

ポケコン用 RAM ファイルシステム



Version 4.2

Welcome! (Rev.2)

ソフトウェア名称	FX-Binary-Filer File Control SubSystem Ver.4.2
ジャンル	バイナリーファイル用 RAM ファイルユーティリティ
著作権者	©CopyRight by JUN AMANO 2003-2024
動作環境	VX-3 RAM 合計 16KB 以上
必須環境	パーソナルコンピュータでシリアルポート経由で通信できる環境
転載可否	可能です。巻末の「12. ご利用に関して」をご覧ください。
問い合わせ先	VYU04421@nifty.com

目次

1.	はじめに	3
2.	インストールする	4
3.	ファイル管理画面を使う	7
4.	コマンドラインを使う	11
5.	プログラムから利用する	12
6.	バックアップ／復元する	14
7.	アンインストールする	16
8.	コマンドリファレンス	17
9.	FBF 活用方法	26
10.	FBF で何ができるのか	28
11.	改変履歴	30

1. はじめに

カシオポケットコンピューター「VX-3」には、機械語プログラムが実行できます。
しかしながら「隠し」の存在でため、手順が統一されておらず、複雑で扱いづらい面がありました。

この問題を解消するために開発されたのが、
この「FX Binary Filer –File Control Subsystem」(以下、FBF)という「ファイルシステム」です。
「ファイルシステム」とは、データに名前を付けて保存して管理できるようにするシステムで、
このファイルシステムでは、機械語プログラム・バイナリーデータを「ファイル」として扱います。

FBF を利用することにより、機械語プログラムを実行したり、画像ファイルを扱ったりできます。

従来のアーカイブでは、システム部分だけの提供で、どちらかというと開発者向けでした。
FM 命令(ファイル管理アプリ)を組み込んだかたちの提供により、
すぐに簡単に、VX-MENU システム並みにご利用いただけるようになりました。

このような VX-3 の可能性を広げる「FBF」を、
ユーザー作成プログラムの一部として組み込む、また複数の機械語プログラムを実行したり等、
是非ご活用ください。

本ドキュメントでは以下の名称を略称で記載しています。

名称	本ドキュメントでの略称
FX-Binary-Filer File Control SubSystem	FBF
F.COM(改) VX-MENU システム (PB-1000 MENU SIMULATOR)	VX-MENU

2. インストールする

FBF をインストールするためには以下の環境が必要です。

FX-Binary-Filer (FBF)の動作環境	
対応機種	VX-3
必要メモリー	RAM 合計 16KB 以上
通信環境の必要性	必須

- (1) 既存のプログラムを保存した後に ALL RESET スイッチを押して、ポケコンをリセットしてください。

```
*****
All reset/Memory
RAM : 8KB + 32KB
*****
```

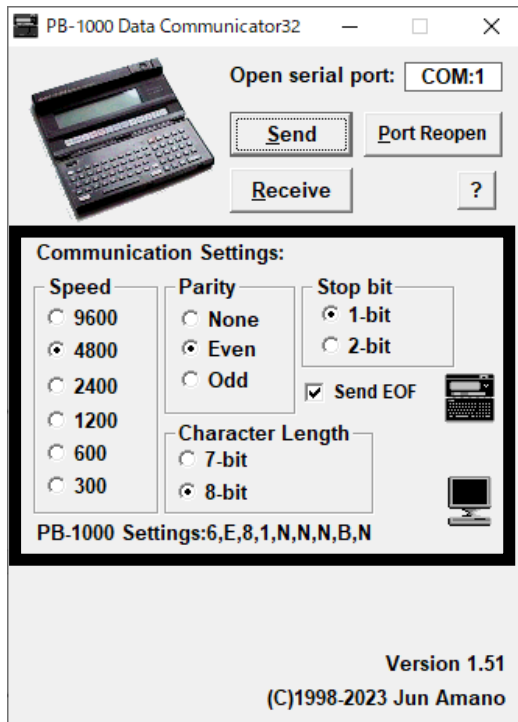
- (2) RAM 容量の合計が 16KB の場合(RP-8 増設時の VX-3)のみ、

CLEAR 1024, 2048 **EXE** を実行してください。

```
CLEAR1024,2048
```

- (3) パソコン側で、「PB-1000 Data Communicator32」を起動しておきます。

通信設定は以下のようにしてください。



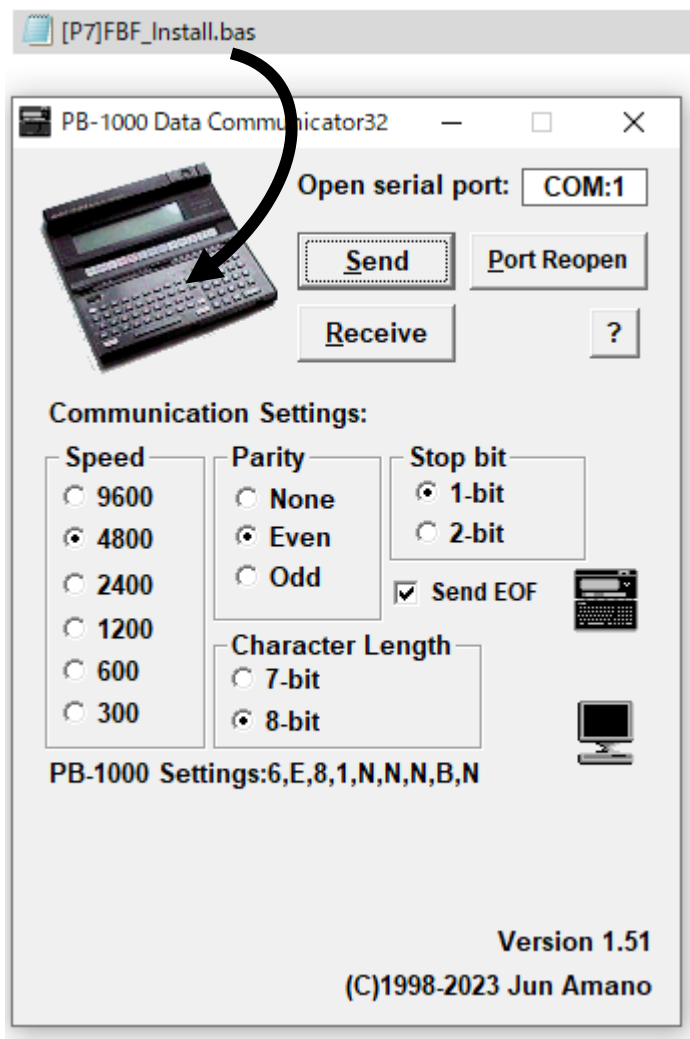
- (4) **[S]** **[F.COM]**を押して、F.COM モードにして、P7 を選択して **[L]** を押してください。

```
P 0 1 2 3 4 5 6 7 8 9 [RS232C]
F 0 1 2 3 4 5 6 7 8 9 29467B
P7>Save / Load / Merge / Print
    Edit / New / Device
```

- (5) **COM0:** の後ろの数字を 6 に変えてから、**[EXE]** キーを押します。
これで受信待機状態になります。

```
P 0 1 2 3 4 5 6 7 8 9 [RS232C]
F 0 1 2 3 4 5 6 7 8 9 29467B
P7>LOAD"COM0:6,E,8,1,N,N,N,B,N"
```

- (6) **[P7]FBF_Install.bas** を PB-1000 Data Communicator32 のポケコンの画像のところにドロップしてください。送信が開始されます。



- (7) プログラムの転送が完了すると、再び F.COM 画面に戻ります。

MODE **0** **S** **7** の順に押して、インストールプログラムを実行します。


```
P 0 1 2 3 4 5 6 7 8 9 [RS232C]
F 0 1 2 3 4 5 6 7 8 9 27242B
P7>Save / Load / Merge / Print
    Edit / New / Device
```

- (8) 以下の画面が表示されますので、

画面に表示されている環境にあったファイルをパソコンから転送してください。

送信方法は、手順(6)と同じです。

```
* FX-Binary-Filer(FBF) Install *
Please send from PC:
FBFwithFM_RP-33.fv1
```

 FBFwithFM_RP-8.fv1
 FBFwithFM_RP-33.fv1

- (9) パソコンから送信が開始されると、音が鳴って、「.」が表示されます。

```
* FX-Binary-Filer(FBF) Install *
Please send from PC:
FBFwithFM_RP-33.fv1
...
```

- (10) 転送が完了すると、ファイル管理画面が表示されます。

```
$CP.R      R      25026B
[ name  ][ kill  ][ load  ][ *sys  ]
```

- (11) ファイル管理画面は **BRK** キーを押すと、終了して CAL モードに戻ります。

再びファイル管理画面を表示したいときは、**S** **7** を押します。

(P7 エリアにはファイル管理画面を起動するプログラムが入っていますので、削除しないでください。)

- (12) 以上でインストールは完了です。さあ、FBF へようこそ！


3. ファイル管理画面を使う

ファイル管理画面を使ってみましょう。

FBF のインストーラーにはファイル管理画面(FM 命令)を組み込んでおり、機械語プログラムなどを簡単に扱えます。

(1) 画面の見かた

ファイル名	属性	空き容量
TEST	M	45198B
[name][kill][load][*sys]		


MODE キーを押すと情報が切り替わります。

ファイル名	属性	サイズ	格納番地	実行アドレス
TEST	M	100	\$0000-0063	00000
[name][kill][load][*sys]				

(2) ファイル名について

ファイル名は「.」を含めて8文字となります。

ファイル名のピリオドからは「拡張子」と呼ばれ、ファイルの種類などを示します。

1	2	3	4	5	6	7	8
S	A	M	P	L	E	.	R

(3) 属性について

ファイルの属性は以下の3種類になります。

属性	説明
M	実行できるファイル
m	実行できないファイル
R	機械語エリアの確保が不要なファイル

属性はファイルのロードアドレスから判断していますので、

実行できないファイルにも、「M」の属性が付く場合があります。

不用意に EXE キーを押して実行してしまわないように、

実行できないファイル(機械語プログラムではない画像データなど)は、ファイル名の最後に「. D」を付けておきましょう。

(4) 特別な意味を持つファイル名について

ファイル名には特別な意味を持つものがあります。

文字	機能
最後に. R (拡張子 R)	機械語エリアの確保が 不要 な機械語プログラム
最後に. r (拡張子 r)	機械語エリアの確保が 不要 な機械語ライブラリー (機械語からの呼び出し用のため、ファイル管理画面から実行不可)
最後に. D (拡張子 D)	バイナリーデータファイル(実行不可)
先頭に \$	削除禁止ファイル

(5) 意味は持ちませんが、このように名前を付けるとファイルが扱いやすいです。

ファイル名には特別な意味を持つものがあります。

文字	機能
最後に. EXE または . E	機械語エリアの確保が 必要 な機械語プログラム (詳しい説明は実行する機械語プログラムの説明書をご覧ください)

(6) ファイルの登録数について

ファイル名は 255 個まで登録できます。

(7) ファイルの選択

ファイル管理画面を起動したときは、最初に登録したファイルが表示されます。

キーを押して、操作するファイルを選択します。

(8) [name] / 名前の変更

キーを押して、新しい名前を入力して、 キーを押すとファイル名が変更されます。

```
TEST      M 100  $0000-0063 000000
new name?TEST2
```


(9) [kill] /ファイルの削除

[K] キーを押して、[Y] キーを押すとファイルを削除できます。

([Y] キー以外はキャンセル扱いになります)

ファイル削除後は、最初に登録したファイルの表示に戻ります。

※「\$」が付いたファイルを削除する際は、[name] で「\$」を省いてから、[kill] してください。

※「\$FM.R」はファイル管理画面のプログラムになりますので、削除しないでください。

```
TEST      M 100  $0000-0063 00000
kill?y/n
```

(10) [load] /ファイルの受信

[L] キーを押すと、PBF ファイル(バイナリーデータをテキストに変換したファイル)の受信画面になります。

パソコンから PBF ファイルを送信すると、ファイルが登録されます。

通信設定は、F.COM モードの設定が使用されます。

中止したい時は、[BRK] キーを押してください。CAL モードに戻ります。

```
$CP.R      R 402  $0000-0191 00000
loading pbf.
```

(11) [*SYS] / システム設定

[*] キーを押すと、システム設定画面になります。

```
$CP.R      R 402  $0000-0191 00000
[BINARY][DEVICE]
```

[B] キーを機械語エリアの設定になります。

```
$CP.R      R                               52010B
Bin Size:100
>binary ?
```

上部の「Bin Size」が現在の機械語エリアのサイズです。

確保したいサイズを入力して、**[EXE]** キーを押します。

&H6CD0 番地からの機械語エリアが確保されて、

最初のファイル名の表示に戻ります。

[D] キーを押すと、F.COM の通信設定画面になります。

設定が完了したら、**[EXE]** キーを押すと、F.COM の画面に戻りますので、

[BRK] **[S]** **[7]** の順に押して、再びファイル管理画面を起動してください。

```
BPS [9600] Parity [E] Data [8]
Stop [1] CTS [OFF] DSR [OFF]
CD [OFF] Busy [ON] SI/SO[OFF]
End [ON] MTphase[0] MTspeed[F]
```

(12) ファイル管理画面のエラーメッセージ

「 No area. 」・・・機械語エリアが確保できていません。

「 Name error. 」・・・すでに登録されているファイル名と同じ名前の PBF ファイルを受信しようとした。

4. コマンドラインを使う

FBF にはコマンドライン機能を装備しており、BASIC プログラムからも FBF を利用して、機械語プログラムを利用できます。

F.COM を起動し、パソコンから、「FBF-BASE4-VX3.BAS」を P9 エリアに転送したら、

BRK **S** **9** で実行してみましょう。

以下の画面が表示されます。

```
MODE110(BASE)_
```

「BASE」というのは変数で、ファイルシステムの実行アドレスが格納されています。

MODE110(BASE)部分に続けて、試しに以下のように実行してみましょう。

```
MODE110(BASE)BS"DISP.D",&H6343,768
```

画面には何も起こりません。 **CALC** キーを押すと、「0」と表示されます。

コマンドは結果が 0 で成功で、0 以外はエラーになります。

OUT キーを押してみると、PEEK&H67A0 と表示されます。

FBF では、コマンド実行後に PEEK&H67A0 することでファイル操作の成功・失敗を確認できます。

```
0
```

```
PEEK&H67A0
```

S **7** でファイル管理画面で、「DISP.D」というファイル名が追加されていることが分かります。

このコマンドは「&H6343 番地から数えて 768 バイト分を "DISP.D" という名前で保存せよ」という命令だったのです。

```
DISP.D    m 768  $6343-6642  000000
```

```
[name  ][kill  ][load  ][*sys  ]
```

別なコマンドを実行したい場合は、**S** **9** を押してからコマンドを入力してください。

(変数 BASE の再計算のため)

このように BASIC からファイルシステムを利用できます。

どのようなコマンドがあるかは、「コマンドリファレンス」をご覧ください。

5. プログラムから利用する

コマンドライン機能はもちろん BASIC プログラムからも利用できます。

従来機能にはない、追加コマンドのような感覚で利用できます。

ファイルシステムは機械語で作成されており、機械語プログラムは **MODE110(アドレス)** で実行できます。

このマニュアルではコマンドライン時に出てきた通り、変数 BASE に実行アドレスを代入しています。(VX-MENU と同等の呼び出し方法です)

```
10 B=(PEEK&H68CA+PEEK&H68CB*256)
+2:BASE=PEEK(B)+PEEK(B+1)*256
```

変数 BASE の算出は一度だけで構いません。

都度の計算をしたくない場合は、行 10 に加え、行 20 を追加します。

```
20 MODE110(BASE)JP
```

これで固定アドレスで呼び出しできます、

```
MODE110(&H6AD0)
```

先ほど「4. コマンドラインを使う」で作成したファイル「DISP.D」を利用してみましょう。

以下のプログラムを入力して実行してください。

```
10 B=(PEEK&H68CA+PEEK&H68CB*256)
+2:BASE=PEEK(B)+PEEK(B+1)*256
20 MODE110(BASE)JP
30 CLS:MODE110(&H6AD0)BL"DISP.D"
,&H6343
40 MODE110(&H6AD0)$DD
50 A$=INKEY$:IFA$=""THEN50
```

実行すると以下のように表示されます。

```
MODE110(BASE)BS"DISP.D",&H6343,7
68
```

ファイルを保存したときに入力したコマンドが表示されますね。

実は &H123C から &H153B までの 768 バイトは画面の表示内容となり、

先ほどの手順ではこれを保存していたのです。

上のプログラムでは、行 30 で表示データを画面に書き込み、行 40 で画面表示を行っているのです。

DISP.D を用意すればこのような画像も簡単に表示できるのです。

(パソコンを使っでの画像の作成方法は巻末をご覧ください)



(「芸夢職人ゲームパック」より)

ファイルシステムを利用することで、従来難しかった処理も簡単にできるようになります。
コマンドラインもほぼ BASIC と同じような文法で、少しでもプログラムを組んだことのある方なら
ご理解いただけると思います。

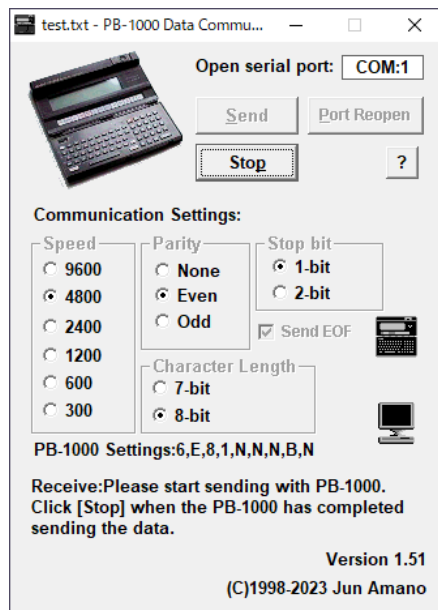
6. バックアップ／復元する

ポケコンで機械語プログラムを開発する際には暴走はつきもの。
万が一のために、ファイルシステムのバックアップを取っておきましょう。

＜バックアップ＞

- (1) PB-1000 Data Communicator32 を起動し、ポケコンに「FBF_BR4-VX3.BAS」を転送します。
- (2) PB-1000 Data Communicator32 の **Receive** をクリックします。

保存するファイル名を決めたら、**保存** をクリックします。これで受信待機となります。



- (3) FBF_BR_V4.BAS を実行します。 **B** **EXE** と操作します。

```
* FBF-BR *
B: BACKUP/R: RESTORE: ?B_
```

- (4) 任意のコメントを入力します。

```
** BACKUP **
COMMENT=BACKUP 2023/12/31_
```

- (5) パソコンへの転送が開始されます。しばらくお待ちください。

完了すると以下のメッセージが表示されます。

```
Backup completed!
```

- (6) PB-1000 Data Communicator32 の **Stop** をクリックすると受信が完了します。
- (7) バックアップしたファイルには、ファイルシステムと登録されているファイルが含まれています。メモ帳で開くと、どのようなファイルがバックアップされているか確認することができます。

```
57020,65535
;FBF file volume name:BACKUP 2023/12/31:DEBC-FFFF,0
;$CA.R,0
;$CP.R,0
;$DD.R,0
;$KY.R,0
;$FM.R,0
3730FB05015200D11CDF2A566054772153D700D71BD660D01CD1
97566054772153D640BFDEA919011930FCDF42008002611A7705
40C2DE3B1EB014D100011150401AD11CE220566054772153B737
0F93566054372153D600BFDEE80640022141001FB00C4201FF41
```

<復元>

- (1) ページ 4～6 の「2. インストールする」の手順 (1) ～ (7) を実行します。
- (2) 手順 (8) で **FBFwithFM_RP-●.fvl** の代わりに
バックアップしたファイルを送信します。
- (3) 送信が完了すると、ファイル管理画面が表示されます。

7. アンインストールする

FBF を利用しなくなった場合は、ALL RESET スイッチを押してポケコンを初期化するか、P7 エリアの 行 60020 を実行してください。

確認画面で ☐ Y ☐ EXE で、他のプログラムを保持したまま、ファイルシステムが利用していたメモリーを解放します。

```
Ready P7
RUN60020_
```

```
FBFシステムヲ アンインストール シマスカ?:Y_
```

P7 にあるファイル管理画面の起動プログラムも削除されます。

```
P 0 * 2 3 4 5 6 7 8 *      55785B
Ready P7
_
```

FBF をアンインストールした後に、
FBF を利用するプログラムを実行すると暴走するので注意してください。

8. コマンドリファレンス

コマンドライン機能の文法について説明します。

●機械語の実行のしかた

VX-3 では、以下のように機械語を実行します。

MODE110(実行アドレス)

●実行アドレスの求め方

コマンドラインにて FBF を利用する際には事前に変数 BASE を設定する必要があります。

ファイルの追加や削除が発生するたびに算出し直す必要があります。

(後述の JP コマンドで固定アドレスでの呼び出しが出来るようになります)

$B = (\text{PEEK}\&H68CA + \text{PEEK}\&H68CB * 256) + 2$: $\text{BASE} = \text{PEEK}(B) + \text{PEEK}(B+1) * 256$

●内部コマンドと外部コマンド

コマンドには「内部コマンド」と「外部コマンド」があります。

- ・内部コマンド・・・FBF 内蔵コマンド
- ・外部コマンド・・・ファイルを追加することで利用可能になるコマンド

●コマンドリファレンスについて

- (1) 書式中、下線部は必須パラメーターとなります。
- (2) 文字や数値を指定するパラメーターでは変数が利用できます。
- (3) BASIC と同様。文字列はダブルクォーテーション(")で囲みます。
- (4) 実行結果は以下で確認できます。

●実行結果について

コマンドの記述に誤りがある場合は、SN Error(文法誤り) や BS Error(パラメータ範囲外)などが発生します。

コマンドの実行結果は、以下のアドレスの値を参照することで成功／失敗を確認できます。

PEEK &H67A0 (成功:0 失敗:1)

エラーについては、別紙の「失敗要因一覧」をご覧ください。

【BL】ファイルのロード

内部コマンド

書式:MODE110 (BASE) BL ファイル名, ロード開始アドレス**解説:**

- ・指定したファイルの内容を指定したアドレス空間にロードします。
- ・ロード開始アドレスを省略した場合は、保存時のアドレスのロードされます。

例:

MODE110 (BASE) BL "TEST. D", &H0500

「TEST.D」を &H0050 番地からロードする。

【BS】ファイルの保存

内部コマンド

書式:MODE110 (BASE) BS ファイル名, 開始アドレス, サイズ, 実行アドレス**解説:**

- ・指定したアドレス空間をファイルを保存します。
- ・実行アドレスを省略した場合は、&H0000 が指定されます。
- ・実行できないファイルの場合は、拡張子(.D)を付与してください。
- ・実行開始番地が指定されていると、
ファイル管理画面でファイルを選んで **EXE** キーを押して実行できます。

例:

MODE110 (BASE) BS "TEST. E", &H0000, 10, &H0005

&H0000～&H0009 番地までの内容を「TEST.E」という名前で保存し、
実行開始番地を &H0005 とする。

【JP】ジャンプテーブルをコピー

内部コマンド

書式:

MODE110 (BASE) JP

解説:

- ・FBF を呼び出す際は、前述の通り変数 BASE を算出しますが、ファイルを増減するたびに算出することになるため、固定アドレスで呼び出せるようにするためのコマンドです。実施後は MODE110 (&H6AD0) で利用できるようになります。
- ・JP コマンドが実行されていない環境もあるため、プログラムの最初で必ず JP コマンドでジャンプテーブルをコピーするようにしてください。

例:

```
10 B=(PEEK&H68CA+PEEK&H68CB*256)+2:BASE=PEEK(B)+PEEK(B+1)*256
20 MODE110 (BASE) JP
30 MODE110 (&H6AD0) BS "TEST.D", 0, 100
40 MODE110 (&H6AD0) KL "TEST.D"
```

【KL】ファイルの削除

内部コマンド

書式:

MODE110 (BASE) KL ファイル名

解説:

- ・ファイルを削除します。
- ・ファイルを保存したら、次回 FBF 利用前に変数 BASE を算出し直してください。
JP コマンドで固定アドレス(&H1AD7)で FBF が利用できます。
- ・先頭に「\$」が付いたファイルは削除できません。
RN コマンドで名前を変更してから削除を実施してください。

例:

```
MODE110 (BASE) KL "TEST.D"
ファイル「TEST.D」を削除する。
```

【MV】ファイルの移動

内部コマンド

書式:MODE110 (BASE) MV ファイル名**解説:**

- ・ファイルをファイル保存エリアの先頭に移動します。
- ・拡張子 R のプログラム中からファイルを削除を行ってメモリーの詰め処理が実施されても、移動しないようにするためのコマンドです。詳しくは User's Manual をご覧ください。
- ・ファイル名の先頭に「\$」が付いているファイルしか移動できません。
- ・ファイル管理画面から「\$」付きのファイルを受信すると、自動的に MV コマンドが実行されます

例:

MODE110 (BASE) MV "\$DD. R"

ファイル「\$DD.R」をファイル保存エリアの先頭に移動する。

【OF】FBF のアンインストール

内部コマンド

書式:

MODE110 (BASE) OF

解説:

- ・FBF システム および 登録ファイルを削除し、メモリーを解放します。
- ・二度実行しようとする、暴走してしまいますので注意してください。

【RL】RR 形式のプログラムの実行

内部コマンド

書式:

MODE110 (BASE) RL ファイル名, ロード開始アドレス

解説:

- ・拡張子 R の実行ファイル(RR 形式プログラム)を指定アドレスにロードして実行します。
 - ・拡張子 R のファイルは実行時にファイル末尾に用意された再配置データによって、プログラムの書き換えが実施されます。
- そのため、ファイルを格納している状態で実行され、ロードする必要がないため、機械語エリアの確保が必要ありません。
- RR コマンドの場合、毎度書き換えが実施されるため、書き換えにかかる処理(時間)を省きたい場合は、RL コマンドを利用してください。

RL コマンドは任意アドレスにロードしたあと、書き換えを実施して実行します。
書き換えは一度だけで、次から指定アドレスで実行できます。

例:

1 回目: MODE110 (BASE) RL "\$DD. R", &H0000

2 回目: MODE110 (0)

ファイル「\$DD.R」を &H0000 番地にロードしてプログラムを書き換えて実行する。

2 回目からは &H0000 を実行する。(書き換えは発生しない)

注意:

必ずしもロード開始アドレスが実行アドレスとは限りません。プログラムの説明書をご覧ください。

【RN】ファイル名の変更

内部コマンド

書式:MODE110 (BASE) RN 変更したいファイル名, 変更後のファイル名**解説:**

- ・ファイル名を変更します。
- ・変更したいファイル名と変更後のファイル名を同じ名前を指定するとエラーになります。
- ・変更後のファイル名をすでに存在しているファイル名を指定した場合もエラーになります。

例:

MODE110 (BASE) RR "\$DD. R", "DD. R"

ファイル名「\$DD.R」を「DD.R」に変更する。

【RR】RR 形式のプログラムの実行

内部コマンド

書式:MODE110 (BASE) RR ファイル名**解説:**

- ・拡張子 R の実行ファイル(RR 形式プログラム)を実行します。
- ・拡張子 R のファイルは実行時にファイル末尾に用意された再配置データによって、プログラムの書き換えが実施されます。
- そのため、ファイルを格納している状態で実行され、ロードする必要がないため、機械語エリアの確保が必要ありません。
- ・ファイル名が「2文字. R」または「\$2文字. R」の場合はコマンドを簡略化できます。

MODE110 (BASE) RR "CL. R", 4095 → MODE110 (BASE) CL, 4095

MODE110 (BASE) RR "\$DD. R" → MODE110 (BASE) \$DD

※内部コマンドと同じ場合は、内部コマンドが優先されます。

例:

MODE110 (BASE) RR "\$DD. R"

RR 形式のプログラムを実行する。

【SF】ファイルの検索

内部コマンド

書式:

MODE110 (BASE) SF

解説:

- ・ファイルを検索します。

ファイルが見つかったと以下の変数に情報がセットされます。

変数名	内容
FN\$	ファイル名
ST	開始アドレス
SZ	ファイルサイズ
EX	実行アドレス
S0	格納アドレス

- ・ファイルが見つからない場合は、PEEK(&H17A7)=1 になります。

例:

MODE110 (BASE) SF"\$DD. R"

【SI】システム情報の取得

内部コマンド

書式:

MODE110 (BASE) SI

解説:

- ・PEEK (&H678D) することで、ファイルの登録数を取得できます。

例:

MODE110 (BASE) SI:PRINT PEEK (&H678D);"/ ファイルカ トロクサレタイムス"

【\$CP】ファイルのコピー

外部コマンド

書式:

MODE110 (BASE) \$CP コピーしたいファイル名, コピー後のファイル名

解説:

- ・指定したファイルを、コピー後のファイル名で複製を作成します。
- ・コピー後のファイル名がすでに存在する場合は、元のファイルは削除されます。
- ・実行結果はリザルトコード(&H67A0)で確認します。コピー失敗時は BASIC のエラーは発生しません。

例:

MODE110 (BASE) \$CP "TEST. D", "TEST2. D"

TEST. D のコピーを TEST2. D の名前で作成する。

【\$DD】画面表示

外部コマンド

書式:

MODE110 (BASE) \$DD

解説:

- ・画面の表示を行います。
 - ・表示ドットバッファ(&H6343～&H6642)にドットパターンを書き込んでも表示されません。
- このコマンドを実行することで画面に表示されます。

例:

MODE110 (BASE) \$DD

【\$FM】 ファイル管理画面の起動

外部コマンド

書式:MODE110 (BASE) \$FM**解説:**

- ・ファイル管理画面を起動します。
ファイル管理画面を利用するには、P7 エリアに MODE110(BASE)\$FM を実行するプログラムがなくてはなりません。
- ・「6. バックアップ／復元する」でも、このコマンドを利用していますので、このファイル(\$KY.R)は削除しないでください。
削除してしまっても、「FBF_BR_V4-VX3.BAS」を利用して復元できます。

「\$FM」は、あお氏の著作物です。

例:

MODE110 (BASE) \$KY "\$FM"

【\$KY】 キーバッファ入力

外部コマンド

書式:MODE110 (BASE) \$KY 文字列**解説:**

- ・32 文字程度の文字をキーバッファに送ります。
[EXE]キーは CHR\$(13) などポケコンのマニュアルのキャラクターコード表を参照してください。
- ・キーバッファは、INPUT 文やプログラムが終了時に読み込まれます。
これにより、BASIC プログラムの自己整形などができます。
- ・「6. バックアップ／復元する」でも、このコマンドを利用していますので、このファイル(\$KY.R)は削除しないでください。
削除してしまっても、「FBF_BR_V4.BAS」を利用して復元できます。

例:

MODE110 (BASE) \$KY "RUN"+CHR\$(13)

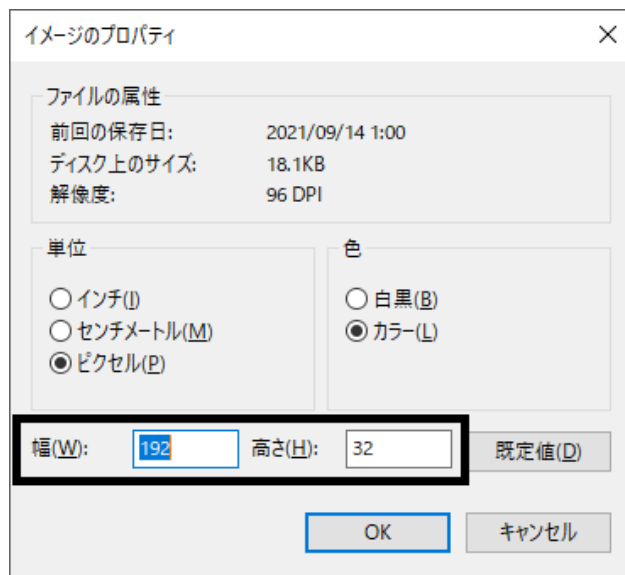
9. FBF 活用方法

「5. プログラムから利用する」でご紹介した画像の作成方法を記載します。

- (1) ペイントを開いて、[ファイル]→[プロパティ]でキャンバスのサイズを 横 192×縦 32 ピクセルに設定します。

色は作成時は「カラー」で作成してください。

最初から白黒にすると、ペイントのバグで正しく描画できません、

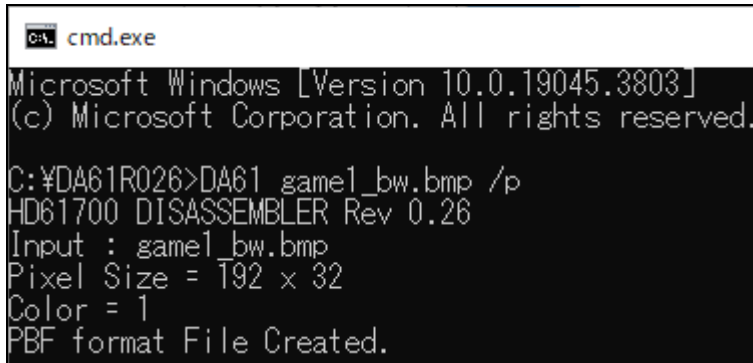


- (2) ペイントで画像を作成したら、[ファイル]→[名前を付けて保存]で、ファイルの種類を「モノクロビットマップ」にして保存します。
修正することを考えて、別のファイル名で保存することをお勧めします。

- (3) 逆アセンブラ「DA61」を使って、画像ファイルから PBF ファイルを作成します。

DA61△ファイル名△/p [Enter] を実行します。(△は半角スペース 1 個です)

※DA61 は「HD61700 SPRITS」(<https://hd61700.yukimizake.net/>)から入手できます。

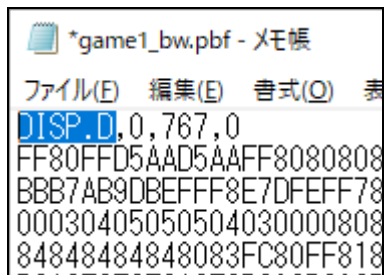


```

C:\>cmd.exe
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\>DA61 game1_bw.bmp /p
HD61700 DISASSEMBLER Rev 0.26
Input : game1_bw.bmp
Pixel Size = 192 x 32
Color = 1
PBF format File Created.
  
```

- (4) 生成された PBF ファイルをメモ帳で開いて、ファイル名の部分を DISP.D に変えれば完成です。



- (5) この PBF ファイルをポケコンにロードして、P.12 で紹介した BASIC プログラムを実行してみてください。

全画面のグラフィックが一瞬で表示されたでしょう！？

ゲームのタイトル画面などに活用しましょう。

```

10 B=(PEEK&H68CA+PEEK&H68CB*256)
+2:BASE=PEEK(B)+PEEK(B+1)*256
20 MODE110(BASE)JP
30 CLS:MODE110(&H6AD0)BL"DISP.D"
,&H6343
40 MODE110(&H6AD0)$DD
50 A$=INKEY$:IFA$=""THEN50
  
```

10. FBF で何ができるのか

FBF を導入することによって、様々な活用方法があります。

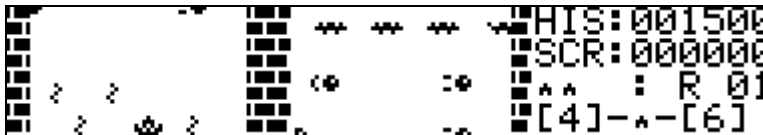
① プログラムをたくさん入れて楽しもう。

Web や雑誌に掲載のアプリやゲームをいくつも入れて、ポケコンを活用しましょう。

RAM が 64KB あるとかなりたくさんのプログラムを入れることができます。



「Calc870」 ©T.Matsu 「PB-Prog」(<http://pb-prog.sakura.ne.jp/index.html>)



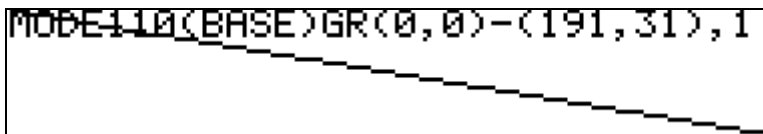
「GalaxyUFOs」 ©芸夢職人



「FRUIT FIELDS」©こたちゃん/F.F.C.「ポケコンジャーナル」掲載

② BASIC を拡張する。

外部コマンド追加して、BASIC をパワーアップしましょう。



「GR 命令」©あお / 「HD61700 SPRITS」(<https://hd61700.yukimizake.net/>)

③ 自作アプリに組み込む！

FBF を自作アプリに組み込んで利用しましょう。



「芸夢職人ゲームパック」©芸夢職人/あお



「JK」 ©Jun Amano

● FBF(FX-Binary-Filter)は無償で利用できます！ ●

1. FBF は著作権を放棄しないフリーソフトウェアになります。
2. 自作プログラムに組み込んで配布する場合は FBF のクレジット表記は不要！
(有償販売するプログラムでも OK！)
3. 自作プログラムに組み込んで配布する場合は FBF を改良しても構いません。
4. FBF そのものを転載する場合は、事前連絡が必要です。
(FBF そのものを販売できません)
5. FBF はご自身の責任において無償で利用できるフリーソフトウェアです。
利用することによって発生したいかなる損害も責任を負わないこととします。

さあ、FBF をインストールして、快適なポケコンライフを！

11. 改変履歴

バージョン	更新内容
Ver.1.3	最初の公開バージョン
Ver.1.3B	<ul style="list-style-type: none"> ・DIRECT_BIN2PBF.B を追加(FBF 自身は変更無し)。 ・マニュアルに DIRECT_BIN2PBF.B の記述追加。 ・「FBF Q&A」の A3 のアドレスCの記述を修正。
Ver.1.3C	<ul style="list-style-type: none"> ・FBF_BR.BAS を追加(FBF 自身は変更無し)。 ・マニュアルに FBF_BR.BAS に関する記述を追加。
Ver.2.0	<ul style="list-style-type: none"> ・FBF システムをファイル名指定主体動作にするように書き換え。 ・コマンドライン実行のサポート(機能追加による Ver.UP)。
Ver.2.1	<ul style="list-style-type: none"> ・コマンドラインで &H8000 以上のアドレスを指定すると、BS error が発生する現象を修正(協力 あお氏)。(FBF_INST21.BAS 以外は変更ありません)
Ver.2.2	<ul style="list-style-type: none"> ・セーブ時、DIREN アドレス更新処理に不備があったのを修正(非公開)
Ver.2.3	<ul style="list-style-type: none"> ・セーブ時、指定メモリ空間の桁あふれによるエラー判定を追加(非公開)
Ver.2.4	<ul style="list-style-type: none"> ・コマンドライン時、SN/BS/TM Error 時、リザルトコード(&H17A7)=0 なのを、エラー時は「1」になるよう統一(非公開)。
Ver.2.5	<ul style="list-style-type: none"> ・ファイル名指定ロード処理をサブルーチン化(非公開)。
Ver.3.0	<ul style="list-style-type: none"> ・RR 形式プログラム Ver.1 インターフェイスを搭載(機能追加による Ver.UP)。
Ver.3.1	<ul style="list-style-type: none"> ファイル移動機能(MV コマンド/12)を実装。
Ver.3.2	<ul style="list-style-type: none"> ・先頭に\$が付くファイルのみ削除禁止/MV コマンド可能にした。
Ver.3.3	<ul style="list-style-type: none"> ・2 文字ファイル名の際、MODE110(XXXX)nn で実行できるようにした。 ・RR 形式「.r」(CAL 呼び出し専用)機能を実装。
Ver.4.0	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・BASIC 変数代入機能(FN\$,ST,SZ,EX,SO)の実装(機能追加による Ver.UP) ・ファイル保存時、現状では無条件で既存ファイル削除を行っていたのを容量チェック後にした(既存ファイルの保護) ・CAL 呼び出しの仕様を変更(IX=0)。(BASIC からの CAL 呼び出しによる暴走の防止) ・RR 形式プログラムの仕様を変更 (RR 形式プログラム Ver.2) ・FBF 終了時に戻る際に &H1793 を 0 にするようにした。(ワークエリア不定値による誤動作の防止) ・特定条件下の MV コマンドで暴走するバグを修正。 ・SI コマンドの出力結果を &H1794 から展開するようにした。 ・SI コマンドの出力結果で &H1797,&H1798 への情報出力の廃止。 ・OF コマンドで暴走する場合があるバグを修正。 <p>[FBF-FM](旧名称 FBFFiler)</p> <ul style="list-style-type: none"> ・全面書き換えによる 1000 バイト程の削減。 ・FBFVerUP 対応による大幅な速度向上。 ・.R 実行のサポート。 ・アドレス判定機能サポート <p>[その他ツール]</p> <ul style="list-style-type: none"> ・FBF 4.0 用に若干の修正を実施。

Ver.4.0a	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・変更無し <p>[一括インストールテンプレート]</p> <ul style="list-style-type: none"> ・アンインストール機能の追加 <p>[FBF CAL 実行用テンプレート]</p> <ul style="list-style-type: none"> ・実行後、リザルトコードに合わせてキャリーフラグの変化するようにした。 <p>[FBF.BR]</p> <ul style="list-style-type: none"> ・64KB 環境でリセット直後、復元しようとする、BSError が発生する現象を修正 <p>[各種マニュアル]</p> <ul style="list-style-type: none"> 資料の更新、ロゴの変更。
Ver.4.1	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・ファイルが 1 つしか登録されていない場合での MV コマンドにおいてリザルトコード=0 になるようにした。 ・変数代入部分のコードの省サイズ化。 <p>[FBF-BIN]</p> <ul style="list-style-type: none"> ・FBF-BINxx.BAS の添付中止。FM2Bin.b の添付開始。
Ver.4.1a	<p>[FBF 本体] (内部バージョン 4.1.1)</p> <ul style="list-style-type: none"> ・コード整理により若干の容量削減を行った(機能・動作面は完全同一)。 ・FBF インストーラーの変更(あお氏提供)。 <p>[PBF2Bin.b]</p> <ul style="list-style-type: none"> ・名称の変更(FM2Bin.b → PBF2Bin.b)
Ver.4.2	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・ジャンプテーブルコピー領域を &H1684 → &H1AD7 へ変更。 <p>[FBFInst.b]</p> <ul style="list-style-type: none"> ・インストール時にジャンプテーブルコピーするよう変更。
Ver.4.2d	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・変更無し <p>[FBF パッケージ]</p> <ul style="list-style-type: none"> ・FM 命令をプリインストールしたボリュームイメージでの提供を開始。 ・RR サンプル(F1.R/F2.r/F3.r)が現行バージョンで動作しなかったため、修正。 ・ドキュメントの刷新。
Ver.4.2e	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・変更無し <p>[FBF パッケージ]</p> <ul style="list-style-type: none"> ・VX-3 向け FBF の公開開始。 ・ドキュメントの刷新。
Ver.4.2f	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・変更無し <p>[FBF パッケージ]</p> <ul style="list-style-type: none"> ・インストールイメージ中の \$FM 命令を更新 ・PBF2Bin.b を更新。名称を FBF-BIN4.BAS に変更。
Ver.4.2g	<p>[FBF 本体]</p> <ul style="list-style-type: none"> ・変更無し <p>[FBF パッケージ]</p> <ul style="list-style-type: none"> ・ドキュメントの誤りを修正。