

書き込みソフト「f r w 7 2 0 6 . e x e」の使い方

2 0 0 6 . 6 . 2 6

BCSH7206 CPUボードは無償ダウンロード、またはCDに添付されている「f r w 7 2 0 6 . e x e」でユーザープログラムのフラッシュROMへの書き込み「ROM_WRITE」、内部RAMへのダウンロード「RAM_WRITE」ができます。

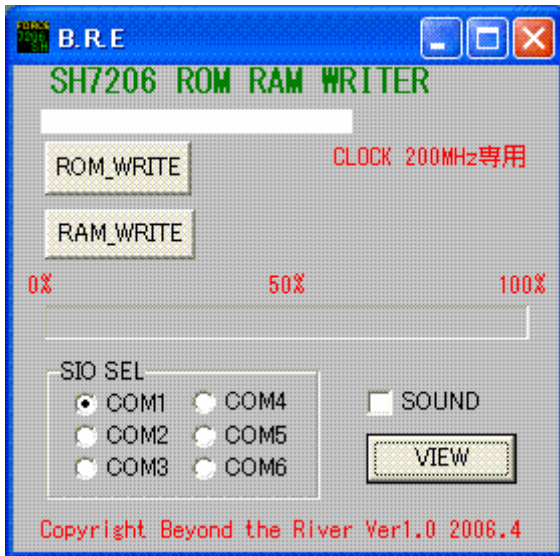


図1 「f w r i t e 7 2 0 6 . e x e」画面

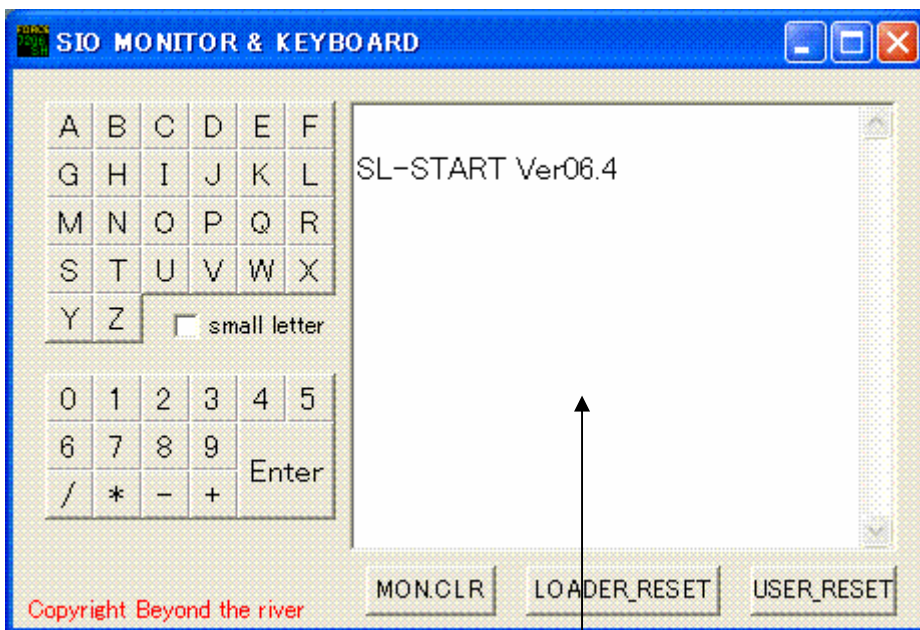


図2 「V I E W」をクリックすると現れるS I Oモニタ画面とキーボード

キー、画面の説明

【 A , B , C , , , , Z 】キーをマウスでクリックするとそのアスキーデータがS I O (シリアルI / O) より出力されます。小文字は「small letter」をチェックして下さい。

【 0 , 1 , 2 , 3 , , , , Enter 】キーをマウスでクリックするとそのアスキーデータがS I O (シ

リアルI/O)より出力されます。

【 S I O モニタ画面 】 S I Oから入力されたアスキーデータが表示されます。アスキーデータ以外はエラーとなります。エラー時は「 R i c h E d i t x xエラー」と表示されますので、xをクリックして消してください。再び表示を開始します。

【 M O N . C L R 】 S I Oモニタ画面のクリア。スクロールバーがありますので、以前の表示データもバーの操作で見ることができますが、以前のデータを消したいときに使用します。

【 L O A D E R _ R E S E T 】

あらかじめ書き込まれているローダーを動作させるクリア。ユーザープログラムのROMへの書き込み、RAMへのダウンロードの前に必ずこのローダーを使用します。 S L - S T A R T V e r x x xと表示されます。

【 U S E R _ R E S E T 】

ユーザーが書き込んだ100000H番地からのプログラムを再動作させるリセット。

ROM書き込み、RAM書き込み、J1の設定

【 ROM書き込み、動作 】

B C S H 7 2 0 6 CPUボードは動作モードの選択はありません。リセット時、外部ROMの0番地から書き込んである4バイトの数値をアドレスとして読み込み動作します。本ボードの外部フラッシュROM T C 5 8 F V T 1 6 0 の最上位アドレスA19はJ P 1の切り替えにより動作を選択できるようになっています。書き込めるファイルフォーマットはモトローラS1, S3です。(自動判別)

J P 1

ショートされるピン	動作
1 - 2	A 1 9とA 2 0接続
3 - 4	A 1 9外部コントロール *デフォルト

1 - 2番間のショートは通常の接続です。3 - 4間ショートはEケーブルで接続されたパソコン側からA19のレベルがコントロールできるようになります。これにより、ユーザーのプログラムを書き込んだ100000HからのデータをCPU側から見て0番地として実行させることができます。

物理メモリマップ(最上位がA19)	動作
1 F F F F F H 1 0 0 0 0 0 H	ユーザープログラム書き込み領域
0 F F F F F H 0 0 0 0 0 0 H	メーカープログラム書き込み済み領域 (メーカー出荷時、既にモトローラS3フォーマットローダーが書き込まれています)

T C 5 8 F V T 1 6 0は10⁵回以上の書き換えを保証しています。書き込み手順は

ユーザープログラムを作る。開始アドレスは0番地からとする。

書き込みソフト「f r w 7 2 0 6 . e x e」で自動的にオフセット100000Hが加算され、フラッシュROMに書き込まれます。

書き込みソフト「f r w 7 2 0 6 . e x e」は転送終了後、ROMのA19を1にしてからCPUボードにリセットをかけます。ユーザープログラムが0番地から動作します。ここまで自動です。

また、

Eケーブルをはずし基板単独動作とした場合でもA19は1になりますからリセット後、そのままユーザープログラムが動作します。例えば製品として出荷できます。

【 RAM書き込み、動作 】

基板に実装されているT C 5 8 F V T 1 6 0の物理アドレス0番地からは弊社によりあらかじめ「モトローラS3フォーマットローダー」が書き込まれています。このローダーは内部RAMにS3データを転

送り、終了すると内部RAM開始番地+800Hにジャンプします。RAM上でプログラムを動作させたいときに使用します。RAM上で動作させるというのは32ビットバス幅、ノーウェイトで動作させたい場合です。このCPUの最高性能が期待できる動作となります。この場合、開始アドレスは内部RAM開始アドレス+800Hにする必要があります。(書き込みソフトではオフセットしません) プログラム例は開発セットのサンプルプログラムにありますので参考して下さい。

製品としてRAM動作を行う事例としてはROMに書き込まれたプログラムをRAMに転送し、RAMアドレスにジャンプして動作する、という使い方になります。パソコンや組み込みマイコンの一部でも行われている手法で、バス幅が広く取れる以外にROMがRAMに比べて速くできない欠点を補う意味もあります。

書き込み例

【 接続 】

事前にEケーブルをパソコン、BCSH7206のCN6に接続。CN5に電源ケーブルを挿入し電源3.3Vを入れてください。黄色が+3.3V、黒がGNDです。5Vを使用すると壊れますのでご注意ください。

COMポートをご使用のパソコンのCOMに合わせてください。不明な点は別ファイルの「COM番号を調べる」をダウンロードしてご参照ください。

【 ROM書き込み、動作 】

ROM書き込みでは

ユーザープログラムの開始番地は0番地としてください。

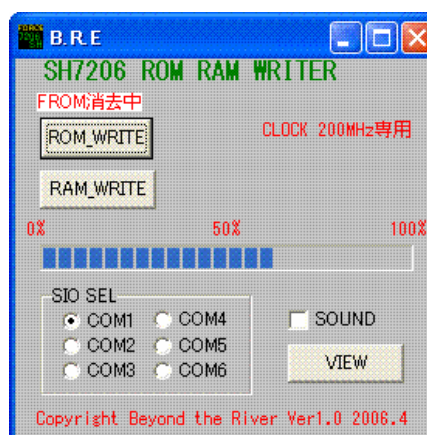
書き込めるファイルフォーマットはモトローラS1、S3フォーマットです。自動判別しますので、特にユーザーで設定を行う必要はありません。

「frw7206.exe」の「ROM_WRITE」をクリックします。

例として弊社サイトから無償でダウンロードできるsample2_7206.s(モトローラS1フォーマット)を書き込んでみます。

【 VIEWを開いておく 】

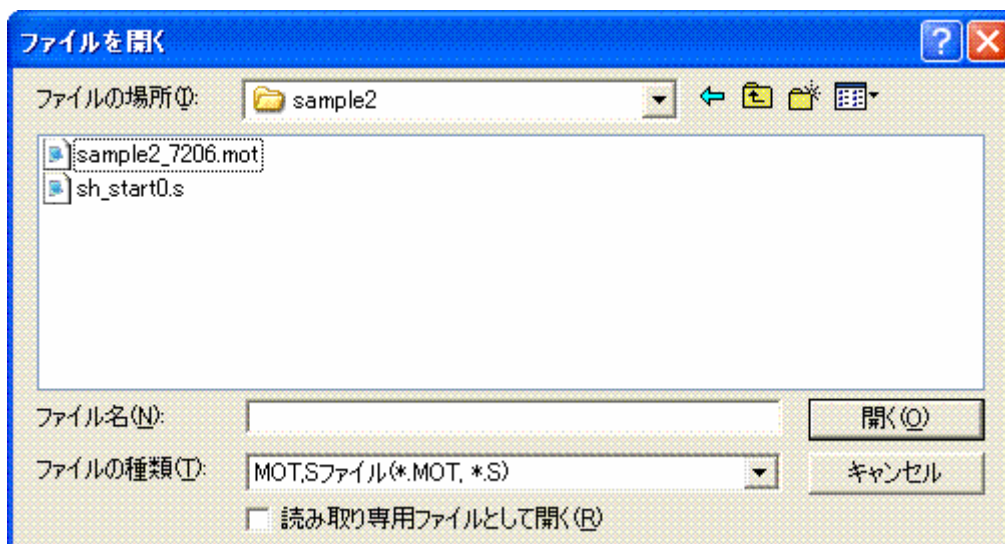
「VIEW」画面を開いて、BCSH7206に電源を入れます。正常に接続されていれば「SL-START Verxxx」とSIOモニタに表示されます。



「ROM_WRITE」をクリックします。「frw7206.exe」は自動的に外部フラッシュROM

のクリアを行います。この作業は数秒要します。

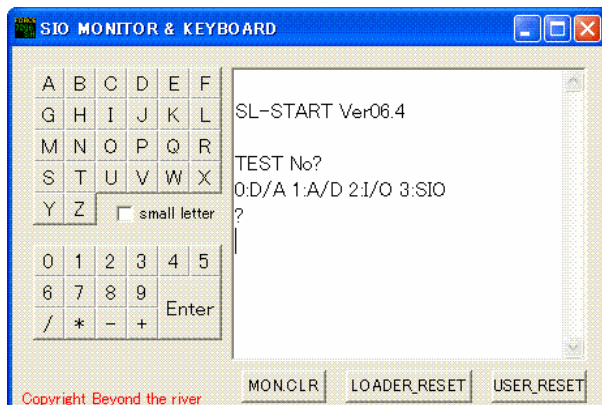
消去が終了すると書き込む「ファイルを開く」画面が表示されますので、sample2_7206.motを選択します。



正常に書き込みが行われますと「書き込み正常終了」の文字が表示されます。



また、プログラムは既に動作していますので、SIOモニタに「TEST No? ...」と表示されます。



プログラムの動作についてここでは触れませんが、これがROM書き込み、動作の方法です。弊社の他のフォースライタの特徴同様、プログラムの書き込みから、動作まで特に基板側で操作をする必要はありま

せん。

【 RAM書き込み、動作 】

ROM書き込みとの違いは以下の3点です。

ユーザープログラムの開始番地を $0 \times f f f 8 0 8 0 0$ としてください。

書き込めるファイルフォーマットは**S3フォーマット限定**です。他のフォーマットは正常に書き込みできません。

「`frw7206.exe`」の「RAM_WRITE」をクリックします。

「開発セット」のCDに添付されている `sample3_7206.s` (モトローラS3フォーマット) を書き込んでみます。このプログラムは `sample2_7206` をRAM動作用に多少の改造を行ったもので、プログラムの番地割り振りは以下のようになっています。

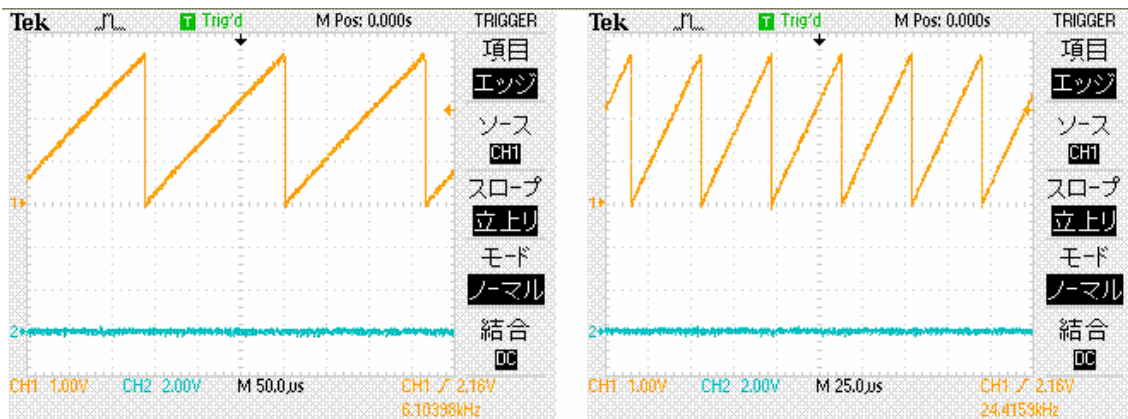
```
rom(rx)      : org = 0xffff80800, len = 64k
ram(wx)      : org = 0xffff90000, len = 63k
stack(wx)    : org = 0xffff9fff0, len = 1k
```

ROM書き込みと同じように「ファイルを開く」画面で「`sample3_7206.mot`」を選択してください。転送され、動作します。

ROM動作とRAM動作の違い

下記D/A波形は左がROM動作、右がRAM動作です。プログラムは同じソース、同じ最適化レベルです。

ROM動作が 6.10398 KHz 、RAM動作が 24.4159 KHz です。RAM動作がROM動作に比べて約4倍速くなっています。これはバス幅が2倍違うのと、ROMはウエイトが入っているためだと推測されます。ROM動作もキャッシュの設定でもっと高速にすることは可能です。



〒350-1213

埼玉県日高市高萩1141-1

有限会社ビーリバーエレクトロニクス

TEL 042-985-6982

FAX 042-985-6720

<http://beriver.co.jp>