

MSX2用モード2スプライト支援ツール

S・S・T

—Super Sprite Tool—

使用説明書

(C) 1992,2018 PSYZANS

「S・S・T」 クイックリファレンスー操作早見表

起動方法

RUN"SST.UTY"

基本操作

| | |
|-----------|---------------------------------|
| カーソル | カーソル移動・アイコン/パターン選択・アニメパターン登録/解除 |
| スペース | 決定・ドット消去 |
| リターン | コマンド選択モードに入る |
| Z・X・C | ドット配置(Z:1枚目,X:2枚目,C:ORカラー) |
| HOME/CLS | 全消去 |
| CTRL+STOP | 終了(エディットモード時のみ) |

コマンド一覧

| | | |
|---|--|---|
|  エディットモードに戻る |  上下反転 |  右シフト |
|  色指定 |  左右反転 |  アニメーション確認 |
|  パターンゲット |  上シフト |  プログラムセーブ |
|  調色 |  下シフト |  途中セーブ |
|  コピー |  左シフト |  ロード |

「偽SC4」 使用法

"SSTNSC4.UTY"ロード後、MERGE"(ファイル名).DAT" でDATファイルを結合。

1040行の「PT=15」の数値をDATファイルに含まれるパターンの数に変更して実行。

スプライトは以下の形式に従って表示する。

PUT SPRITE <面番号>,(X座標,Y座標),0,<スプライト番号>:A=USR1(面番号)

「偽SC4」USR関数

A=USR1(面番号) :指定した面番号のスプライトに着色する
A=USR2(面番号) :指定した面のECビットをONにする
A=USR3(0) :PCGカラーを初期化する。

はじめに

このたびは、PSYZANS謹製MSX2用モード2スプライト作成・使用支援ツール「S・S・T」(エス・エス・ティー)をダウンロードしていただき誠にありがとうございます。当ツールを活用するため、ご使用前にこの説明書をよくお読みください。お読みになった後も大切に保管してください。

MSXの大きな特徴として、スプライト機能があります。背景に干渉せず簡単にキャラクターを表示して動かせるこの機能は、たとえばゲームで自機の表示に使ったり、ツールのカーソルに使ったり等、プログラムを自作する際、非常に重宝します。

MSX2ではさらに機能が強化され、スプライトモード2が追加されました。このモードでは1ラインごとに1色の指定ができるようになっていました。さらに複数枚を重ねれば、各ラインの色をORした色を使うこともできます。モード2でスプライトの表現力は格段に上がりましたが、定義が面倒なこと、デザインの手間が複雑になるのと、SCREEN4以降でしか使えないのが大きな欠点です。

カラフルなスプライトをもっと手軽にデザインしたい。もっと手近に扱いたい。そんなご要望にお応えするのがこの「S・S・T」です。

「S・S・T」の主な特徴は以下のとおりです。

2枚重ね合わせに完全対応

モード2スプライトは2枚重ねで飛躍的に表現力が増します。「S・S・T」はひとつのエディット画面で2枚を同時にエディットする仕様です。凝った重ね合わせスプライトが簡単に作れます。

SCREEN1でスプライトモード2が扱える

データローダープログラム「偽SC4」を使用することで、SCREEN1でSCREEN4相当のスプライトが表示可能。SCREEN1のテキスト・PCG表示性能はそのままに、強化されたスプライトが扱えます。

シンプルで直感的な操作性

各コマンドはアイコン選択式。直感的な操作が可能です。2枚重ね合わせを前提とした設計ながら、操作性は1枚用のスプライトエディターと大きく変わらず、エディット結果は画面上に即反映されます。しかもほとんどの操作がカーソル・スペース・リターンキーだけで完結します。

アニメーション確認機能付

複数のスプライトパターンを順繰りに表示するアニメーション確認機能付。動きのあるスプライト作成に便利です。

アスキー形式での出力をサポート

作成したスプライトデータはアスキー形式のDATA文として出力可能。BASICで簡単に読み出せます。「偽SC4」とMERGEするだけで、BASICですぐにご利用いただけます。

スプライトモード2をもっと身近に！ゲームプログラミングやツール作成に。あなたの創作の手助けに。MSXを今よりもっと使いこなしたい方、どうぞこの「S・S・T」をご利用ください。

目次

| | |
|----------------------------|-------|
| はじめに | 2 |
| 目次 | 3 |
| 動作条件・構成ファイル・実行方法・注意事項と免責事項 | 4 |
| スプライトエディター「S・S・T」 | 5-9 |
| 画面説明 | 5 |
| 使用方法 | 6 |
| エディットモード | 6 |
| コマンド選択モード | 6 |
| 各コマンドの説明 | 7 |
| ツールの終了 | 9 |
| データローダー「偽SC4」 | 10-11 |
| 「偽SC4」でできること | 10 |
| 「偽SCREEN4」とは | 10 |
| 「偽SC4」使用方法 | 11 |
| スプライトを表示する | 11 |
| 勝手にFAQ | 12 |
| 技術資料 | 15-18 |
| VRAMマップ | 15 |
| DATファイルのデータ構造 | 16 |
| 機能ビットについて | 17 |
| プログラマからひとこと | 19 |
| 後記 | 19 |
| 参考文献 | 21 |

動作条件

MSX2以降 (turbo Rは標準モード推奨)
RAM32K・VRAM64K以上 要ディスクドライブ
(SANYO PHC-23+SONY HBD-20Wで製作)

構成ファイル

- SST. UTY :スプライトエディター「S・S・T」本体プログラムファイル
- SSTNSC4. UTY :データローダー「偽SC4」プログラムファイル
- SSTSC5. UTY :スクリーン5用スプライトローダーサンプルファイル

- SST. DSK :エミュレーター用ディスクイメージ。上記3ファイルを収録
- SST取説. PDF :使用説明書。当ファイル。読むにはPDFビューアが必要です。

実行方法

適当なフロッピーディスク/ディスクイメージ (以下ディスクと略) にSST. UTYとSSTNSC4. UTYをコピーして、作業用ディスクを作ってください (アーカイブのSST. DSKには上記2ファイルが収録済みですので、エミュレーター上でそのままお使いいただけます)。

作業用ディスクをカレントドライブに挿入し、BASICのコマンドラインから

RUN"SST.UTY"

を実行すればエディターが起動します。オールBASICなので起動まで少し時間がかかります。

スプライトを作るだけならSST. UTYのみで可能ですが、作ったデータをSCREEN1で利用したい場合はSSTNSC4. UTYが必要です。SSTNSC4. UTYの使用方法については「データローダー『偽SC4』」の項で説明いたします。

注意事項と免責事項

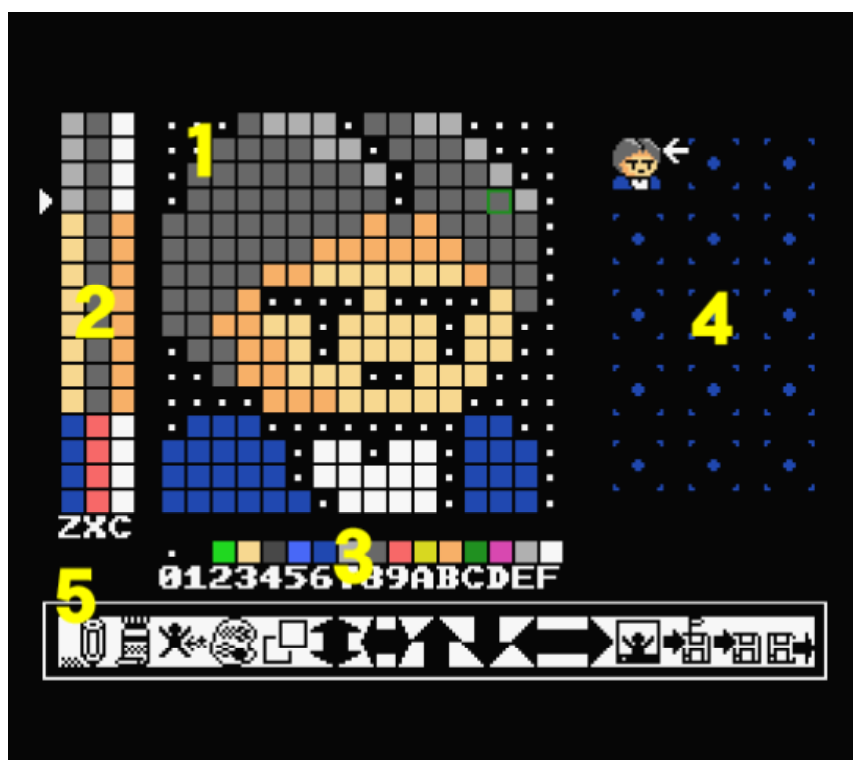
当ツールで製作されたスプライトはご自由にお使いください。「偽SC4」もご自作のソフトに組み込んでいただいて構いません。

ただし「S・S・T」エディター自体の著作権は開発元PSYZANSに帰属します。エディターの有償配布や、改変したエディターを権利者PSYZANSの許可なく配布する等、開発元の権利を侵害する使用方をなさらぬようお願いいたします。

また、当ツールを使用して発生した事故・犯罪・損害・損失について、当方はいっさい責任を負いませんのであしからず。

スプライトエディター「S・S・T」

画面説明



1・エディットエリア

ここでスプライトを描画します。16x16ドットサイズのスプライトが作成できます。作成中のスプライトパターンはここに拡大されて表示され、エディットの結果は隣のスプライトテーブルに即反映されます。1つのスプライトパターンは2枚のスプライトより構成され、それぞれを同時にエディットする仕様です。

2・ラインカラーエリア

各ラインに現在指定されている色を表示しています。Zが1枚目、Xが2枚目、Cは2枚を重ね合わせた部分の色(=1枚目と2枚目の論理和をとった色)です。

3・カラーテーブル

MSXの基本色16色を表示しています。色を選ぶ際に使います。

4・スプライトテーブル

エディットしているスプライトパターンが表示されます。一度にエディットできるのは最大15パターン(30枚)まで。この上限値は同時表示数制限ゆえです。エディット中のパターンの右側には、マーカーとして「←」が表示されます。

5・コマンドアイコン

各コマンドのアイコン。各種操作はこのアイコンで選びます。

使用方法

「S・S・T」にはふたつのモードがあります。ひとつはスプライトを描画するエディットモード、もうひとつはスプライトに対する操作等をするコマンド選択モードです。これらを切り替えながらスプライトを作成していきます。

エディットモード

カーソルキーでエディットエリア内のカーソルを4方向に移動します。Z, X, Cキーで対応する色のドットをカーソル位置に配置します。スペースキーでカーソル位置のドットを消します。HOME/CLSキーで全消去します。全消去すると各ラインの色配置も初期化されます。

リターンキーでコマンド選択モードに入ります。コマンド選択モードでは色やパターンの選択、データの入出力といった操作が可能です。

モード2の仕様では、あるスプライトのCCビットが1の場合、そのスプライトより優先順位が高く、かつCCビットが0で最もこのスプライト面に近いスプライトとの優先順位が等しくなります。それら優先順位の等しいスプライトが重なり合った場合、重なった部分の色は、それらの色の論理和をとったものとなります。

つまりモード2ではスプライトを2枚重ねた場合、1ラインにつき最大3色が使えます。「S・S・T」では2枚のスプライトを同時に扱うことにより、凝った多色スプライト作成を可能としています。

コマンド選択モード

各コマンドはアイコンで表示されています。カーソルキー左右でアイコン上のカーソルを移動し、スペースキーでカーソル位置のアイコン=コマンドを実行します。各コマンドについては次項で詳述します。

エディットモードに戻るときは、一番左の「鉛筆」のアイコンを実行してください。

各コマンドの説明



鉛筆/エディットモードに戻る

エディットモードに戻ります。コマンド選択モードが終わるときに選びます。



絵の具/色指定

スプライトの各ラインの色を指定します。

実行すると、カラーテーブル上にカーソルが表れます。カーソルキー左右でカラーテーブル上のカーソルを動かして色を選びます。カーソルキー上下でラインカラーエリアのマーカーを動かしラインを指定します。Z, Xキーで、選択中の色を指定中のラインに割りあてます。Zキーは1枚目のスプライト、Xキーは2枚目。1枚1ラインにつき1色の指定が可能です。コマンド終了はスペースキーです。

Cは1枚目と2枚目が重なる部分の色で、ZとXをORした色が自動で割り当てられます。Cが任意の色になるようにZとXの色を決めるのが、色選びのポイントです。



人/パターンゲット

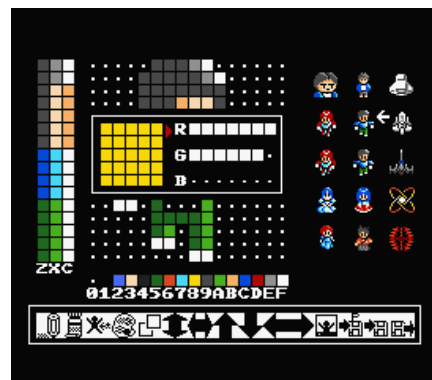
エディットするパターンを変えます。カーソルキー上下でスプライトテーブル上のマーカーを動かし、エディットするパターンを選びます。スペースキーを押すと、選んだパターンがエディットエリア上に読み込まれます。



パレット/調色

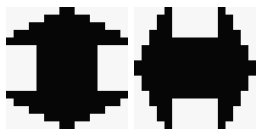
カラーパレットを変えます。カーソルキー左右でカラーテーブル上のカーソルを動かし、パレットを変えたい色を選んでスペースキーを押してください。選んだ色のカラーパレット数値がバーメーターで表示されます。カーソルキー上下でRGBを選び、カーソルキー左右でバーメーターを増減します。好みの色ができたらスペースキーで確定です。

パレットの変更は即座に画面に反映されます。ただし画面表示が乱れないよう、0番透明色と15番白色のパレットは変更できなくしてあります。



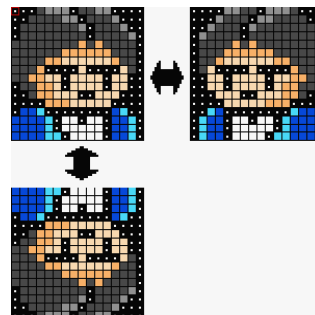
四角2枚/コピー

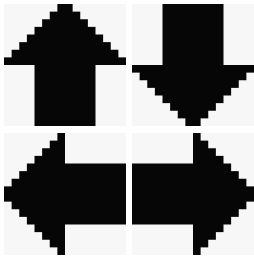
エディット中のパターンをコピーします。実行するとスプライトテーブル上にカーソルが表れます。カーソルキー上下でコピー先のパターンを選び、スペースで決定すると、選んだところにエディット中のパターンがコピーされます。



反転矢印(上下・左右)/上下反転・左右反転

エディット中のパターンを反転します。実行するとエディット中のパターンが指定した向きに反転されます。計算処理がBASICゆえ速度が遅いのはご勘弁を。





矢印(上下左右)/上下左右シフト

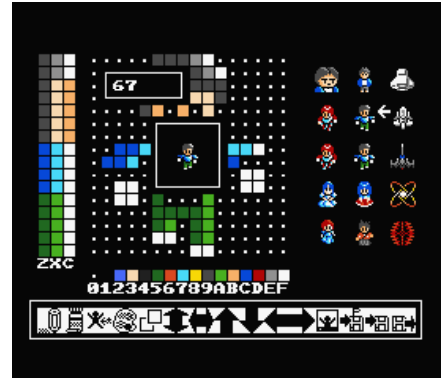
エディット中のパターンをずらしします。実行すると選んだアイコンと同じ方向に、パターンが1ドット分移動します。なお、エディットエリアからはみ出た分は消えてしまうのでご注意ください。



テレビ/アニメーション確認

複数のパターンを連続的に表示します。実行すると画面上に、カーソルとアニメーションテーブルが表示されます。カーソルキー上下で表示させたいパターンを選び、カーソルキー右でアニメーションテーブル上に登録します。同じく左で登録抹消します。表示させたいパターンを全て選んだ後、スペースキーを押すと、登録されたパターンが連続で表示されます。もう一度スペースキーを押すと表示を終了します。何も登録しない状態でスペースキーを押すとキャンセルしてコマンド選択に戻ります。

テーブルには最大5セットのパターンが登録できます。いくつかのスプライトによるアニメーションパターンを作るときに便利な機能です。



P→フロッピー/プログラム形式でのセーブ

指定したパターンを、BASIC上で読み出せるDATA文として記録します。実行したらまず、どこからどこまでのパターンを記録するかを指定します。スプライトテーブル上に右向きの指標が現れるので、カーソルキー上下で記録したい冒頭のパターンを選択し、スペースキーで決定してください。続いて同様に記録したい末尾のパターンを選択・決定してください。(指定中にリターンキーを押すとキャンセルしてコマンド選択モードに戻ります。)

指定後、画面上に表示されるソフトウェアキーボードで任意のファイル名を入力してください(カーソルキーで文字選択、スペースキーで確定。最大8字まで)。ソフトウェアキーボード上のリターンを選択・確定するとセーブされます。何も文字を入力しない状態でリターンを選択・確定するとキャンセルしてコマンド選択モードに戻ります。データは挿入中のディスク上に記録されます。ディスクはあらかじめ書き込み可能な状態にしておいてください。

この形式でセーブすると、「.DAT」という拡張子のファイルが生成されます。このファイルはMSX-BASICよりLOAD命令で読み込みできます。BASIC上で使う方法やデータの書式については「偽SC4」の項で説明します。



→フロッピー/途中セーブ

作成中のスプライトデータを保存します。実行すると「プログラム形式でのセーブ」同様にファイル名の入力を求められますので、同様にセーブしてください。なお、このコマンドではエディット中のスプライトが全て保存されるため、パターン指定は不要です。途中セーブのデータは拡張子「.COL」「.PAL」「.SPR」のファイル3つより構成されます。

プログラム形式でも同様ですが、セーブ時にディスク上に同一のファイル名が存在する場合、なんのこともなしに上書きします。更新の際はご注意ください。なお、Disk offlineやFile not found等のエラーが発生しても停止しない=データがパーにならない親切仕様です(おい)。

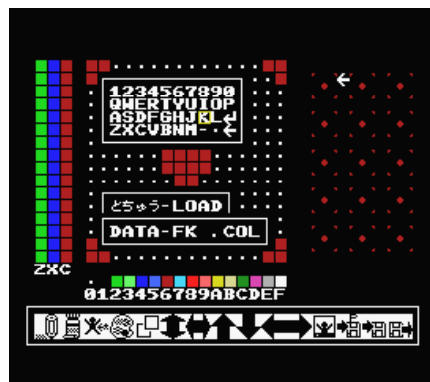
フロッピー→/ロード



保存してあるスプライトデータを読み込みます。実行したら同様に読み込みたいデータのファイル名を入力してください。FILES機能はないため、保存したデータのファイル名はあらかじめどこかに控えておいてください。読み込むとそれまでエディットしていたデータは消えます。それとデータが入ったディスクをセットしておくこともお忘れなく。

なお、このコマンドではDATファイルこと、プログラム形式でセーブしたデータは読み込めません。作成中のデータは途中セーブコマンドでセーブするようにしてください。

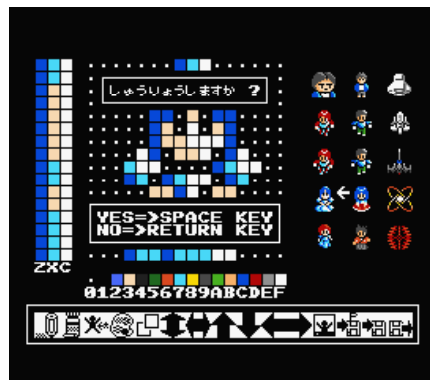
ついでに、セーブ/ロード時のソフトウェアキーボードは、開発にあたり荒井兄がこだわったところのひとつです。出来が良いと思ったら褒めてやってください(おい)。



ツールの終了

エディットモードにおいてコントロールキー+STOPキーを押すと(コマンド選択モード時に押した場合はエディットモード復帰時に)ツールを終了します。終了前に確認を取りますが、終了するとエディット中のデータが失われます。消えたら困るデータは必ず事前にセーブしてください。

プログラム本体はBASICで組まれています、エディット中のデータ消失対策として、Breakができないようにしてあります。



データローダー「偽SC4」

スプライトモード2では従来に比べて表現の幅が大きく広がりました。ただしSCREEN4 (GRAPHIC 3) 以降でなければ使えないという大きな制約があります。SCREEN4以降でのソフト製作、特に美しくかつ高速な画面表示を実現するには、相当な技術力が要求されます。

「S・S・T」では、強化されたスプライトをより気軽に利用できるよう、本来対応していないSCREEN1において、モード2相当のスプライトを表示できるようにしました。そのための設定とスプライト定義をするためのユーティリティーソフトが「偽SC4」(にせエスシーフォー)です。

「偽SC4」でできること

「偽SC4」の機能は大きく以下の3つです。

- ・「偽SCREEN4」の設定

VDP操作により、SCREEN1でSCREEN4相当の画面表示を可能にします。

- ・「S・S・T」で作成したスプライトの定義

「S・S・T」エディターで作成したスプライトをBASIC上で使えるようにします。

- ・スプライト表示用USR関数の提供

モード2スプライト表示を補助するマシン語サブルーチンを提供します。

「偽SCREEN4」とは

MSXのSCREEN1では、VDPの状態のみをGRAPHIC2モードに切り替えることで、SCREEN1の文字表示機能はそのままに、SCREEN2相当の画面表示が可能になります(「多色刷り」「スクリーン1.5」

「ニセスクリーン2」等々、各種雑誌媒体でその呼称は異なりますが、どれもこのテクニックを指しています)。

SCREEN1ではVRAMのカラーテーブルを書き換えることで、文字色や背景色を変えることができます。通常は連続する8文字ごとに文字色と背景色を指定できるのみですが、モードを切り替えることで、SCREEN2と同じく1文字横8ドットにつき2色が使えるようになります。

特にこの恩恵を受けるのが、MSXでのゲーム作りに欠かせないPCG機能(Programable Character Generator: パターンジェネレーターテーブルとカラーテーブルを書き換えて任意のBGを作る機能)です。SCREEN1の文字表示の手軽さで、カラフルな画面づくりが可能となります。

MSX2で追加されたSCREEN4は、SCREEN2と同等の画面表示機能に、スプライトモード2の表示機能が加わったものです。「S・S・T」はVDPに操作を加えた上、VRAMを操作するマシン語サブルーチンを組むことで、SCREEN1でSCREEN4相当の画面・スプライト表示を可能としました。つまりSCREEN1で、SCREEN2相当のPCGと、SCREEN4相当のスプライトが使用できるのです。

この強化されたSCREEN1をここでは「偽SCREEN4」と呼ぶことにします。

「偽SC4」使用方法

使用にあたっては、事前に「S・S・T」でスプライトデータを作成し、プログラム形式でセーブしておく必要があります。このデータをDATファイルと称します。DATファイルには拡張子「.DAT」が付いています。

まずはBASICを起動し、LOAD命令でSSTNSC4.UTYをロードしてください。その後、DATファイルが入ったディスクをドライブに挿入し、BASICのコマンドラインからMERGE命令を実行し、ターゲットとなるDATファイルを取り込んでください。MERGE命令の書式は以下のとおりです。

MERGE"(ファイル名).DAT"

DATファイルを取り込んだら、1040行の「PT=15」の「15」の部分、DATファイルに含まれるパターンの数に置き換えます。

ここで注意していただきたいのは、PTの数値はスプライトの枚数ではなくて、定義するパターンの数であることです。たとえば、8つのパターンは最大16枚のスプライトから構成されますが、PTの数値は16ではなく、飽くまでパターンの数こと8となります。

書き換えたらあとはRUNするだけ。これで偽SCREEN4で自作スプライトを使う準備が整います。

「偽SC4」はご自由に自作プログラムに組み込んでお使いいただけます(自作プログラム本体は、50行から990行の間に記述することになります。行数が足りなければ適宜RENUMしてください)。どうぞ皆さんの創作にお役立てください。

スプライトを表示する

「偽SC4」を使ってBASIC上からスプライトを表示する場合、基本的にはPUT SPRITE命令を使用します。その書式は以下のようになります。

PUT SPRITE <面番号>,(X座標,Y座標),0,<スプライト番号>;A=USR1(面番号)

スプライトの色は0(透明色)を指定してください。着色処理は続くUSR1関数が受け持ちます。

「偽SC4」で定義したスプライトの番号は、データ先頭から順に0, 1, 2, …となっており、2n番と2n+1番(n:0以上の整数)がペアとなって1パターンを構成しています。重ね合わせをする場合は、同じ座標にペアのスプライトを同様に表示して、USR1で着色してください。表示するスプライト面は1枚目より優先順位が低くなければなりません。通常は1枚目の直下にしておきましょう。

通常スプライトモード2では、COLOR SPRITE/COLOR SPRITE\$命令で、色をはじめとする各種設定をしますが、これら命令はSCREEN1では使用できません。

「偽SC4」実行直後、各スプライト面には面番号と同じスプライト番号の色が設定されています。一度着色した後は、同じ面に対して再びUSR1を実行するまでその設定が残ります。表示面に対して着色済みのスプライトを同じ面に表示させる場合、USR1の実行は省いて構いません。

スプライトを画面左端、X座標-1~-32の範囲に表示させると、ECビットの設定上、正常な位置に表示されません。これを相殺するための関数がUSR2です。スプライトのX座標が-1~-32になる場合、A=USR2(面番号)も併せて実行してください(X座標が0以上の場合は無効になります)。

ただしX座標がマイナスになった瞬間にUSR2を使用するとちらつきが目立ちます。USR2関数でセットされたECビットは、同じ面に対してUSR1関数を実行するとリセットされます。

勝手にFAQ

Q1・作ったスプライトをSCREEN4以降で使いたいのですが

「S・S・T」で作成したデータは、MSX-BASIC上で利用できるDATA文形式で出力できます。このデータをSCREEN4以降で使うためには、READ命令で読み出した上、SPRITE\$やCOLOR SPRITE\$等の命令で適宜定義してやるか、直接VRAMの各種テーブル上に書き込む必要があります。いちおうサンプルプログラム(SSTSC5.UTY)を作ってみましたのでどうぞご参考に。

もっとも、途中セーブのデータをBLOADのオフセット指定を使い、直接VRAMに読み込む方法が一番手っ取り早いのではないかとおもいます(おい)。

Q2・プログラム中でスプライトの色を変えたいのですが

全部を単色にする場合は、PUT SPRITE命令で色を指定後、A=USR1(面番号)を実行してください。この場合該当するスプライト面の各機能ビットはOFFになります。

1ライン単位で色を変えたい場合は、VRAM上のスプライトカラーテーブルかカラーデータを書き換えるという手順が必要になります。スプライトカラーテーブルはVRAMの1C00H番地以降に配置されており、スプライト面ひとつにつき16バイト分のデータから構成されます。VPOKE命令で対象となる箇所のカラーテーブルを書き換えるというのが、一番手軽でしょう。この方法ではUSR1関数による色つけが必要ありません。

「S・S・T」で作成したスプライトパターンのカラーデータはVRAM4000H番地以降に格納されています。こちらを書き換えると作成したパターンそのものの色を変えることができます。ただしUSR1関数で色を付ける際のマスターテーブルとして利用していますので、ここを弄ると元の色に戻せなくなります。

「ダメージを受けた際、主人公が一瞬点滅する」「無敵になると自機が光る」といった、一時的にスプライトの色を変えるような使い方をする場合は、一番目か二番目の方法がよいでしょう。「パワーアップや装備によって色が変わる」といった使い方には三番目が合っているとおもいますが、この場合、マスターテーブルにデフォルト値を書き戻す処理を、どこかに入れておいた方がよいでしょう。

なお、スプライトモード2での着色は、飽くまでスプライト面に対してのものです。スプライト面に対する色の設定を変えない限り、表示するスプライトを変えても前と同じ配色で表示されてしまいます。対して「偽SC4」のUSR1関数は、指定したスプライト面に表示中のスプライト番号を参照し、対応するスプライトの色データを自動で拾って着色します。表示面を気にする必要がない便利仕様ですな。

Q3・ON SPRITE GOSUB等のスプライト割り込み命令は使えますか

結論から言うと使えますが制約を受けます。「S・S・T」で作成したスプライトデータは、重ね合わせのため奇数番号のスプライトのCCビットが0Nになっています。そのため衝突するスプライトの組み合わせによっては割り込みが発生しません。

また、ICビットが0Nになっているラインに対しては、割り込みが無効になります。

Q4・重ね合わせをしないスプライトを作りたいのですが

「S・S・T」エディターでも、1枚のみで重ね合わせを想定しないスプライトは作成できます。ただし出力が1パターンごとスプライト2枚単位であるため、1枚のみのデータを作ると、同じパターンに空白のスプライトデータも出力されます。つまりスプライトとカラーデータが1枚分無駄になってしまいます。

そういうわけで1枚パターンを多数作りたい場合には、手動で空白のデータを消去して、データを詰めるという作業が必要になるでしょう。面倒かけてごめんなさい。

Q5・2枚重ねをしたのに正常に表示されないぞ！

まずはパターンを構成する各スプライトの表示面を疑ってください。同じパターンであっても、表示面の優先順位を誤ると正常に表示されません。パターンの上位側は常に優先順位の高いスプライト面に表示させてください。

次に表示後にUSR1関数で着色しているか確認してください。表示面を変えた場合など、前の色設定や機能ビット設定が残っていると、正常表示されません。

スプライト面の優先順位についていちおう説明しておきますと、MSXシリーズには32枚のスプライト面が存在します。それぞれ0から31までの番号が付けられており、若い番号の面ほど表示の優先順位が高くなります。複数のスプライトを重ねた場合、若い面のものほど上に表示されます。制限枚数以上に横に並べた場合、優先順位の低いものから消えます。

パターン下位側のスプライトは各ラインのCCビットがONになっていますので、ペアのパターンより優先順位が低いスプライト面に表示させる必要があります。

Q6・パターンを構成するスプライトをそれぞれ別のY座標に表示しようとしたら表示されないんだけど

スプライトモード2の仕様です(汗)。CCビットがONになっているスプライト面に表示されたスプライトは、CCビットがOFFかつ優先順位の高いスプライト面に表示されているスプライトとY座標が重なっている部分しか表示されません。

Q7・機能ビットを切り替えたい場合はどうすれば

偽SCREEN4モードは飽くまでSCREEN1であるため、COLOR SPRITE等のスプライトモード2操作命令が使えません。EC・IC・CC各機能ビットの状態はカラーデータに含まれているため、必要に応じてこれらを書き換える必要があります。DATA文の該当箇所を書き換えるか、VPoke命令でスプライトカラーテーブル上に直接書き込むかしなければなりません。どこを書き換えるかは後述するデータ構造や機能ビットの項を参考にしてください。

なお、実験したところ、モード1用のECビットは無効になっています。

表示にてきめんの不都合が出るゆえか、うちの兄もECビットはちょっとだけ操作できるようにしたようですが、他のビットの切り替えまでは手が回らなかったようです。本来はこれら进行操作できる関数も準備するべきだったんですが、兄に代わって謝るときです。

Q8・ところで機能ビットってなんですか

機能ビットとは、スプライトの各種表示設定を切り替えるスイッチのようなものです。詳しいことはあとで述べますが、スプライトモード2にはEC, CC, ICビットの三種類があり、割り込みや重ね合わせ等の設定ができるようになっています。機能ビットを理解するとスプライトモード2がより効果的に使えますので、どういうものか知っておくとよいでしょう。

Q9・「S・S・T」で3枚重ね以上のスプライトは作れますか

ごめんなさい！ 「S・S・T」は2枚重ねまでしか対応してません。

スプライトモード2では重ねる枚数が1枚増えるたび、1ラインごとに使える色数が倍々で増えるのですが、横8枚という水平方向表示制限を考えると、重ねられる枚数が多くても実用性は低いとかがえま...と言いつつときま(汗)。

Q10・32x32とか64x64ドットのデカキャラが作りたいんですが！

「S・S・T」のエディット領域は16x16ドットです。それ以上の大きさのキャラクターを作る場合、分割してパターンを作成するという方法を採用することになります。パターンテーブルの表示間隔や行数・列数は変えられないので、デザインしづらいことでしょう。これもごめんなさい！

Q11・偽SCREEN4でPCGを使う場合はどうすればいいですか

偽SCREEN4モードのPCG機能は、いわゆる「多色刷り」となります。GRAPHIC1モードに比してパターンジェネレーターテーブルとカラーテーブルの領域が3倍になるため、それぞれに対してパターンと色の定義をする必要があります。

なお、偽SCREEN4モードでは、PCGの使用に一部制約があります。上から8行目の範囲内でグラフィックキャラクタの「木」「金」「土」「日」を使うと色が化けるため、これらキャラクターだけは任意に書き換えることができません。VRAM上のパレットテーブルが上記キャラクターのカラーテーブルと重なってしまうことに由来する症状です。

Q12・マシン語でプログラムを作る場合、どこに置けばよいですか

「偽SC4」のマシン語サブルーチンは、メインメモリ上のD000H～DE0FH番地に配置してあります。その直後にはディスクワークエリアが控えています。ですのでマシン語でサブルーチンやプログラムを作る場合はCFFFH番地以前に配置してください。

Q13・16個以上のパターンを定義して使いたいのですが

「偽SC4」ではMSX2の最大数である64枚までスプライトを定義することは可能ですが、DATファイルに手を加えなければなりません。

「S・S・T」がDATファイルとして一度に出力できるスプライトは最大30枚(15パターン)です。それ以上の枚数を「偽SC4」で定義する場合、いくつかのDATファイルの一つにまとめてから、マージして実行することになります。

出力されたDATファイルは単純に連結すればいいというものではなく、パターンデータ部分を切り出してRENUMしてセーブして連結して、続いてカラーデータを切り出してRENUMしてセーブして連結して...という手間が必要です。まあ、面倒くせえ(おい)。

また、「偽SC4」1040行の「PT=15」の数値も、定義するパターン数に書き換える必要があります。PTが17以上になると1070行にIllegal function callが発生しますので、原因となる「PT*2-1」の部分を書き換えてください。

枚数が増えるほど、スプライトの定義には時間がかかります。一度定義したらデータを格納してあるVRAM上の領域をBSAVEのSオプションで保存しておき、以後はそれを読み出すという方法が、速くていいんじゃないかなとおもいます。

Q14・ついでに「PSYZANS」ってなんですか

「さいざんす」と読みます。創作物のクレジットを決めるにあたり、英語にした時の字面はカッコイイのに読みはそうでもない言葉がいいということで、荒井兄が見つけてきたのがこの単語でした。元ネタは荒井兄弟が敬愛するボードピリアン・トニー谷の決め文句です。くれぐれも「ムテキング」ではありません(おい)。

そのときどきでPSYXANSとかPSYZANSUとか微妙に綴りが変わってますが、ただの表記揺れです。ザインソフトを意識したわけではありません(おい)。

技術資料

VRAMマップ

「偽SC4」使用時のSCREEN1のVRAMのマップは以下のようになっています。

偽SCREEN4 (SCREEN1 GRAPHIC3)

| | |
|-------------|--------------------------------------|
| 0000H～07FFH | パターンジェネレータテーブル1 |
| 0800H～0FFFH | パターンジェネレータテーブル2 |
| 1000H～17FFH | パターンジェネレータテーブル3 |
| 1800H～18FFH | パターンネームテーブル1 |
| 1900H～19FFH | パターンネームテーブル2 |
| 1A00H～1AFFH | パターンネームテーブル3 |
| 1C00H～1DFFH | スプライトカラーテーブル |
| 1E00H～1E7FH | スプライトアトリビュートテーブル |
| 2000H～27FFH | カラーテーブル1 そのうち2020H～203FHはパレットテーブル |
| 2800H～2FFFH | カラーテーブル2 |
| 3000H～37FFH | カラーテーブル3 |
| 3800H～3FFFH | スプライトジェネレーターテーブル |
| 4000H～43FFH | スプライトカラーデータマスターテーブル |

DATファイルの構造

「S・S・T」で出力されるDATファイルのデータ構造は以下のとおりです。なお、データの内容はサンプルとして用意したものですので、当然作成されたデータによって異なります。

```
10000行 'PALETTE DATA
10010行   パレットデータ
10020行 'SPRITE PATTERN
10030行～ スプライトパターンデータ (パターン数x4行分)
10xx0行 'SPRITE COLOR
10yy0行～ スプライトカラーデータ (パターン数x2行分)
```

行番号はパターンの数によって変わります。

カラーパレットデータ

```
10010 DATA 000,000,237,765,111,131,621,267,760,222,251,753,026,500,444,777
```

コンマで区切られた3桁の数値が、各カラーのパレットを構成するR, G, Bのパラメーターです (上一桁目:R, 同二桁目:G, 同三桁目:B)。データの順序はカラーコードの順番に対応しています (1番目:0=透明色, 2番目:1=黒, …)。データ数は16個です。

スプライトパターンデータ

```
10030 DATA 00,07,0F,0F,08,0F,1F,3F,7F,FF,F0,97,78,10,0F,00
10040 DATA 00,C0,A0,A0,20,E0,F0,D8,DC,DE,1E,D2,3C,10,E0,00
10050 DATA 00,07,0F,0F,0F,0F,1F,3F,7F,FF,FF,FB,7F,1F,03,00
10060 DATA 00,80,C0,C0,C0,C0,C0,E0,E8,EC,EC,AC,C0,E0,80,00  ……
```

1パターンを構成する2枚のスプライトを16進数化したものです。スプライト1枚は各2行・32バイトで記述してあります。最初の2行がパターンの優先順位上位側のスプライトデータで、続く2行がパターン下位側のスプライトです。

1パターンで4行・合計64バイトを占めます。データは冒頭から4行ごとにパターン1、パターン2、パターン3…と順に並んでいます。全部でパターン数x4行となります。

スプライトカラーデータ

```
10240 DATA 0E,0E,0E,0E,0E,0E,0E,0E,0E,0E,0E,0E,0E,0E,0E
10250 DATA 49,49,49,49,49,49,49,49,49,49,49,49,49,49,49  ……
```

同じく1パターンを構成する2枚のスプライトの各ラインのカラーデータを16進数化したものです。スプライト1枚のカラーデータは各1行です。最初の1行がパターン上位側で、続く1行が下位側のカラーデータです。

スプライト1枚につき1行・16バイト、1パターンで2行32バイトを占めます。これも冒頭から2行ごとにパターン1、パターン2、パターン3…の順に並んでいます。全部でパターン数x2行です。

「偽SC4」でスプライトの定義をする場合、この部分を書き換えることで、各スプライトの色や機能ビットの初期値を変えられます

各行を構成する各1バイトが各スプライトの各ラインの設定に対応しています。各行冒頭より1ライン目、2ライン目、3ライン目…16ライン目となっています。

カラーデータにはEC, IC, CC各機能ビットの設定が含まれます。各バイトの下1桁が色で、上1桁が機能ビットの状態です。下位側のデータは重ね合わせのため、CCビットが0Nになっています。

機能ビットについて

これまで何度か触れてきましたが、スプライトモード2には機能ビットというものがあり、表示や割り込みといった機能の設定ができるようになっています。ここではその機能ビットについて少し説明しておきます。

スプライトモード2には3種の機能ビットが用意され、それぞれに異なる機能が与えられています。その内訳は以下のとおりです。

EC:シフト表示・スプライト表示を32ドット左にずらす。スプライトを画面左端から徐々に出したり消したりしたいときはここを操作する。

CC:優先順位・このビットを0Nにすると、スプライトが重なったところの色が、それぞれの色の論理和を取ったものとなる。実際には複雑な条件があるのだが説明はのちほど。

IC:衝突検出・このビットが0Nになっているラインはスプライトの衝突判定がなくなる。

SCREEN4以降のモードでは、COLOR SPRITE/COLORSPRITE\$命令で操作できます。各命令の使い方は各種MSX2マニュアル等をご参照ください。

偽SCREEN4モードではこれら命令が使えないため、VPOKE命令等で直接VRAMを書き換えて操作することになります。

| | MSB | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | LSB | スプライトカラーテーブル先頭アドレス |
|--|-----|----|----|----|---|-----------|---|---|---|-----|--------------------|
| 各機能ビットはスプライトカラーテーブル上に存在します。スプライトモード2ではスプライトの各ラインに対して色の指定が可能ですが、機能ビットの状態も各ラインに対して指定可能です(注)。カラーテーブルの構造は右記のとおりです。 | 0 | EC | CC | IC | 0 | ColorCode | | | | | スプライト#0 : 1ライン目 |
| | 1 | EC | CC | IC | 0 | ColorCode | | | | | スプライト#0 : 2ライン目 |
| | | | | | | | | | | | ⋮ |
| | 15 | EC | CC | IC | 0 | ColorCode | | | | | スプライト#0 : 15ライン目 |
| | 16 | EC | CC | IC | 0 | ColorCode | | | | | スプライト#0 : 16ライン目 |
| | | | | | | | | | | | ⋮ |
| | 511 | EC | CC | IC | 0 | ColorCode | | | | | スプライト#31:16ライン目 |

(注・厳密にはスプライト面に対する色や機能ビットの設定が、その面に表示されるスプライトに対して適用されます。)

図1・スプライトモード2のスプライトカラーテーブルの構造

スプライトカラーテーブルに格納されたデータの各バイトを2進数で表した場合、上1ビット目がEC, 同2ビット目がCC, 同3ビット目がICビットとなります。設定を変える場合はこれらビットを書き換えます。それぞれ1にするとON、0にするとOFFです。

ちなみに4ビット目は未使用で、下4ビットはカラーコードを表します。

ECビットとICビットの機能は上記したとおりです。ラインごとに設定可能なので、たとえば「スプライトの一部のみを左に32ドットずらす」といったことや「スプライトの下の部分だけ当たり判定を消す」といったこともできます。



4EH=01001110B

0 1 0 0 1 1 1 0

| | | | | | |
|----|----|----|---|-------|------|
| EC | CC | IC | 0 | Color | Code |
|----|----|----|---|-------|------|

ECビット:OFF カラーコード:1110B

CCビット:ON

=14

ICビット:OFF

図2・カラーデータと機能ビットの関係

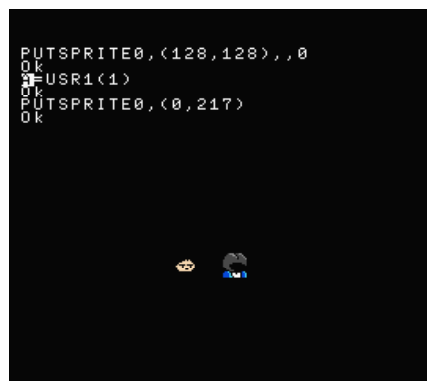
CCビットは少々立ち入った説明が必要です。このビットが0Nになったスプライトは優先順位を放棄し、通常とは異なる挙動をするようになります。

CCビットが0Nになったスプライトは単体では表示されなくなります。かわりに優先順位が高いスプライトが表示されている場合、元の表示面に最も近いものと優先順位が同じになり、再び表示されるようになります。つまりCCビットが0Nのスプライトを表示するには、優先順位が高い面にCCビットがOFFのスプライトも表示しなければなりません。表示されるのはCCビット0NのスプライトとY座標が重なる部分のみです。

CCビットOFFのスプライトとCCビット0Nのスプライトのドットが重なると、その部分の色が元のスプライトどうしの色の論理和 (OR) を取った色になります。この特性により、単純に重ね合わせたスプライトよりも多くの色が使えるというわけです。

1ラインに使える色数は、重ねる枚数が1枚増えるごとに2倍に増えます。そのうち1色は背景となる透明色ですのでその分を差し引いて、実際には2のn乗-1色 (n:重ねるスプライトの枚数) が使える、ということになります。2枚重ねなら1ライン最大3色、3枚重ねなら7色、4枚ならなんと15色が使える計算です。

また、CCビットが0Nになったスプライトは、衝突判定がなくなることも大きな特徴です。CCビットに関しては他にもいろいろな性質があるのですが、より詳しいことは各種専門書や雑誌記事等をご参考ください。



▲ECビットを弄って顔のところだけずらして表示。アイディア次第で様々な使い方が可能です。

プログラマからひとこと

この作品は時間に余裕があるときに少しずつ作ってやっとなさ完成した代物です。マニュアルを書くのも、ワープロなんて便利なものは持ってないので「こんなことなら日ペンのペン習字でもやっていれば良かったぜ」と思いながら書きました。

さて、俺が買ったMSXは、カシオのMX-10にサンヨーのPHC-23といった低価格マシンだったりします。現在のturbo Rは値段が高くて手が出ません。あの機能であの値段は安いけれど、グリコのお菓子のようにおまけばかりで、基本はどうでもという気がします。もし、次期MSXが出るならプログラミングできるゲームマシンといったようなシンプルな機械でもいいんじゃないかと思っていたりする次第です。

が、しかし、うだうだいっても始まらないわけで、とりあえず今のMSXを使いこなしたい人は、どうぞこの「S・S・T」を使ってみてください。

プログラマから
ひとこと

使ってみてください

この作品は時間に余裕があるときに少しずつ作ってやっとなさ完成した代物(しろもの)です。マニュアルを書くのも、ワープロなんて便利なものは持ってないので「こんなことなら日ペンのペン習字でもやっていれば良かったぜ」と思いながら書きました。さて、俺が買ったMSXは、カシオのMX-10に、サンヨーのPHC-23といった低価格マシンだったりします。現在のturbo Rは値段が高くて手が出ません。あの機能であの値段は安いけれど、グリコのお菓子のようにおまけばかりで、基本はどうでもという気がします。もし、次期MSXが出るならプログラミングできるゲームマシンといったようなシンプルな機械でもいいんじゃないかと思っていたりする次第です。が、しかし、うだうだいっても始まらないわけで、とりあえず今のMSXを使いこなしたい人は、どうぞこの「S・S・T」を使ってみてください。

PSYZANSU SOFT 山形・20歳

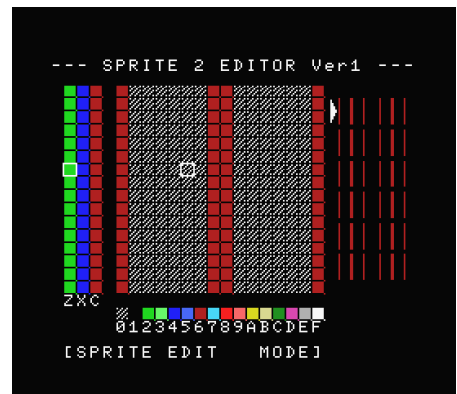


作者荒井兄・記～『MSX・FAN』1992年9月号から

後記

「S・S・T」には参考にしたツールがあります。アスキーのポケットバンクシリーズ「便利ツール、コキ使って!」収録の「スプライトモード2用エディター」です。そのツールは2枚同時エディット・各ライン色指定が可能でしたが、最低限の機能しかなく、今振り返れば決して使いやすいものではありませんでした。

自分こと本名荒井がそのツールで連日スプライトを作っていると、その様を見た荒井の兄がモード2に興味を示し、自分ならもっといいのが作れそうだと、少しずつ様々な機能の実験を繰り返し、実装していった結果できあがったのがこの「S・S・T」です。製作はほとんど荒井兄一人。荒井は動作テストを兼ねてサンプルデータを作成したり、外観について意見した程度です(ついでにMファン掲載時のトニー谷のイラストは荒井が描きました)。



▲「便利ツール、コキ使って!」収録
「スプライトモード2用エディター」

当初は使いやすいツールを作ろうということで製作していましたが、できあがってみるとこのほか出来がよかったので、徳間書店のMSX専門誌「MSX・FAN」に投稿したところ見事採用が決まり、1992年9月号に掲載とあいなりました。当時のMファンには付録として、掲載作品等を収録したディスクが毎号付いてきたのですが、該当号のディスクには「S・S・T」も収められています。

当ツールはエディターとデータローダーのふたつより構成されます。しかし収録されたのはエディターだけでした。容量不足か、データローダーは次号にて収録予定と告知されていたのですが、その後収録されることはなく、結局配布されないまま休刊を迎えたのであります。

このツールの真骨頂は、SCREEN1でスプライトモード2を扱えることにあります。件のモード2エディターの他にも、「SEPT」といった非常に優れたツールは存在するのですが、どれもSCREEN4以降で、SCREEN1での表示を可能とするものはありません。「SCREEN4よりSCREEN1でやる方が作りやすかったからSCREEN1を採用した」と兄は語っていますが、SCREEN4より格段に扱いやすいSCREEN1にスプライトモード2を実装できれば、MSX2での創作の可能性が大きく広がるだろうという考えもあったようにおもいます。それだけにデータローダーがお蔵入りしたことを、弟ながら残念におもっていたのであります。

それから時が経ち、一次市場でのMSXは滅亡を迎えてしまいました。しかし熱心な愛好家たちが、21世紀になった今なお新作ゲームやツールを発表しているのを見て、年々荒井も何かしなければという念が強くなり、兄の許可を得た上で「S・S・T」を拙「何とか庵」にて配布することにしたしました。26年の時を経て、ようやく完全な形で公開することができたというわけです。

後年、ゲーム製作者M-KAIさんが代表作「完全攻略キョクゲン」の開発で、この「S・S・T」を使っていたらしいという話を知りました。兄に伝えたところ、へえ、そうだったのかと意外そうな反応をしながらも、まんざらでもない様子でありました。

というわけで、この「S・S・T」がみなさんの創作の一助となり、ひいてはレトロコンピューティングの創作文化に少しでも寄与できれば、製作にほんの少し関わった人間として、これよりうれしいことはありません。

そう大げさでなくとも、もっとMSXを使いこなしてみたい方、とりあえずご利用ください！

2018年6月 「メッコール」改 本名荒井

参考文献

『MSX-Datapack』 1991年 アスキー
『MSX2テクニカルハンドブック』 1986年 アスキー出版局
『あるとないとじゃ大違い 便利ツール、コキ使って!』 1987年 MSXマガジン編集部
『MSX・FAN』 1992年1月号～4月号 1992年 徳間書店インターメディア
『MSX・FAN』 1992年9月号 1992年 徳間書店 インターメディア
各種MSXマニュアル

MSX2用モード2スプライト支援ツール『S・S・T』使用説明書
第1版 2018年6月
第2版 2022年1月
第3版 2022年6月
本名荒井 著
(C) 1992, 2018 PSYZANS (荒井兄と本名荒井)

メモという名の穴埋めスペース

PSYXANS