

(最終更新時間 : 2023 年 12 月 09 日 16 時 08 分 40 秒)

http://star.gmob.jp/koji/data/nucleo_sd_s.jpg 拡大

mosaku さん作の mbed 版 CP/M を試す

細かい配線は気にならないので、こういうものは買うまいと思っていたのですが、妙に安いしちょっと試してみたいことがあったので、ST Nucleo F401RE を買いました。

試したのは mosaku さんが取り組んでおられる mbed 版 CP/M エミュレータです。なかなか良いですね。mbed アプリのサンプルとしてもオススメだと思います。SD シールドも一緒に買えば、工作はピンヘッダーの取り付けのみというお手軽さ。速度感も本物の Z80 と変わらない気がします。動かしてみて気になった点をレポートします。

【2020.8.14】サイトが閉鎖されたようなのでファイルを上げておきます

【2023.03.31】mbed-os6 に対応しました

(旧) ソースファイル一式 廃止

ディスクアクセスを繰り返すうちにフリーズする

例えば、^C(ウォームブート) を 10 回繰り返すと固まる。WS を 2 回起動すると終了できない、など。

しばらく原因が分からなかったのですが、common.cpp で seek のたびに fopen し、read/write で close しているからでした。CP/M 側は何をするか分からない (seek を連続して掛けてくるかもしれない) ので、多重に fopen していたのではないかと思います。fopen を setup() の中で 1 回のみ行い、read/write では close しないようにしたら改善しました。ディスクアクセスも格段に早くなりました。

今のところ問題は起きていませんが、同様 (CP/M 側は何をするか分からない) の理由で、BZ80.cpp の cpm_WriteIO() の DMA アドレスの設定と headpos の算出は DMA 実行の直前に行う方が安全だと思います。

以下は、私的な試みなのでご参考まで。

拙作の CpmttoolsGUI に添付の IPL と BIOS を基にディスクイメージ (cpm80.dsk) を作成されたとのことなので自前のディスク (AVR と Z80 で CP / M の改良用) がそのまま使えるか試してみましたが動きませんでした。以下の変更で共用できるようになりました。

ディスクの IPL(正確にはシステムローダーと言うべきか、コールドブートローダー) が読み込まれない

ROM のような扱いで固定化されているので、読み込むように変更。common.cpp の setup() の ipl 部分。

```
// ipl
/*-----
for (uint16 i=0; i<128; i++) {
    WriteMemory(GO IPL+i, rom[i]);
}
```

```

}
-----*/
void cpmdsk_seek(uint32);
void cpmdsk_read(void);
dma = GOIPL;
cpmdsk_seek(0);
cpmdsk_read();

```

但しこの修正をすると、添付されている cpm80.dsk は IPL が入っていないので起動できなくなります。IPL が入ったディスクイメージを作り直す必要があります。

用語が曖昧で紛らわしいのですが、村瀬さんの「実習 CP/M」にブートのシーケンスが解説されており、要約すると、IPL ROM がディスクの最初のセクタ (コールドブートローダー) を RAM に読み込んで、処理をそちらに渡す。と説明されています。

avrcpm で「IPL」と呼んでいるものは「実習 CP/M」では「コールドブートローダー」に相当し、「ROM」は「IPL ROM」と称して解説されています。紛らわしいですね。

CP/M マシンも現用のパソコンも基本的には同じで、ハードウェア側 (ROM) はあくまでもディスクの最初のレコード (MBR) を読み込むだけです。これにより高い汎用性を保っているのです。「コールドブートローダー」は Windows の NTLDR に相当するものです。

I/O の修正

エミュレータの想定しているハードウェアが AKI-80 の SIO を使う仕様になっていますが、私の AVR のシリアル IF を使っているので BZ80.cpp の cpm_ReadIO() の case 項に、CONSTS(0x80) の処理を追加しました。

```

case CONSTS:
    if(BusySioZ80()) ret=0; else ret=0xFF;
    break;

```

ハードウェアの仕様変更がソフトだけでできるのがエミュレータのいいところ。半田付け作業が億劫な人にはエミュレータはうってつけですね。私は逆にソフト作りのほうが億劫です。

mbed-os6 に対応した更新【2023.03.31】

長らく放置していて久しぶりに動かしてみようと Web を見たら、オンラインコンパイラが大幅に変わっておりログインすらできない。

再度ユーザー登録して入ってみると Keil Studio Cloud と名称まで変わっており別物になっておりました。

ソースは残っていたのでインポートしたのですがコンパイルできません。

最新の mbed-os6 に合わせた修正をしました。os6 の本プロジェクトに関わる変更点は、

1. シリアル通信オブジェクトがデバッグ用? としてデフォルト (使用宣言不要、但し 9600bps 固定) になった
2. FatFs ライブラリが取込済
3. wait_ms() は廃止

元のソースはなるべく変えないポリシーで修正しました。

1. は、速度変更できないのは不便なので別途シリアルクラスを使い、修正しなくて済むようにラッ

パークラスを作って対応。

2. は使用方法が異なるので書き換えになりますが、変更が必要なのは使用宣言部分のみでした。
3. は代替りの関数をマクロ定義することで修正不要にしました。

新しい os で大きく変わっているのは、使用デバイスやピンアサイン等の情報を .json ファイルで管理するようになっていることです。

ライブラリの階層にあちこちに存在しますが基本的にはプロジェクトのルートにある mbed_app.json でオーバーライドして使うようになっており、こちらを修正してビルドすれば別のボードでも動くと思います。

あと、私みたいにこのボードをずっと放置していた人は、SWD のファームをアップデートしてください。

STM32CubeProgrammer をインストールし起動してファームが古いと警告が出たら右下の方の水色のボタン Firmware upgrade で更新できます。

関連リンク

Re: 糠を足せ (mosaku さん) サイトが閉鎖

あ～み～プロジェクト (abe さん)

<http://www.nextftp.com/abe/>

Z80 エミュレータ (TAKEDA さん)

<http://takeda-toshiya.my.coocan.jp/>

スイッチサイエンス

<http://www.switch-science.com/catalog/1619/>

<http://www.switch-science.com/catalog/317/>

秋月電子通商

<http://akizukidenshi.com/catalog/g/gM-07723/>

当サイト内

AVR と Z80 で CP / M の改良

CpmtoolsGUI

CP / M ディスクイメージ作成手順メモ

-
- ・ keil のおしゃれな環境に戸惑いました。保有ボード F446RE ですが json の変更のみで OK で、ダウンロードしたファイルのみですべて事足りました。久しぶりにワクワクさせて頂きました。 - あ～み～ (2023 年 12 月 09 日 16 時 08 分 40 秒)