

改良版も公開しています。

[2012.11.11] AVRとZ80でCP/Mの改良(高速化対応)

[2014.05.31] 部品を減らす工夫(よりシンプルにする方式)

あわせてご覧ください。

## Z80の時代

Z80が登場した頃までは、マイコンといえば自作することも珍しくはありませんでした。最終目標はCP/Mを動かすことでした。当時多くのマイコン自作本が出版されていました。私は6800で自作したことはあるのですがCP/Mマシンは本を買うまでで止まっていた。

[http://star.gmob.jp/koji/data/Z80\\_book.jpg](http://star.gmob.jp/koji/data/Z80_book.jpg) [http://star.gmob.jp/koji/data/CPM\\_book.jpg](http://star.gmob.jp/koji/data/CPM_book.jpg)

[拡大1](#) [拡大2](#)

### CP/M on an AVR(avrcpm)

AVRでZ80をシミュレートしてCP/Mを動かす「CP/M on an AVR」というのがある。シンプルな手書きの回路図。オーバースペックなハード設計。ソフトはアセンブラ1本のみでDRAMのリフレッシュまでもやっている、というとっても怪しげなもの。

だめもとで作ってみました。DRAMが手に入らないのでジャンク箱から探し出したold-MacのSIMMメモリ(秋葉原にAppleのショップがたくさんあったころ、無料で配布されていたやつ)を使っています。Mega88は手持ちに無かったのでMega168Pを使っています。ファームの修正は割り込みベクトルを2バイトに変更するだけでOKでした。

<http://star.gmob.jp/koji/data/avrcpm.jpg> [拡大](#)

結果はとりあえず動いたんですが以下の問題がありました。

1. ウォームブート時のシステムのリロード処理が実装されていない。
2. エミュレータに実装されていないコードがある。
3. エミュレーションにバグがある。

1.についてはBIOSを自分で修正して解決したのですが、別のサイトがやっていました。このサイトに置かれているbios.asmは実装されています。

問題は2.と3.で、エミュレートしているのはZ80ではなく8080で、しかもDAA命令が実装されていません。ASMやDDTの一部のコマンドが動きません。自分でDAA命令を実装しASMとDDTは動くようになったのですがMBASICが動かない。どうやらエミュレーションにバグがあるようです。さすがにこちらを修正するのは無理。作者はゲームさえ動けばよいと考えていた模様。

【追記】このサイトやこのサイトに改善版があるようです。試してませんけど。 [検証例](#)

結局お手上げとなりました。日本でも同じようなことをやっている人がいて、こちらはz80packというエミュレータパッケージを使っているようですが今は入手できないようです。入手できるそうです(コメント欄参照)

結論として、ちゃんとした CP/M を動かしたいならエミュレータは危険じゃなからうかということとです。配線は楽ですが検証には相当手間が掛かりそうです。

### 本物の Z80 を使う

そんなわけで Z80 は実物を使い AVR でなるべくシンプルにまとめることにしました。Z80 は「鈴商」や「丹青通商(ヤフオク)」で入手可能です。

<http://star.gmob.jp/koji/data/CPMdiag.PNG>

I/O アドレスを「CP/M on an AVR」に合わせてあるのでそこで公開されているディスクイメージがそのまま使えます。入出力インターフェースは UART(38400,8N1) なのでシリアル端末か、USB シリアル変換器が必要です。私は、自作のシリアル端末か Recursion さんの「CDC-232 for ATtiny2313」を使っています。

<http://star.gmob.jp/koji/data/avrz80cpm.jpg>

### 拡大

[http://star.gmob.jp/koji/data/CPM\\_pc.jpg](http://star.gmob.jp/koji/data/CPM_pc.jpg) <http://star.gmob.jp/koji/data/cdc232.jpg>

[LCD 画面](#) [CDC-232 拡大](#) [TeraTerm 画面](#)

## 自作の LCD シリアル端末と USB シリアル変換器

### 設計概要

- ・ブートローダー機能を実現するため、Z80 にリセットをかけているときにコントロールバスも開放するように工夫しています。
- ・I/O リクエストがあったときのウェイトをかけるタイミングのみソフトでは難しかったのでハードでやってます。
- ・AVR は擬似 I/O、DMA コントローラのような動作をし、シリアル I/F、ストレージデバイス I/F として機能します。
- ・起動時は SD カードの先頭 128 バイトをメモリにロードし、バスを返却して Z80 をリセットします。カードにアクセスできないときは AVR の EEPROM の内容をロードするようにしてあります。
- ・SD カードのアクセス処理は [elm-ChaN さん](#) の FatFs サンプルに付属の mmc 用下位レイヤインターフェースモジュールを使用させていただきました。(FatFs 自体は導入していません)

### [クロック回路図](#) [CPU 回路図](#)

製作は自己責任でお願いします。

### 関連ページ

[CP / M コーナー](#)

[AVR と Z80 で CP / M の改良](#)

[mbed\(ST Nucleo F401RE\) 版 CP/M エミュレータ](#)

[部品を減らす工夫](#)

[メモ集](#)

[CP / M ディスクイメージ作成手順メモ](#)

[CP / M コマンド概要メモ](#)

[CP / M 資料メモ](#)

DSK 形式のディスクイメージからファイルを取り出す方法メモ  
DISKDEF マクロの使い方メモ  
CP / M メモリサイズ変更手順メモ  
CP / M でソフト開発メモ

ソフトウェア

CpmtoolsGUI

関連サイト

CP/M とは

<http://ja.wikipedia.org/wiki/CP/M>

マニュアル

<http://www.d1.dion.ne.jp/~ytera/sub2.htm>

非公式サイト

<http://www.cpm.z80.de/index.html>

ソース

<http://www.cpm.z80.de/source.html>

アーカイブなど

<http://www.nasekouki.co.jp/~shig/cpm.html>

<http://www.retroarchive.org/cpm/>

<http://www.e-tech.net/~pbetti/mirrors/oak.oakland.edu/>

<http://www.seasip.demon.co.uk/Cpm/index.html>

[http://retrotechnology.com/dri/d\\_dri.html](http://retrotechnology.com/dri/d_dri.html)

CP/M on breadboard

<http://searle.hostei.com/grant/cpm/index.html>

CP/KM web(Web 版 CP/M エミュレータ)

<http://www.recfor.net/jeans/index.php?itemid=920>

これ、欲しいかも

<http://microvga.com/uvga-images>

<http://ddllab.jp/shop/edisp/syouhin.cgi>

MMC(SDC) の使いかた

<http://elm-chan.org/docs/mmc/mmc.html>

Atmel AVR について

<http://elm-chan.org/docs/avr.html>

千秋ゼミ ISP ライター HID.aspx

AVR Wiki

<http://www34.atpages.jp/avrwiki/wiki.cgi?page=AVRWiki>

その他 AVR 関連の製作

AVR ビデオ端末の製作

AVR パラレルライターの製作

AVR 学習リモコンの製作

AVR と Z80 で TK-80

Mega64 で LED GAME

- z80pack は <http://www.autometer.de/unix4fun/z80pack/index.html> にあるようですよ。 - kyo (2011年12月31日20時22分51秒)
- 情報ありがとうございます。 - 管理人 (2012年01月01日01時19分42秒)
- Z80(NMOS) は電源を切ると、全ての信号線がハイインピーダンスになりますよ～。 - ama (2014年11月19日18時07分34秒)
- 続きです。Zilog のデータブック (<http://www.z80.info/zip/ZilogProductSpecsDatabook129-143.pdf>) の 3-130 ページの 3 番目のアンサーにバスの解放方法が載っています。 - ama (2014年11月19日18時09分48秒)
- ama さん。コメントありがとうございます。改良版 (部品を減らす工夫) ではトライステート (367) は無くなっています。詳しくは AVR のソースを見て下さい。これから同様の製作を考えている方は部品を減らす工夫の方もぜひ参照ください。さらにシンプルになっています。 - 管理人 (2014年11月19日22時16分40秒)
- 遅くなりましてすみません。HLT を使うと、他の割り込みでも HALT 状態を抜け出してしますね。 - ama (2017年02月05日18時11分18秒)
- HALT 中は INT が掛けられないように回路を組めばいいのでは。CP / M しかやらないなら関係ないですけどね。 - 管理人 (2017年02月05日21時01分07秒)