V 16MHz

Arduino Compatible gehen (9,9 €) allerdings müsste bei allen Varianten (auch China) ein 3.3V Regler eingesetzt werden.

Platine derzeit 4 Lagen, es geht sehr eng zu! bei einer Abnahmemenge von 10 liegt der Preis ca. um 11€.

Joe

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 20.04.2012 10:23

---

Nachtrag:

16 RAM's habe ich derzeit. Eine Quelle ist auch hier:

<http://www.demotronic.net/>

Stückpreis 1,95€

Autor: Peter Sieg (Gast)

Datum: 23.04.2012 18:59

---

[Joe G. schrieb:](#)

> USB über [FT232RL](#), die Variante über den [MCP2200](#) würde noch einen externen Quarz benötigen.

> DRAM ist ein HM51W17805 (16M EDO RAM 2-Mword x 8 Bit)

> Als Arduino Pro Mini würde auch ein DFRobot Pro Mini V1.2 5V 16MHz

> Arduino Compatible gehen (9,9 €) allerdings müsste bei allen Varianten (auch China) ein 3.3V Regler eingesetzt werden.

> Platine derzeit 4 Lagen, es geht sehr eng zu! bei einer Abnahmemenge von 10 liegt der Preis ca. um 11€.

> Joe

Hmm.. ich werde (für mich) das Gefühl nicht los, das wir dann mit dem 'USB Stick' Konzept besser dran sind.. was meinen die anderen..

Peter

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 24.04.2012 07:16

---

War nur ein Vorschlag Peter...

ZSDOS Portieren ist ja auch eine nette Aufgabe.

Joe

Autor: Peter Sieg (Gast)  
Datum: 24.04.2012 18:58

---

@Joe: Die/Deine Arbeit an sich ist klasse!!  
Nur wenn halt alles so mal zusammen rechne.. entschwinden z.Z alle  
Vorteile..

Eine separate Platine - evtl. direkt für CP2102 Dongle Anschluß.. geht  
evtl. etwas größer mir Bestückung von beiden Seiten.. und viel  
günstiger..  
Atmega328P+DRAM+Micro-SD+Hühnerfutter

Peter

Autor: Leo C. ([rapid](#))  
Datum: 24.04.2012 19:36

---

[Peter Sieg schrieb:](#)

> Hmm.. ich werde (für mich) das Gefühl nicht los, das wir dann mit dem  
> 'USB Stick' Konzept besser dran sind.. was meinen die anderen..

Definiere "wir".

[Peter Sieg schrieb:](#)

> Nur wenn halt alles so mal zusammen rechne.. entschwinden z.Z alle  
> Vorteile..

Also weiter oben hast Du noch ganz anders argumentiert.

Könnte ja sein, daß das Ding in der "Arduino-Szene" richtig einschlägt.  
(Zumindest hier merkt man aber leider bisher nichts davon.)

[Joe G. schrieb:](#)

> ZSDOS Portieren ist ja auch eine nette Aufgabe.

Genau. Super.

Autor: Leo C. ([rapid](#))  
Datum: 25.04.2012 11:36

---

SUBMIT

hatte ich mir gestern Abend nochmal angeschaut.

Joe, falls der Submit-Job "nur" dann nicht ausgeführt wird, wenn er  
**nicht** auf Laufwerk A gestartet wird, handelt es sich (leider) um das  
dokumentierte Verhalten von CP/M 2.x. [1]:

"If the SUBMIT function is performed on any disk other than  
drive A, the commands are not processed until the disk is inserted  
into drive A and the system reboots."

It's not a bug, ...

Beim simh Altair 8800 Emulator funktioniert Submit auch auf anderen  
Laufwerken, weil dort dafür (und einige andere Kleinigkeiten) im BIOS  
Patches vorhanden sind, die bei jedem Warmstart an CCP und BDOS  
appliziert werden.

Man kann sich das Patchen aber sparen, wenn man Submit durch eine der  
zahlreich vorhandenen Alternativen (DO, SUPERSUB, JOB, ...) ersetzt.

[1] <http://www.cpm.z80.de/manuals/archive/cpm22htm/ch1...>

# Wir haben ZSDOS

Autor: Joe G. ([feinmechaniker](#))

Datum: 25.04.2012 12:15

---

[Leo C. schrieb:](#)

> Um das zu ändern braucht man eine möglichst vollständige  
> CPM-Distribution.

Ich habe mal das CCP und das BDOS von ZSDOS übersetzt und statt des CPM.SYS von sprite\_tm eingesetzt. ZSDOS läuft nun auf unserem System. Jetzt müßte natürlich noch das BIOS angepaßt werden. Für alle die selbst mal damit experimentieren wollen:

cpm.bin - beinhaltet das neue ZSDOS  
CCP.BIN - CCP Binärcode für das 62K Modell  
ZSDOS.BIN - BDOS Binärcode für das 62K Modell  
CCP.MAC - CCP Quelle  
ZSDOS.MAC - BDOS Quelle  
CPMDSK\_A.IMG - Bootfile

Um cpm.bin zu bilden müssen bei dd die Speichermodelladressen berücksichtigt werden.

```
dd conv=sync bs=128 count=1 if=ipl.bin > cpm.bin
dd conv=sync bs=128 count=16 if=CCP.BIN >> cpm.bin
dd conv=sync bs=128 count=28 if=ZSDOS.BIN >> cpm.bin
dd conv=sync bs=128 count=6 if=bios.bin >> cpm.bin
```

Viel Spaß beim ausprobieren!

Joe

A>ddt

DDT VERS 2.2

-de400

```
E400 00 00 00 00 00 00 C3 9B E4 4C E7 4C E7 4C E7 4C .....L.L.L.L
E410 E7 F1 F1 00 00 6D DD F1 DD F1 DD F1 DD F1 DD F1 .....m.....
E420 DD F1 DD F1 00 00 17 01 00 00 00 00 00 80 00 .....;.....
E430 00 00 15 F5 7B F4 23 F5 32 F5 42 F5 1A 00 03 07 ....{.#.2.B....
E440 00 F2 00 3F 00 C0 00 10 00 02 00 02 20 00 00 00 ...?.....
E450 00 00 00 00 02 00 40 00 CD E3 00 0F 00 20 E4 20 .....@.....
E460 E4 A5 E3 5A 53 44 4F 53 20 31 2E 31 20 43 6F 70 ...ZSDOS 1.1 Cop
E470 79 72 69 67 68 74 20 28 63 29 20 31 39 38 37 2C yright (c) 1987,
E480 38 38 20 6F E5 BA E6 83 E6 6F E5 BA E6 83 E6 20 88 o.....o.....
E490 00 20 E4 20 00 BB E8 E6 E7 00 00 AF 47 6F 67 22 .....Gog"
E4A0 4C E4 22 4E E4 ED 73 61 E4 31 9B E4 DD E5 D5 DD L."N..sa.1.....
E4B0 E1 DD 22 5D E4 DD 22 5F E4 21 E6 E7 E5 79 32 4B .."].."_.!...y2K
```

Autor: Joe G. ([feinmechaniker](#))

Datum: 25.04.2012 12:20

Angehängte Dateien:

- [ZSDOS.zip](#) (47,1 KB, 19 Downloads)

---

Und hier die Files

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 25.04.2012 15:52

Angehängte Dateien:

- [CPMDSK\\_D.zip](#) (243,4 KB, 23 Downloads)

---

[Leo C. schrieb:](#)

> Man kann sich das Patchen aber sparen, wenn man Submit durch eine der  
> zahlreich vorhandenen Alternativen (DO, SUPERSUB, JOB, ...) ersetzt.

Leider bricht DO oder SUPERSUB auch mit einem Fehler ab, M80 und L80  
einzeln benutzt geht jedoch (Die Dateien sind auch auf der HD).

```
D>do makecpm
SuperSUB V1.1
Memory full error on line number: 2501
```

Anbei noch alle Dateien um sich ein komplettes ZSDOS zu basteln. Das  
Diskimage ist wie immer im Altairformat.  
Joe

Autor: Leo C. ([rapid](#))

Datum: 25.04.2012 17:58

Angehängte Dateien:

- [makecpm.sub](#) (256 Bytes, 25 Downloads)

---

Joe, versuch's mal hiermit.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.04.2012 10:00

---

[Leo C. schrieb:](#)

> Joe, versuch's mal hiermit.

Danke! Es geht damit. Doch warum? Liegt der Unterschied wirklich nur  
darin, dass dein File genau 256 Byte lang ist?

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.04.2012 10:31

---

@ Leo C.  
Ok, es war ^Z

Autor: Leo C. ([rapid](#))

Datum: 26.04.2012 10:41

---

> Ok, es war ^Z

Es hätte mich auch gewundert, wenn Du nicht drauf gekommen wärst. Ich war aber gerade dabei, eine längere Antwort zu tippen. Vielleicht interessiert's ja noch jemanden.

Die 256 Byte sind eher Zufall. Ich habe mit ddt den Sektor nach dem Text mit 01AH (== ^Z == CP/M End Of File) aufgefüllt. Die Datei habe ich danach mit SAVE 1 <dateiname> gespeichert.

SuperSub scheint beim Einlesen das physikalische Dateiende nicht zu beachten, und versucht dann den gesamten Arbeitsspeicher als Text zu interpretieren, bzw. bis ein EOF-Zeichen erkannt wird.

CP/M 3 (und ZSDOS glaube ich auch) können den Füllgrad des letzten Sektors speichern (und damit die exakte Dateilänge), aber wahrscheinlich macht kein CP/M-Programm davon Gebrauch. CP/M Texteditoren speichern natürlich immer (mindestens) ein 01Ah am Textende.

Vielleicht hat Dein PC-Texteditor ja eine Möglichkeit, Steuerzeichen direkt einzugeben. Ansonsten: echo, dd, copy, cat, ...

Autor: Joe G. ([feinmechaniker](#))  
Datum: 26.04.2012 10:57

---

[Leo C. schrieb:](#)

> Vielleicht hat Dein PC-Texteditor ja eine Möglichkeit

Ich nehme nun WinVi...

@alle

Um die Vorteile des ZSDOS nun auch wirklich nutzen zu können, braucht es nun der Uhr. Vielleicht können wir uns auf eine gemeinsame Hardwarevariante verständigen. Anbei mein Vorschlag. Die Uhr ([DS1307](#)) ist über einen Pegelwandler am I2C Bus angeschlossen. Weiterhin ist am Bus ein [PCF8574](#) (8 Bit I/O). Somit hätte man neben der Uhr noch die Möglichkeit einen vollständigen 8 Bit Port zu benutzen (LCD, Tastatur, Led, wie auch immer). Außerdem können ja weitere I2C Geräte angeschlossen werden.

Kommentare, Vorschläge?

Joe

Autor: Leo C. ([rapid](#))  
Datum: 26.04.2012 11:58

---

[Joe G. schrieb:](#)

> Um die Vorteile des ZSDOS nun auch wirklich nutzen zu können, braucht es  
> nun der Uhr.

Um die Zeit bis zur funktionierenden RTC zu überbrücken, bastel ich gerade an der Interrupt-Uhr.

> Die Uhr ([DS1307](#))  
> ist über einen Pegelwandler am I2C Bus angeschlossen.

2. Stromversorgung und Pegelwandler wegen eines unpassenden Uhrenbausteins?

Wie wärs mit [PCF8583](#) oder [DS1337](#) + 2 Dioden?

> Weiterhin ist am

> Bus ein [PCF8574](#) (8 Bit I/O). Somit hätte man neben der Uhr noch die  
> Möglichkeit einen vollständigen 8 Bit Port zu benutzen (LCD, Tastatur,  
> Led, wie auch immer). Außerdem können ja weitere I2C Geräte  
> angeschlossen werden.

Was könnte man denn noch anschließen wollen?

Drucker vielleicht? Dann bräuchte man 2. [PCF8574](#) oder zB. [PCA9555](#).

2. Serielle? Gibs I2C-UARTS? 2. Software-UART stelle ich mir schwierig vor.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 26.04.2012 12:48

Angehängte Dateien:



- [avrcp\\_m.pdf | anzeigen](#)  
117,5 KB, 68 Downloads

---

Ok, der [PCF8574](#) geht auch bei 3.3V und ist bei [Reichelt](#) als DIL und Smd verfügbar. [PCF8574](#) und [PCA9555](#) gehen auch bei 3.3V und haben 5V tolerante Eingänge. Der Pegelwandler kann also entfallen.

Bei der Erweiterung dachte ich in Richtung Anwendung Arduino. Ein embedded System (ich hab das CP/M gerade geadelt) lebt ja von externer Hardware. Mit 8 oder 16 Bit I/O könnte der potentielle Nutzer schon eine Menge anschließen. Ideal wäre die Ansteuerung wie ich schon mal vorgeschlagen habe über die ungenutzten Biosfunktionen Reader.

Ersetzt man den mega328p durch die smd Version, bekommt man noch zwei zusätzliche Pins in Form von 2 AD-Wandlern spendiert (ADC6 und ADC7). Mein derzeitiger Entwurf sieht gerade so aus (pdf). Er hat die Maße 100mm x 70mm und passt damit genau in ein Fischer Alu-Gehäuse (AKG 7124 ME). Das System beinhaltet neben der USB ein VT100 Terminal mit VGA und Tastaturanschluss den zwei ADC + I2C, sowie 8 Bit I/O.

Eine ähnliche Variante hatte ich schon mal vorgestellt. Sie ist hier nicht sosehr auf Interesse gestoßen. Macht jedoch nichts, die neue baue ich mir selber auf. Ich glaube im Gegensatz zum embedded Linux (raspberrypi u.ä.) kann ein potentieller Interessent (Schüler, Studenten, wie auch immer) damit ein komplettes System komplett selber aufbauen, sein CP/M (Kernel) selbst konfigurieren und zusammenstellen. Das System ist wirklich einfach und überschaubar.  
Joe

Autor: Markus G. ([thechief](#))

Datum: 26.04.2012 13:57

---

[Peter Sieg schrieb:](#)

> [Joe G. schrieb:](#)

>> USB über [FT232RL](#), die Variante über den [MCP2200](#) würde noch einen  
>> externen Quarz benötigen.  
>> DRAM ist ein HM51W17805 (16M EDO RAM 2-Mword x 8 Bit)  
>> Als Arduino Pro Mini würde auch ein DFRobot Pro Mini V1.2 5V 16MHz  
>> Arduino Compatible gehen (9,9 €) allerdings müsste bei allen Varianten  
>> (auch China) ein 3.3V Regler eingesetzt werden.

>> Platine derzeit 4 Lagen, es geht sehr eng zu! bei einer Abnahmemenge von  
>> 10 liegt der Preis ca. um 11€.  
>> Joe  
>  
> Hmm.. ich werde (für mich) das Gefühl nicht los, das wir dann mit dem  
> 'USB Stick' Konzept besser dran sind.. was meinen die anderen..  
>  
> Peter

Hallo zusammen,

ich werde, für mich persönlich, mit meinem "USB Stick" weiter arbeiten.

@Joe: Aber meinen ausdrücklichen Respekt, wie Du immer die Layouts hier zauberst... :-)

Und vielen Dank für den Anstoss zur ZSDOS Portierung! Klasse!

Viele Grüße

Markus

Autor: Markus G. ([thechief](#))

Datum: 26.04.2012 14:10

---

Leo C. schrieb:

> Joe G. schrieb:

>

>> Um die Vorteile des ZSDOS nun auch wirklich nutzen zu können, braucht es  
>> nun der Uhr.

>

> Um die Zeit bis zur funktionierenden RTC zu überbrücken, bastel ich  
> gerade an der Interrupt-Uhr.

>

>> Die Uhr ([DS1307](#))

>> ist über einen Pegelwandler am I2C Bus angeschlossen.

>

> 2. Stromversorgung und Pegelwandler wegen eines unpassenden  
> Uhrenbausteins?

> Wie wärs mit [PCF8583](#) oder [DS1337](#) + 2 Dioden?

>

>> Weiterhin ist am

>> Bus ein [PCF8574](#) (8 Bit I/O). Somit hätte man neben der Uhr noch die  
>> Möglichkeit einen vollständigen 8 Bit Port zu benutzen (LCD, Tastatur,  
>> Led, wie auch immer). Außerdem können ja weitere I2C Geräte  
>> angeschlossen werden.

>

> Was könnte man denn noch anschließen wollen?

> Drucker vielleicht? Dann bräuchte man 2. [PCF8574](#) oder zB. [PCA9555](#).

> 2. Serielle? Gibs I2C-UARTS? 2. Software-UART stelle ich mir schwierig  
> vor.

Hallo zusammen,

bezüglich des Uhrenbausteins tendiere ich zum [PCF8583](#) wie von Leo vorgeschlagen, hat auch noch 240 Byte RAM als "Dreingabe" kann man bestimmt einmal gebrauchen... ;-)

Oder gibt es Etwas das uns veranlassen sollte explizit einen [DS1337](#) zu nutzen?

Die I2C Bus-Erweiterung per [PCF8574](#) o.ä. finde ich auch super!

In Zusammenhang mit meinem "AVR/CPM-Stick" denke ich gerade an eine "Sandwich"-Platine als Aufsatz auf den [AVR](#)-Sockel des [AVR/CPM](#)-Sticks.

Somit hätte ich dann beliebig viel Platz für Erweiterungen...

Just my 2 cents

Viele Grüße

Markus

Autor: Leo C. ([rapid](#))

Datum: 26.04.2012 14:44

Angehängte Dateien:



• [PCF8563jingdu\\_en.pdf](#) | [anzeigen](#)

142,5 KB, 31 Downloads

---

> bezüglich des Uhrenbausteins tendiere ich zum [PCF8583](#) wie von Leo  
> vorgeschlagen, hat auch noch 240 Byte RAM als "Dreingabe" kann man  
> bestimmt einmal gebrauchen... ;-)

Das RAM könnte man zum bisher ungenutzten EEPROM legen. ;-)

> Oder gibt es Etwas das uns veranlassen sollte explizit einen [DS1337](#) zu  
> nutzen?

Der [PCF8583](#) ist schon etwas älter und wohl auch einige Designschwächen.  
Ein besserer Nachfolger ist der [PCF8563](#), der aber schwieriger zu  
bekommen ist. Ob der [DS1337](#) irgend einen Vorteil hat, kann ich jetzt  
auch nicht sagen.

Autor: Leo C. ([rapid](#))

Datum: 26.04.2012 15:44

---

[Joe G. schrieb:](#)

> Mir ist es egal, der PCF8563 ist u.a. bei Farnell zu haben, also kein  
> Problem.

Mir ist's auch egal. Die Softwareunterschiede sind gering. Man kann  
sicher mit einem Treiber beide ansprechen. Leider sind sie nicht ganz  
pinkompatibel. Aber wenn man an Pin 3 (A0/INT) und 7 (INT/CLKOUT) je  
einen Pullup vorsieht, müßten beide Chips damit klarkommen. Dann kann  
jeder seinen Wunschchip einstecken.

Ich habe vor, bei [Reichelt](#) einen [PCF8583](#) zu bestellen. Für mein anderes  
Projekt möchte ich einen [DS2417](#) (1-wire) nehmen. Ginge hier auch. ;-)

Autor: Leo C. ([rapid](#))

Datum: 27.04.2012 21:50

## Angehängte Dateien:

- [AVRCLOCK-01.zip](#) (5,7 KB, 12 Downloads)



- [Bildschirmfoto-avrclock.png](#)  
45,3 KB, 99 Downloads

---

Im trunk gibts jetzt eine Uhr.  
Im Anhang ist der passende ZSDOS-Treiber, den ich noch nicht eingecheckt habe.

Nach einem Reset wird die Uhr z.Zt. mit Tag und Monat 0 initialisiert.  
Dies dürfte zu verschmerzen sein, da sie ja ohnehin gestellt werden muß.

Das ganze ist vorläufig. U.a. deshalb, weil ich die BCD/binär- und zurück-Wandlung noch vom Z80- in den [AVR](#)-Teil verlegen will. Außerdem behandelt sie 2100 noch irtümlicherweise als Schaltjahr.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.04.2012 10:18

---

@ Leo C.  
SUPER Leo!

D>ldtim

ZSDOS Time Stamp Loader, Ver 1.1  
Copyright (C) 1988 by H.F.Bower / C.W.Cotrill

hallo welt ...loaded at D8D8H  
Clock is : AVRCPM Clock 0.1

D>td  
- 00, 2000 00:02:45  
D>td 4/28/12 10:12:28  
Press any key to set time  
Apr 28, 2012 10:12:28  
D>td  
Apr 28, 2012 10:12:31  
D>

Die Datei svnrev.inc fehlt noch im trunc.  
bios.asm und ipl.asm würde ich langsam entsorgen, führt nur zu Verwirrungen.  
Wollen wir wirklich cpm6X.sys einführen? Ich fand die Trennung von CCP und BDOS recht transparent.

Autor: Leo C. ([rapid](#))

Datum: 28.04.2012 11:05

---

[Joe G. schrieb:](#)

> Die Datei svnrev.inc fehlt noch im trunc.

Hier gibts das Tool, das die Datei erzeugt.

<http://www.compuphase.com/svnrev.htm>

Ich weiß noch nicht, ob ich das Ganze so lassen soll.

> bios.asm und ipl.asm würde ich langsam entsorgen, führt nur zu  
> Verwirrungen.

Ja.

> Wollen wir wirklich cpm6X.sys einführen?

Nein. Das war/ist nur ein Ergebnis meiner Experimente mit movcpm. Da steckt auch noch viel Arbeit drin, weil die Distri, die Sprite\_tm als Grundlage genommen hat, kein movcpm enthält, und die Seriennummern angeglichen werden mußten.

> Ich fand die Trennung von CCP und BDOS recht transparent.

Ja, in diese Richtung sollte es gehen. Ich würde "unser" CP/M gerne durch ein wirklich vollständiges ersetzen. ZSDOS dann als Alternative in einem weiteren Zweig.

Außerdem sollte irgendwann noch ein Buildscript kommen, das eine Release als Zip-Paket für Endanwender erzeugt. Das Paket sollte dann ein fertiges CP/M-Image enthalten, und keine Abhängigkeiten von [SVN](#).

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 29.04.2012 17:44

Angehängte Dateien:

- [AVRCPM.SUB](#) (1,6 KB, 18 Downloads)

---

[Leo C. schrieb:](#)

> Außerdem sollte irgendwann noch ein Buildscript kommen, das eine Release  
> als Zip-Paket für Endanwender erzeugt.

Hier ein kleines Script welches eine bootfähige cpm.bin Datei aus BIOS, CCP, BDOS und IPL erzeugt. Es kann vollständig in AltairSIMH gestartet werden und schreibt das File auch gleich wieder zurück. dd entfällt damit. Die Version ist derzeit nur für ZSDOS könnte aber sicher leicht für die Altair cp/m Version angepaßt werden.

Joe

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.04.2012 13:57

Angehängte Dateien:

- [System.zip](#) (44 KB, 19 Downloads)

---

Ich hab mal das BIOS etwas nach der obigen Variante geändert. Die Speichergröße wird in dieser Variante nur noch in der MEMCFG.LIB angegeben. Damit kann nun eine beliebige (in den Grenzen des CP/M) Speicherkonfiguration übersetzt werden. Anbei ein Bsp. mit 53K Speicher.

CPM on an [AVR](#), v3.1 r180  
Testing RAM: fill...wait...reread...  
Initing mmc...

A:FAT16 File-Image at: 8450, size: 018KB.  
B:FAT16 File-Image at: 8959, size: 256KB.  
C:FAT16 File-Image at: 8703, size: 256KB.  
Partinit done.  
Ok, Z80-CPU is live!

ipl  
53k cp/m vers 2.2

A>dir  
A: CPMSTAT COM  
A>cpmstat  
- Logged - ---- Records ---- - Tracks - --- Capacity --- --- TPA  
---  
Drive User Block Track Drive Sys. Drive Directory Drive Bytes  
K  
A: 0 8 26 1944 2 77 64/ 64 243K 46854  
45.76  
  
- Operating System -  
Version BDOS BIOS  
CP/M 2.2 B806 C600

A>

Autor: Leo C. ([rapid](#))  
Datum: 30.04.2012 14:06

---

[Joe G. schrieb:](#)

```
> Ich glaube in unsrer Berechnung der Speichergröße ist noch eine Altlast  
> sprite_tm eingebaut. Derzeit wird der CCP Start im BIOS über  
>  
> msize equ 62 ;size of available RAM in k  
> bias equ (msize-20) * 1024  
> ccp equ 3400h+bias ;base of cpm ccp  
> bdos equ ccp+806h ;base of bdos  
> bios equ ccp+1600h ;base of bios
```

Sprite\_tm ist daran eher unschuldig, glaube ich. Das war ein Mißverständnis meinerseits. Ich hatte msize vom altairz80 Emulator übernommen. Dort steht msize aber für die tatsächliche RAM-Größe und nicht für die "CP/M-Größe", für die ich es dann genommen habe.

```
> berechnet. Der Summand -20 bei der Biasberechnung steht auch so im CP/M  
> Operating System Manual worauf sich sprite_tm ja bezieht. Doch  
> tatsächlich ist unser System nicht 20K sondern 22K groß. Das kann  
> schnell nachgerechnet werden.
```

Das CP/M Handbuch definiert ein 20K System, bei dem der CCP auf 3400H liegt. Unser System ist demgegenüber um b=A800H verschoben, und damit per Definition ein 62K System.

<http://www.cpm.z80.de/manuals/archive/cpm22htm/ch6...>

```
> 3400h Sysgen + 800h CCP + E00h BDOS + D00h Bios = 22k
```

```
> User CCP fängt auch bei DC00h an. Hätten wir ein 62k System, so würde es
```

> bei D400h (62K - 9K) beginnen müssen. Im Bios müsste also msize auf 64  
> und im bias (msize-22) \* 1024 geändert werden. Mit dieser Korrektur  
> könnte dann auch eine gemeinsame Speicherconfig für BIOS, CCP, und BDOS  
> verwendet werden.

Die 22 ist im CP/M-Handbuch nicht definiert. Hätten wir ein 64K System,  
würde unser BIOS auf F400H beginnen (CCP bei E400H).

> Lag ich völlig daneben oder ist es nur noch keinem aufgefallen?

Mir ists am letzten Freitag aufgefallen. Da hatte ich in den von Dir  
verwendeten ZRDOS-Sourcecode geschaut und msize (wieder-) entdeckt.

Autor: Leo C. ([rapid](#))

Datum: 30.04.2012 14:12

---

[Joe G. schrieb:](#)

> Ich hab mal das BIOS etwas nach der obigen Variante geändert. Die  
> Speichergröße wird in dieser Variante nur noch in der MEMCFG.LIB  
> angegeben. Damit kann nun eine beliebige (in den Grenzen des CP/M)  
> Speicherkonfiguration übersetzt werden. Anbei ein Bsp. mit 53K Speicher.

```
bias equ (msize-22) * 1024
ccp equ 3400h+bias ;base of cpm ccp
bdos equ ccp+806h ;base of bdos
bios equ ccp+1600h ;base of bios
```

Nix dagegen. Da wir von allen Systemteilen den Sourcecode haben, und  
damit auch alles auf beliebige Adressen linken können, brauchen wir  
diese Offset-Berechnung sowiso nicht mehr.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.04.2012 14:13

---

[Leo C. schrieb:](#)

> Da hatte ich in den von Dir  
> verwendeten ZRDOS-Sourcecode geschaut und msize (wieder-) entdeckt.

Dann mache ich wieder eine -20 und damit ein 62k System daraus und  
ändere es im ZRDOS-Code.

Oh sehe gerade Doppelpost - welche Variante nehmen wir?

Autor: Leo C. ([rapid](#))

Datum: 30.04.2012 16:52

---

[Joe G. schrieb:](#)

> Dann mache ich wieder eine -20 und damit ein 62k System daraus und  
> ändere es im ZRDOS-Code.

Der Original-ZRDOS-Code braucht das eigentlich garnicht. (REL-File,  
Adresse wird erst beim Linken festgelegt.) Quelle hier:

<http://www.gaby.de/ftp/pub/cpm/znode51/specials/zs...>

> Oh sehe gerade Doppelpost - welche Variante nehmen wir?

Im letzten Posting habe ich mir wohl mehrfach selber widersprochen.

Was wir nicht (mehr) brauchen, ist das "20K CP/M System" als Referenz.  
Die Länge von CCP und BDOS könnte man auch anders bestimmen (Linker).

Das ist aber unnötig aufwendig, weil die seit CP/M 2.0 immer gleich geblieben sind. Aus historischen Gründen bin ich dafür, ein CP/M-System, dessen BIOS bei F200H beginnt, weiterhin als 62K-System zu bezeichnen. Eine Definition für die RAM-Größe brauchen wir eigentlich nicht, da sich an unseren 64K voraussichtlich nichts ändern wird. Deine Frage habe ich jetzt immer noch nicht beantwortet. ;)

Vielleicht sollten wir die 3 Dateien MEMCFG.LIB, CFGACPM.LIB, AVRCPM.LIB zu einer einzigen zusammen fassen? Oder AVRCPM.LIB doch getrennt lassen, da sie keine hardwareabhängigen Teile hat, und wirklich nur fürs BIOS gebraucht wird?

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.04.2012 17:14

---

Klingt sinnvoll.

Ein System mit BIOS ab F200H bezeichnen wir mit 62K und MEMCFG.LIB CFGACPM.LIB fassen wir zu einer Datei zusammen. AVRCPM.LIB würde ich auch extra stehen lassen, da sie ja nur fürs BIOS zuständig ist.

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 23.08.2012 11:40

Angehängte Dateien:

- [CPMDSK\\_A.IMG](#) (256 KB, 26 Downloads)
- [CPMDSK\\_D.zip](#) (315,2 KB, 19 Downloads)
- [Source.zip](#) (49 KB, 26 Downloads)

---

@Franz

Noch ein Aufbau, sehr schön! Welchen DRAM verwendest Du?

Anbei zwei ZSDOS Images.

CPMDSK\_A.IMG ist ein bootfähiges Image mit einem Uhrentreiber von Leo C. CPMDSK\_D.IMG beinhaltet das komplette System im Altair HD Format. Hier sind auch die Quellen für das System enthalten. Du kannst das System selbst mit AVRCPM.SUP erzeugen (braucht bei 20 MHz ca. 30 Sekunden). Source.zip enthält die Quellen für das BIOS, CCP und IPL

Gruß Joe

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 23.08.2012 11:59

---

Nachtrag:

Ich hatte mich gewaltig mit der Zeit verschätzt. Um das System vollständig auf unserem System zu übersetzen, d.h. Assembler, Linker, DDT usw. benötigt man ca. 5 Min und 20 Sekunden. Unter Altair natürlich nur wenige Sekunden.

Gruß Joe

# Blaue Platinen

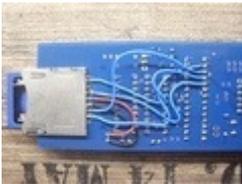
Autor: Peter Sieg ([petersieg](#))

Datum: 15.10.2012 19:50

Angehängte Dateien:



- [avrcpm\\_blaue.jpg](#)  
218,2 KB, 117 Downloads



- [avrcpm\\_attend\\_bot.jpg](#)  
228,7 KB, 107 Downloads

---

Gerade nochmals SCDA6A0101 bei CSD bestellt.. Nun habe ich wieder genug Slots für alle Platine.. habe trotzdem auch mal einen Ersatz-Slot 'drangefriemelt'.. geht auch..

Peter

# Fortran und PL/I

Autor: Peter Sieg ([petersieg](#))

Datum: 28.10.2012 12:54

Angehängte Dateien:

- [disk\\_gh.zip](#) (198,9 KB, 7 Downloads)

---

Hier mal ein paar neue Diskimages und viele Fragen ;-)  
CPMDSK\_G.IMG = Fortran Compiler  
CPMDSK\_H.IMG = PL/I Compiler

Pl/I läuft komplett mit z.B FIB.PLI:

PLI FIB  
LINK FIB  
FIB

Fortran ist ein hello.for dabei, aber der Compiler beschwert sich..?  
(Leider ist das editieren von Files unter CP/M auch nicht so einfach..  
hat da  
jemand ein gut funktionierendes VDE.COM / ZDE.COM?)

Ein paar Fortran Beispielprogramme, die sich hiermit compilieren lassen  
wäre schön..

## AltairSimH zum übersetzen des BIOS

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 28.10.2012 19:42

Angehängte Dateien:

- [Altair.zip](#) (1,1 MB, 17 Downloads)

---

Kurze Anleitung dazu.

1. Altair.zip entpacken
2. AltairAll.bat starten
3. auf Laufwerk I: wechseln
4. mit "do avrcpm.sub" das Übersetzen und Linken starten
5. fettich Meister, im Verzeichnis von Altair steht die fertige CPM.BIN

schönen Abend  
Joe

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 29.10.2012 15:27

Angehängte Dateien:



- [avrcp\\_m.pdf](#) | [anzeigen](#)  
44,6 KB, 10 Downloads

---

@Leo C.

Ja, der hat RV2123-C2 ist auch nicht schlecht. Er hat schon den Quarz integriert.

@Peter

ZSDOS hat bei voller Kompatibilität zu CP/M 2.2 einige wesentliche Verbesserungen. Näheres hier:

<http://susowa.homeftp.net/index.php/software-mainm...>

@alle

Es gibt einen Entwurf mit RTC ([PCF8583T](#)) und einem 8-Bit Port (PCF 8574). Außerdem gibt es einen Erweiterungsport mit I2C und 2 x ADC. Die Stromversorgung ist über den USB-Port realisiert, über den wahlweise das VT100 Terminal angeschlossen werden kann oder der Propeller (VGA mit VT100 Terminal) programmiert werden kann. Weiterhin gibt es einen PS2 Anschluss für die Tastatur. Die Platine passt in ein Fischer Alu-Gehäuse (AKG 7124

ME)beziehen u.a. bei [Reichelt](#).

Da bisher kein großes Interesse vorhanden war, habe ich den Aufbau noch nicht in Angriff genommen. Doch jetzt scheint wieder Leben in das Projekt zu kommen. Welche Wünsche gibt es denn noch?

\* **Anzeige-Link**

Autor: Joe G. ([feinmechaniker](#)) 

Datum: 30.10.2012 16:38

Angehängte Dateien:

- [neu.zip](#) (9,8 KB, 8 Downloads)

---

Ein guter Editor unter Windows ist WinVi  
<http://www.winvi.de/de/>

1. neu.dsk im altairSIMH als HD anlegen

```
d tracks[0-7] 254
attach dsk zsdos.dsk
;attach hdsk i.dsk
attach hdsk neu.dsk
set cpu 64k
set cpu itrap
set cpu z80
set cpu altairrom
set cpu nonbanked
reset cpu
boot dsk
```

2. HD mit Programmen füllen

3. SIMH beenden

4. neu.dsk in CPMSK\_X.IMG umbenennen (X) ist der Laufwerksbuchstabe deiner HD

5. auf SD-Card kopieren

6. freu

Joe

Autor: Peter Sieg ([petersieg](#))

Datum: 01.11.2012 20:14

---

Tja.. es sind immer die dummen Fehler..

So rum gehts:

```
mkfs.cpm.exe -f simhd -b cpm.bin -L test simhd.img
makeimage simhd.img simhd.img 8388608
```

Peter

Autor: Peter Sieg ([petersieg](#))

Datum: 01.11.2012 20:22

---

Auch Interessant!

Auf dem Altair Emulator wird man auf A: wie obern geschildert zurück  
gebeamt.. auf dem avrcpm NICHT:

A>e:

E>user 6

E>dir

E: CREF80 COM : F80 COM : HELLO FOR : L80 COM

E: LIB80 COM : M80 COM : OBSLIB REL : FORLIB REL

E: README TXT : HELLO REL : HELLO COM

E>hello

HELLO, WORLD

STOP \*\* at address 0131 \*\*

E>dir

E: CREF80 COM : F80 COM : HELLO FOR : L80 COM

E: LIB80 COM : M80 COM : OBSLIB REL : FORLIB REL

E: README TXT : HELLO REL : HELLO COM

E>

???

Peter

## MuMath

Autor: Peter Sieg ([petersieg](#))

Datum: 04.11.2012 18:12

Angehängte Dateien:

- [mumath.zip](#) (944,9 KB, 4 Downloads)

---

Hier noch ein ZIP mit MuMATH Package (bisher nur im Altair Emu getestet)  
und eine deutsche PDF Anleitung dazu.

From: <http://www.retroarchive.org/cpm/misc/misc.htm>

This package was originally issued for the OSBORNE 1.  
The MUSIMP.COM file contains my patch to make it useable  
under generic cp/m. The original osborne version of this  
executable is enclosed as MUSIMP.OBJ

The CLES?.\* files are a tutorial in using the calculation  
parts of MUMATH. The PLES?.\* comprise a programming tutorial  
for MUSIMP.

rwd

06/09/98

---

Schnellstart

Zum Kennenlernen von Musimp:

Starte Musimp am CP/M prompt: MUSIMP ALL  
Setze „PAUSE“ für die nachfolgende Demo durch folgende Eingabe am “?”  
Prompt: PAUSE:100;  
(Altair Emulator: PAUSE:1000;)  
Starte die Demo durch folgende Eingabe am “?” Prompt: RDS(DEMO,ALL);  
Beende Musimp

Peter