プチコンmkII（マークツー）

取扱説明書

2 はじめに

プチコンmkIIについて

本製品は内蔵されているBASICでプログラムを作成し、実行することができる愉快なBASIC言語学習ツールです。通信機能を使うことで友達にプログラムを渡したり、友達の作った画像をもらうこともできるので、みんなで協力して作ることもできます。

利用規約

- 本製品により作成されたプログラムや画像等のリソースは、送信命令によって多くの人が見たり実行する可能性があります。他人が不快な気持ちになるようなもの、プライベートな情報、他人の権利（肖像権やプライバシー権、著作権など）を侵害するものなどは送信しないでください。
- 公衆に著しく迷惑をかける不良行為を行ったり、わいせつ、名誉棄損にあたる画像を公表した場合、法令や条例によって罰せされる恐れがあります。
- スマイルブームはお客様が送信した情報および送信したプログラム等によって生じたトラブルについては一切責任を持ちません。

取扱上の注意点

- 本製品に搭載されているBASICは、これまでに発売されているBASICとは、互換性がありません。古い書籍等からの移植の際には、文法の違いにご注意下さい。
- 内部で扱われている数値は、固定小数型を利用しているため計算による誤差が大きく、正確な精度が必要なプログラムには利用できません。
- 本製品では行番号を使った分岐は利用できません。@で始まるラベルを定義した上で分岐命令を利用します。
- 複雑な計算を行ったり、表示する要素を増やすようなプログラムを書くと、実行速度が低下する可能性があります。
- SAVE・RECVFILE・DELETE等のファイルへの書き込みが行われる命令を繰り返し行うと、読み書きに時間がかかるようになることがあります。
- 予約されている命令は、小文字で入力しても自動的に大文字に変換されます。
- 本製品の条件比較は、==で等しい、!=で異なる、となります。(C言語風)
- 本製品のFOR命令は、先に条件を判断します。FOR I=0 TO -1 のようにSTEP1では成り立たない状態の場合、FOR内部を実行せずにスキップしてNEXT以降の命令を実行します。既存BASICのように必ず1回実行されません。
- 本製品に内蔵されているプログラムテキストを編集する機能は、1行100文字、最大9999行まで登録できる仕様ですが、テキスト用に確保されたメモリーの上限に達した時点で入力できなくなります。約52万文字まで入力可能です。
- 画面への描画を行う命令を使った場合、正しく画面に表示されないことがあります。これは、直前に実行していた他のプログラムで、上下画面のページを切り替える命令が下画面を指定していた等が原因で起こります。このような状況になった場合、実行モードに切り替えて、

  ACLS (Enter)

  を実行した上で描画命令を実行してください。これでも出ない場合は、色がすべて見えない値に変更されているか、キャラがすべて透明になっている可能性があります。COLINIT命令やCHRINIT命令を参考に元の状態に戻してください。
- 割り算を含む計算を行い、必要な数値が整数部分だけの場合は、FLOOR 命令を使って整数部分だけを利用して下さい。座標計算などで誤差が積み重なると微妙なズレが生じてしまいます。
- INKEY$() によるキーボードからの情報取得において、TABキーやBSキー等の一部のキーの情報はシステムの都合上取得することができません。全てのキーボード上のボタンの情報を取得したい場合は、システム変数のKEYBOARDを利用してください。
- 文字列変数にラベル名を代入してラベル代わりに利用する方法はラベルが使える一部の命令のみ対応しています。
- BGMと効果音と音声合成による発音は最大16個の音までしか同時に発音することができません。

操作方法

本製品は開発ツールのため、ゲームとは操作方法が異なります。

十字ボタン	テキスト編集モード時にカーソルを移動します。(Lボタンを押しながら上下でページ単位の移動、左右で行頭と行末への移動)
A,B,Y,X	ユーザーがプログラム実行中に利用可能。テキスト編集モード時Aボタンは、Enterキー　Yボタンは、←キーとして機能します。
Lボタン,Rボタン	キーボードにあるSHIFTボタンと同じ扱いとし、キートップ上に紫色で書かれている文字を入力するときに利用します。プログラム実行中はユーザーが利用できます。
STARTボタン	プログラムが登録されている場合、スタートボタンを押すとユーザープログラムが実行されます。プログラム実行中はユーザーが利用できます。
SELECTボタン	ユーザープログラムの実行中、セレクトボタンはESCキーとして機能します。

実行モード時に、十字ボタン左とRボタンとスタートボタンを同時に押すと、画面表示を初期状態に戻すことができます。プログラム実行後に画面が見にくくなった場合に利用すると便利です。

プログラムとは？

コンピュータは人間からの指示に従って画面への表示や、音を鳴らしたり、タッチスクリーンの状態を調べたりすることができる素晴らしい機械です。

プログラムとは、コンピュータに対して指示を出すための命令の集まりです。本製品に搭載されている「BASIC」と言う名前のコンピュータに指示を出すための言語は、人間が使う言葉に近い文法でコンピュータへ指示を出すことができます。

例えば・・

画面に文字を表示しろ！…PRINT命令
音を鳴らせ！…BEEP命令
画面に円を描け！…GCIRCLE命令
キャラクタを表示しろ！…SPSET命令
数字を記憶しておけ！…変数と代入命令
数値の大小を調べろ！…比較と分岐命令
保存しておけ！…SAVE命令

このようにBASICには、コンピュータに指示させたい要素に対応した命令がたくさん用意されています。全ての命令を覚えるのは大変ですが、自分が指示したい命令から少しずつ覚えていけば、ゲームやツールなどを作ることができます。

最初は難しいかもしれませんが、本製品だけあれば、プログラムを作って動かすところまでが実現できます。もちろん間違った命令を書けば、バグと呼ばれる誤動作も生み出せます。

現在、ゲームを作っている会社の人達も小さい頃にBASICを使ってプログラムを覚えた人がたくさんいます。未来のゲーム開発者を目指して、少しずつプログラムの楽しさを味わってみてください。

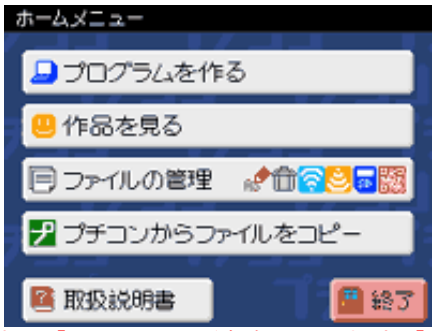
用語集

説明書内で良く使われるプチコン独自の用語や言い回しについての解説。

単語	説明
キャラ	BGやSPRITEで利用する8x8ドット単位16色の画像
カラー	BG・SPRITE・グラフィック用に256色ずつ (0番は透明)
パレット	画像や文字に割り当てる色単位。 1パレット＝16色
BG(ビージー)	キャラをしきつめてマップなどの画像を表現する表示機能
SPRITE(スプライト)	最大100個のアニメ付キャラを表示するための機能
システムアイコン	モードやツールの切替を行うアイコンボタン
ドット	画面上の絵や文字を構成する色のついた点
フレーム	表示同期を取るための時間の単位 1フレーム＝1/60(秒)
VSYNC(ブイシンク)	表示画面を書き換える周期 最低単位は1フレーム
ファイル	作成したプログラムや画像などを保存するメモリー領域
リソース	BASIC内で利用するために、ファイルから読み込んだプログラムや画像などの総称。内部的に確保されたメモリー内またはビデオメモリー内に保持される
トラック	BGMを演奏する場所 最大8曲(8トラック)同時再生可能
チャンネル	MMLを演奏する単位 最大8種類の楽器を同時に演奏可能
MML	Music Macro Language 作曲するためのコマンド文字列
始点x 始点y	矩形領域の左上座標 (x=横、y=縦)
終点x 終点y	矩形領域の右下の座標 (x=横、y=縦)
転送先x 転送先y	転送先矩形領域の左上の座標 (x=横、y=縦)

3 ホームメニュー

ホームメニュー



DSi本体に「プチコン」が存在しない場合「プチコンからファイルをコピー」ボタンは表示されません。

ホームメニューについて

プチコンmkIIのさまざまな機能は、このメニューから分岐し、作業が終了するとホームメニューに戻ります。ホームメニューには以下の5つの項目が用意されています。

SmileBASICが起動します。プログラムを作る場合はこのボタンを押します。

保存されている作品をリストから選択して実行します。

保存されているファイルの名前変更、削除、コピー、送受信等を行います。

古いプチコンがある場合、ファイル内容をプチコンmkII側にコピーします。

プチコンmkIIの基本的な使い方やBASICの命令一覧を表示します。

プチコンmkIIを終了してDSiメニューに戻ります。

DSi本体に「プチコン」が存在しない場合「プチコンからファイルをコピー」ボタンは表示されません。

プログラムを作る

SmileBASICを使ったプログラムの作り方については「プログラムを作る(詳細)」以降のページを参照してください。

作品を見る

サンプルやプログラムを選択して実行する機能です。実行ボタンを押すとSmileBASICが起動してプログラムが実行されます。

・プログラムの終わりに到達した
・プログラム内でエラーが発生した
・ESCキーや中止ボタンが押された
・セレクトボタンが押された
・END命令が実行された

上記いずれかの条件が満たされた場合、実行中のプログラムが停止し、SmileBASICを終了して「ホームメニュー」に戻ります。
なお、終了時の変数の状態は全て失われます。次回も利用したい変数の内容等は、各プログラムでSAVE命令等を利用して値を記録して下さい。また、このモードで実行した場合は編集ボタンでリスト内容を見ることができません。

ファイル管理

ファイル管理については、「ファイルの管理(詳細)」のページを参照してください。

プチコンからファイルをコピー

すでに「プチコン」をご購入いただいたお客様が「プチコン」で作られた作品を「プチコンmkII」のファイルとして再利用される場合に利用する機能。
DSi本体に「プチコン」が存在しない場合「プチコンからファイルをコピー」ボタンは表示されません。

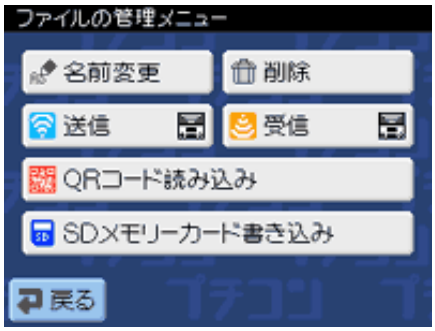
「プチコン」側のすべてのファイルをまとめてコピーします。「プチコンmkII」側に同一名称のファイルが存在している場合、「上書き」「名前変更」「スキップ」のいずれかを選択してください。

「プチコン」側のファイルから選択してコピーします。「プチコンmkII」側に同一名称のファイルが存在している場合、「上書き」「名前変更」「スキップ」のいずれかを選択してください。

「プチコン」からコピーしたプログラム（命令）は、「プチコンmkII」で書式の省略形判定が厳密になったためエラーが出る可能性があります。エラーが出た場合は、命令の説明を確認して不足している引数等を追加してください。

4 ファイルの管理(詳細)

ファイルの管理(詳細)



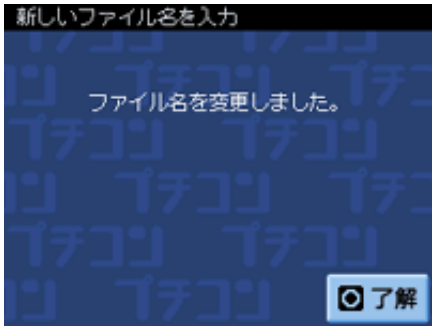
名前変更

記録されているファイルの名前を変更するための機能です。以下の操作でファイル名を変更することができます。

1) 名前を変更したいファイルを選択

2) 簡易キーボードで名前を変更

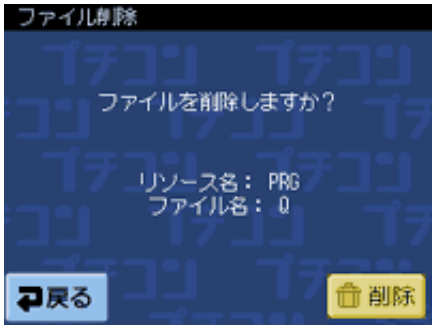
3) 変更終了



削除（ファイルを消す）

記録されているファイルを削除するための機能です。以下の操作でファイルを削除することができます。

1) 削除したいファイルを選択
2) 削除の確認



3) 削除終了

送信（ファイルを渡す）

記録されているファイルを他の人へ送信するための機能です。以下の操作でファイルを送信することができます。

1) 送信したいファイルを選択
2) 送信用のダイアログが開く

※以降の動作はSENDFILE命令と同様

受信（ファイルをもらう）

他の人からファイルを受け取るための機能です。以下の操作でファイルを受信することができます。

1) 受け取るファイル名を入力

2) 受信用のダイアログが開く

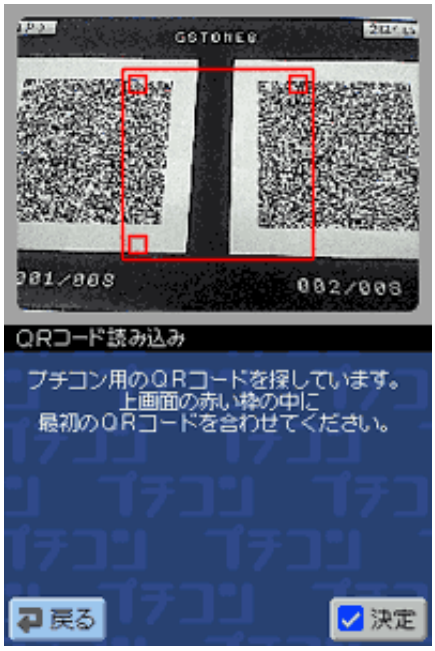
※以降の動作はRECVFILE命令と同様

QRコード読み込み

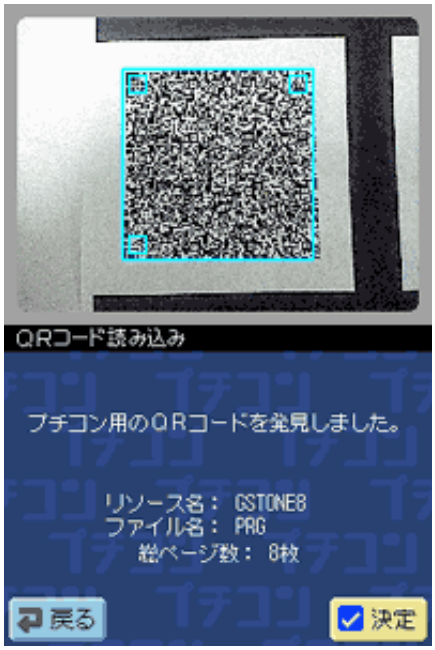
QRコードの読み込み中は、読み込み処理に集中するため背景のスクロールが停止し、ボタンが押しにくい状態となります。

プチコンmkII専用のQRコードを本体の外側カメラを使って読み込む機能です。以下の操作でQRコードを読み込んでファイルを作ることができます。

1) 最初のQRコードを赤枠内に合わせます



2) 発見するとページ数が表示されます



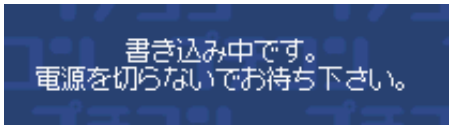
3) 必要なページ数分の読み込みます

4) 読み込み終わるとファイルを生成します

SDメモリーカード書き込み

記録されているファイルをSDメモリーカードへ書き込むための機能です。以下の操作でファイルを書き込むことができます。

1) 書き込みたいファイルを選択
2) 書き込み開始



3) 書き込み終了

\private\ds\app\4B4E414A\ファイル名

正常に書き込みが終了すると、SDメモリーカードに上記フォルダが作成され、管理ファイルが出力されます。

＜書き込みに関するご注意＞
・十分な空きが無い
・書き込み禁止スイッチが「オン」の状態
これらの状態ではSDメモリーカードには書き込めません。十分に空きがあり書き込み禁止スイッチが「オフ」の状態になっているSDメモリーカードをご用意ください。

SDメモリーカード上に出力されるファイルの仕様は一般公開されておりません。お客様がファイルの内容を直接確認することはできません。ファイル内容の解析や改変等は行わないでください。

QRコード生成ツール

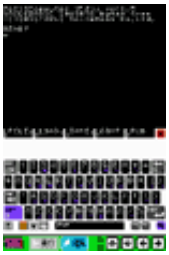
お客様がSDメモリーカードに書き込まれたファイルを取り出すことはできませんが、無料WEBアプリをWEBブラウザから利用することでファイルの内容を確認したりQRコードを生成することができます。詳しくは、弊社ホームページをご覧ください。

http://www.smileboom.com/

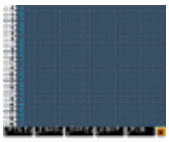
5 プログラムを作る(詳細)

プログラムを作る(詳細)

ホームメニューから「プログラムを作る」ボタンをタッチすると、SmileBASICが起動します。
SmileBASICは以下の3つのモードで構成されています。



●実行モード
キーボードから直接実行可能な命令を入力できます(編集モードから実行ボタンで切替)



●編集モード
プログラムのソースコードを書くためのテキスト編集機能です(実行モードから編集ボタンで切替)

●取扱説明書
プチコンの基本的な使い方やBASICの命令一覧を表示します(説明書ボタンで切替)

モード切替

本製品のモード切替はシステムアイコンで行います。

◆システムアイコン (実行モード時)

1	取扱説明書アイコン 本製品の使い方を表示します。
2	実行モードアイコン 実行モードへの切り替え。
3	編集モードアイコン テキスト編集モードへの切り替え。
4	ユーザー用システムアイコンエリア プログラム内から利用可能です。

◆システムアイコン (編集モード時)

5	コピー カーソル位置の1行を内部に取り込みます。
6	ペースト カーソル位置に内部にコピーされている1行を貼り付けます。

実行モード

「実行モード」は、作ったプログラムを実行するためのモードです。「編集モード」で書いたプログラムは、「実行モード」で、キーボードから"RUN"と入力し、ENTERキーを押すことによって実行することができます。また、作ったプログラムを保存したり、読み込む場合も「実行モード」を利用します。

◆上画面：コンソール

キーボードから入力した命令を表示します。

◆下画面：キーボード表示

文字切替ボタンを押すことでキーボード上の文字が変化します。SHIFTキーを押すと、入力できる文字が変化します。

「実行モード」時は、PNLTYPEによるキーボードの切り替えには対応していません。

文字切替ボタン	
A	英字
♥	記号
ア	カナ

SAVE命令によって保管されたプログラムファイルを消したり、ファイル名の変更等も「実行モード」から行います。詳しくは「17 実行モード専用命令」をご覧ください。

編集モード

「編集モード」は、プログラムを書くためのモードです。様々な命令を使って自由度の高いプログラムを作ることができます。どのような命令があるかを確認する場合は、この説明書を開いて命令と関数に関する説明ページをご利用ください。

◆上画面：ソース表示

キーボードから入力した文字列を表示します。改行で自動的に番号が増加します。1行100文字まで画面に表示し、表示しきれない範囲は横方向へのスクロールします。

◆下画面：キーボード表示

文字切替ボタンを押すとキーボード上の文字が変化します。

命令入力支援機能

プログラムを入力する際に、どのような命令があるか忘れてしまった時に、英文字を入力することでBASICの命令及びシステム変数から、候補を探してくれる機能が搭載されています。

1) 候補選択エリア

下画面のファンクションキーの下にある白い空白部分が候補選択エリアです。

2) 最初の候補表示

キーボードから、G のキーを押すと候補選択エリア内に、Gで始まる命令やシステム変数名が並びます。

3) 候補絞り込み

さらに C キーを押します。候補が変化してGCで始まる命令が候補エリアに並びます。

4) 候補確定

対象となる命令が見つかったら候補をタッチしてください。GCIRCLE のように長い命令の入力手間を省くことができます。

>> << 候補エリアに収まらない場合このボタンで候補ページを切り替えることができます。

検索機能 (編集モード専用)

編集モードでキーボード右下の(虫眼鏡アイコン)を押すとファンクションキーの下に検索文字列入力行が表示されます。プログラムが長くなってしまった場合に、リスト内のラベルや変数名を探す時に利用します。

探したい文字列を入力した後に、Enterキーを押すと検索が始まります。十字ボタンの上下で、現在位置から調べる方向を指定して検索することもできます。虫眼鏡アイコンを押すと検索モードから抜けます。

簡単なプログラムの書き方

コンソール画面に文字列を表示して音を鳴らすプログラムを書いてみましょう。プチコン起動後、サンプル等を実行していない状態で、以下の手順を実行してください。（最初はプログラムが無い状態で実行モードになっています）

1) 編集ボタンを押して編集モードにする
2) 1行目に PRINT "MOJIDESU" と入力
3) 2行目に BEEP と入力

```
0001 PRINT "MOJIDESU"
0002 BEEP
```

4) 実行ボタンを押して実行モードにする
5) RUN (Enter) と入力

```
MOJIDESU
OK
```

コンソールに文字が表示されてピッと音が出ます。また、実行モードで直接命令を実行することも可能です。

```
GLINE 0,0,255,191,15 （Enter）
```

これで、上画面に左上から右下へ斜めの線が描画されます。

```
GCLS (Enter)
```

と入力すると文字は消えずに線だけを消すことができます。どのような動きをするのか、ちょっと試してみたい時には、実行モードで命令を入力して試してみましょう。

6 サンプルプログラム

サンプルプログラム

本製品には以下のサンプルプログラムが含まれています。詳細についてはプログラムの内容をご覧ください。

サンプルファイルの使い方

1) 実行モードにする
2) EXEC"SAMPLE2" (Enterを押す)
3) 停止する場合はセレクトボタンを押す
4) 編集モードでプログラムを確認

EXECの後の""で囲まれた SAMPLE2 の部分を変更することで他のサンプルも確認することができます。なお、含まれている全てのサンプルプログラムについては、SAVE命令を使って別の名前で保存することでスマイルブームに確認することなく自由に改造することが可能です。

基本サンプル

BASICの基本的な命令を利用した学習用のサンプルプログラムです。編集モードでプログラム内容を確認して、BASICでプログラムを作る参考にしてください。

ファイル名	内容
SAMPLE1	コンソールへの文字表示
SAMPLE2	文字入力を使った簡易計算機
SAMPLE3	キーボードを使った簡易鍵盤と十字ボタンによるドラムマシン
SAMPLE4	数当てゲーム
SAMPLE5	バイオリズム
SAMPLE6	下画面を使ったシーケンサー
SAMPLE7	複数の敵を動かして下画面のタッチで弾を発射するデモ
SAMPLE8	変数を操作する便利な命令の使い方
SAMPLE9	スプライトとBGを使ったデモ
SAMPLE10	複数ページを利用したグラフィック画面のデモ
SAMPLE11	MML(MusicMacroLanguage)で作曲と演奏
SAMPLE12	自分だけの楽器音を作る
SAMPLE13	おしゃべりコンピュータ

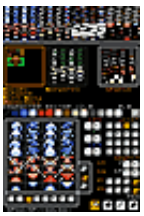
高度なサンプル

様々なBASIC命令を駆使した高度なサンプルです。各種ツールサンプルは、ゲームを作るときに必要なデータを作る道具としても利用できます。これらのサンプルも編集モードでプログラムを確認することができます。BASICプログラムが理解できるようになったら、自分だけの機能を追加することも可能です。

ファイル名	内容
CHRED	SPRITEやBG用のキャラ画像や色を作るドットお絵かきツール
SCRED	BGスクリーンにBGキャラを並べて背景を作るツール
GRPED	256色グラフィックを使ったお絵かきツール
DRWED	線や四角などの操作単位を記録するタイプのお絵かきツール
GAME1	敵の攻撃をよけながら画面上のドットをすべて消すレースゲーム
GAME2	敵を倒しながら3D風ダンジョンを進むロールプレイングゲーム
GAME3	宇宙空間で戦う縦スクロールシューティングゲーム
GAME4	弾幕を避けながら敵を倒すシューティングゲーム
GAME5	巨大なキャラクタが戦う格闘ゲーム
GAME6	質問に答えるといい加減なことを言う占いゲーム

CHRED(キャラ作成)

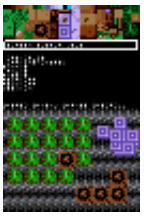
EXEC "CHRED" (Enter) で実行。



BGやSPRITE用のキャラデータを作成するツールです。色見本やツールを選択して、EDITエリアに書き込みます。Aボタンを押してファイルモードに入り、L(またはS)を入力してENTERを押すと読み込み(書き込み)を行います。保存したデータはプログラムから利用可能です。
(例) LOAD "BGU0:MYBG00"

SCRED(BGスクリーン作成)

EXEC "SCRED" (Enter) で実行。



背景用BGキャラを並べて町並み等のBGスクリーンデータを作成するツールです。キャラ見本からキャラを選んでEDITエリアへ貼り付けます。保存したデータはプログラムから利用可能です。

(例) LOAD "SCU0:MYSC00"

GRPED(グラフィック作成)

EXEC "GRPED" (Enter) で実行。

256色まで利用可能なグラフィック画面用の画像データを作成するツールです。色見本から色を選んでEDITエリアで絵を描きます。保存したデータはプログラムから利用可能です。

(例) LOAD "GRP0:MYGRP00"

DRWED(ドロー系お絵かき)

EXEC "DRWED" (Enter) で実行。

線を引いたり、四角を描くと言った操作手順を8000回分記憶するお絵かきツール。ドットを描くツールとは異なり、記録を再生しなおすことで何度でもやり直して絵を描くことが可能です。操作情報は、グラフィック画面上に色として記録されます。

保存された色情報から絵を再生する方法は、ツールのプログラムを編集モードで確認してください。再生用の処理部分を取り出せば、自分のプログラムにも利用できます。

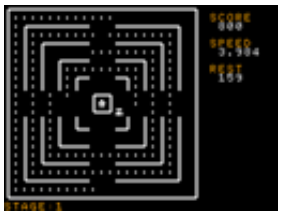
7 サンプルゲーム

サンプルゲーム

プチコンを使って作られたゲームのサンプルです。編集モードでプログラムを見ることができます。敵を弱くしたり、弾のスピードを変えるなどプログラムを改造して楽しみましょう。

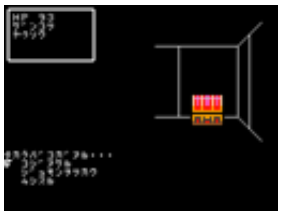
GAME1(レース)

EXEC "GAME1" (Enter) で実行。
十字ボタンで移動、画面上にあるドットを全て通過するレーシングゲーム。ドットを通過するとスピードアップ。



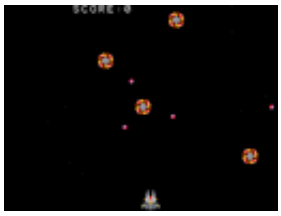
GAME2(3D風ダンジョン)

EXEC "GAME2" (Enter) で実行。
十字ボタンで移動、迷路を進みモンスターを倒しながら薬草や武器を手に入れて、最上階を目指す。
ロールプレイングゲーム。



GAME3(縦シューティング)

EXEC "GAME3" (Enter) で実行。
十字ボタンで移動、Aボタンで発射。
迫りくる敵を倒しながらボスを倒せ！



GAME4(弾幕シューティング)

EXEC "GAME4" (Enter) で実行。
高度なSPRITE命令を駆使した画面を埋め尽くす敵と弾を避けながら倒して巨大なボスを倒せ！背景は3重スクロール。

GAME5(対戦格闘)

EXEC "GAME5" (Enter) で実行。
画面を覆うほどの巨大なキャラクタ同士が剣を使って戦う格闘ゲーム。

GAME6(おしゃべり占い)

EXEC "GAME6" (Enter) で実行。
おしゃべりコンピュータの質問に答えると謎のラッキーアイテムを教えてくれる不思議なゲーム？

8 キャラ表(SPRITE)

SPU0・1(0～127)

SPU0：ゲーム用汎用部品

SPU1：男の子と魔法使いの娘

SPU2・3(128〜255)

SPU2：ザコ敵など

SPU3：エフェクトやアイテムなど

SPU4・5(256〜383)

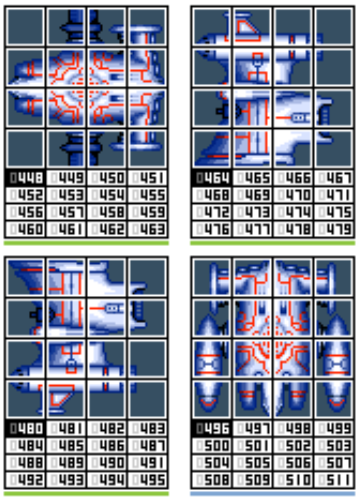
SPU4：飛行物体

SPU5：大きめの敵、爆発など

SPU6・7(384～511)

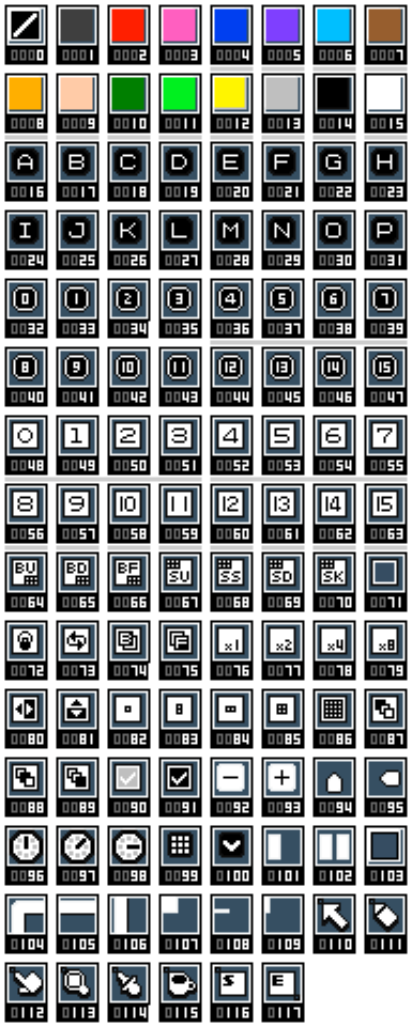
SPU6：横型の大きめの敵

SPU7：巨大戦艦



SPS

下画面専用の汎用キャラクタ



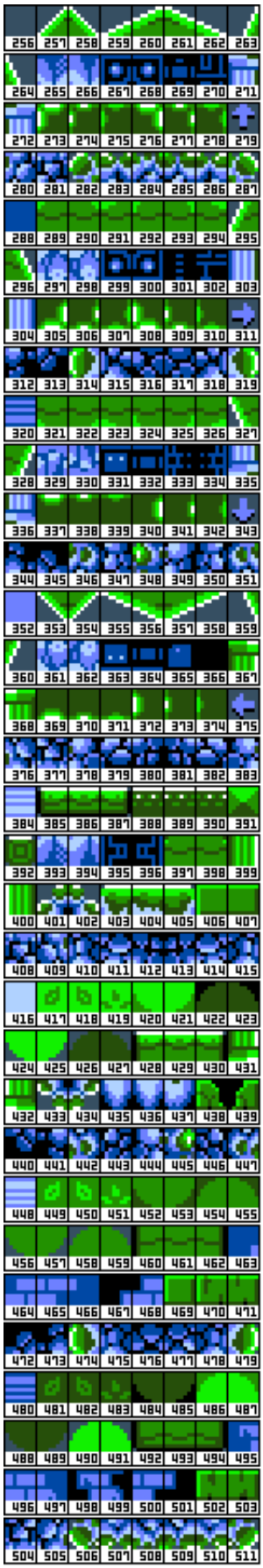
SPD

ツールなどに利用するための汎用アイコン

9 キャラ表(BG)

BGU0(0～255)

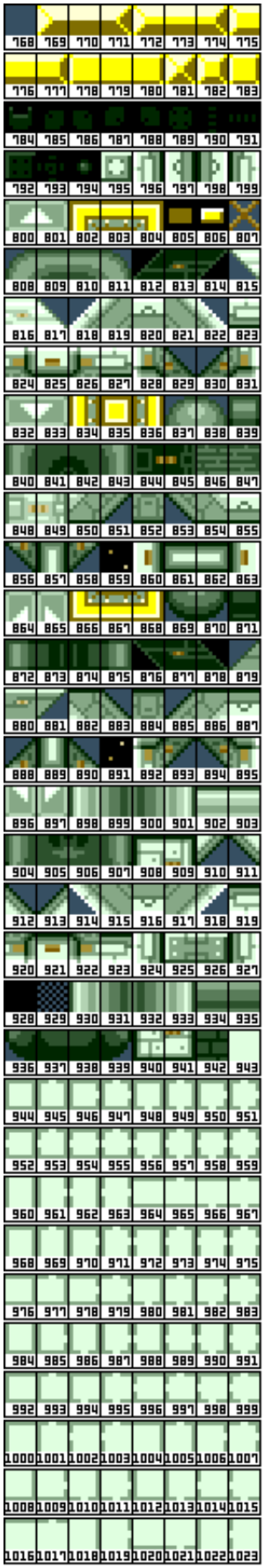
BGU1(256～511)


256 257 258 259 260 261 262 263
264 265 266 267 268 269 270 271
272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287
288 289 290 291 292 293 294 295
296 297 298 299 300 301 302 303
304 305 306 307 308 309 310 311
312 313 314 315 316 317 318 319
320 321 322 323 324 325 326 327
328 329 330 331 332 333 334 335
336 337 338 339 340 341 342 343
344 345 346 347 348 349 350 351
352 353 354 355 356 357 358 359
360 361 362 363 364 365 366 367
368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383
384 385 386 387 388 389 390 391
392 393 394 395 396 397 398 399
400 401 402 403 404 405 406 407
408 409 410 411 412 413 414 415
416 417 418 419 420 421 422 423
424 425 426 427 428 429 430 431
432 433 434 435 436 437 438 439
440 441 442 443 444 445 446 447
448 449 450 451 452 453 454 455
456 457 458 459 460 461 462 463
464 465 466 467 468 469 470 471
472 473 474 475 476 477 478 479
480 481 482 483 484 485 486 487
488 489 490 491 492 493 494 495
496 497 498 499 500 501 502 503
504 505 506 507 508 509 510 511

BGU2(512～767)


512 513 514 515 516 617 518 519
520 521 522 523 524 525 526 527
528 529 530 531 532 533 534 535
536 537 538 539 540 541 542 543
544 545 546 547 548 549 550 551
552 553 554 555 556 557 558 559
560 561 562 563 564 565 566 567
568 569 570 571 572 573 574 575
576 577 578 579 580 581 582 583
584 585 586 587 588 589 590 591
592 593 594 595 596 597 598 599
600 601 602 603 604 605 606 607
608 609 610 611 612 613 614 615
616 617 618 619 620 621 622 623
624 625 626 627 628 629 630 631
632 633 634 635 636 637 638 639
640 641 642 643 644 645 646 647
648 649 650 651 652 653 654 655
656 657 658 659 660 661 662 663
664 665 666 667 668 669 670 671
672 673 674 675 676 677 678 679
680 681 682 683 684 685 686 687
688 689 690 691 692 693 694 695
696 697 698 699 700 701 702 703
704 705 706 707 708 709 710 711
712 713 714 715 716 717 718 719
720 721 722 723 724 725 726 727
728 729 730 731 732 733 734 735
736 737 738 739 740 741 742 743
744 745 746 747 748 749 650 751
752 753 754 755 756 757 758 759
760 761 762 763 764 765 766 767

BGU3(768～1023)



10 サウンド関係の表

サウンド関係の表

BGM、BEEP、TALKに関するプリセットやコマンドなどを表にまとめたもの。

プリセットBGM

番号	説明
0	軽快な曲
1	湿った暗い感じの曲
2	緊張感高まる曲
3	激しくアップテンポな曲
4	スタートジングル
5	クリアジングル
6	ゲームオーバー
7	メニューセレクト
8	結果発表
9	スタッフロール
10	スタッフロール　その2
11	時代劇ゲーム風
12	軽快なマーチバンド風
13	激しいロック調
14	軽快な曲　その2
15	WOND
16	考え中
17	WOND2
18	未来系
19	BAL
20	BAL_2
21	スパイ系
22	SCI
23	シューティング
24	パッド
25	SEN
26	ピュア
27	ROA
28	CUR
29	FIG

プリセット効果音

番号	説明
0	BEEP
1	ノイズ
2	カーソル移動
3	決定
4	キャンセル
5	上昇
6	下降
7	コイン
8	ジャンプ
9	着地
10	発射
11	ダメージ
12	金属
13	爆発
14	叫び声
15	ブレーキ
16	バンジョー
17	シンセストリングス
18	シンセブラス
19	シンセベース
20	ギター
21	オルガン
22	ピアノ
23	カウベル
24	タム
25	シンバル
26	オープンハイハット
27	クローズハイハット
28	ハンドクラップ
29	リムショット
30	スネアドラム
31	バスドラム
32	OK2
33	BALL
34	和風
35	VOLT
36	AUTO
37	キラ
38	ESC
39	バンジョー2
40	スクラッチ

```
41 ギター2
42 オルガン2
43 ピアノ2
44 PASS
45 UP2
46 録音
47 シンセタム
48 カウベル2
49 metro
50 tri
51 コンガ
52 ダンスBD
53 ダンスSD
54 ダンスHH
55 ヒット
56 ティンバレス
57 チャイニーズシンバル
58 ミニシンバル
59 シェーカー
60 鈴
61 太鼓
62 シンセ
63 かっこう
64 パフ！
65 nohkan
66 humandr1
67 humandr2
68 犬
69 猫
```

MMLコマンド

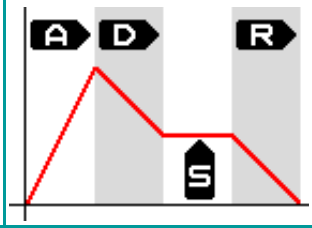
Music Macro Language を略してMMLと呼びます。曲を演奏するための音程や長さを記述するコマンドと音量や音色を切り替えるコマンド等を文字列で表現します。

曲全体用のコマンド	
: 数値	チャンネル指定 (:0～:7) 1曲で8チャンネル同時再生。
T 数値	テンポ (T1～T240)
$変数番号=値	変数への代入 指定された変数番号へ値を代入。 変数番号(0～7)、値(0～255) プログラムからは、BGMSETVで書き込み、BGMGETV()で読み込めます。●のついているコマンドの数値として利用可能です。

音の長さに関するコマンド	
L 数値	デフォルト音長 (L1～L192) 付点は、L4. のように直後に .を付ける。
Q 数値	ゲートタイム (Q0～Q8) 発音時間に対して実際に音が鳴っている時間の割合。
&	タイ（スラー） 2音をつなげた長さで演奏。

音階や音の高さに関するコマンド	
C D E F G A B C# D# E# F# G# A# B# (#は+でも可) C- D- E- F- G- A- B-	音階指定 ● C=ド D=レ E=ミ F=ファ G=ソ A=ラ B=シ。半音は、# か - を付ける。 デフォルト音長が発音時間となるが、G4E#16のように音階直後に長さを書くことで、その音階の発音時間の指定可能。
R	休符 ● 音階のように直接音長指定可能。
N 数値 (O4C=40)	音程数値指定 (N0～N127) ● 音程に対応する数値でキーオン。
O 数値	オクターブ (O0～O8) オクターブを変更。
<	オクターブを上げる
>	オクターブを下げる
@D 数値	デチューン(@D-128～@D127) ● 微妙な周波数のズレを設定。
(ｱﾝﾀﾞｰｽｺｱ)	ポルタメント 2音の周波数をなめらかに変化。

音量や定位に関するコマンド	
V 数値	ベロシティ (V0～V127) ● 音階発音時の音量変更。
(数値	ベロシティ加算 ((0～(127) 直前のベロシティ値に加算。
) 数値	ベロシティ減算 ((0～(127) 直前のベロシティ値に減算。
P 数値	パンポット (P0～P127) ● 定位変更。P64=中央
@V 数値	チャンネル音量(@V0～@V127) ● チャンネル音量変更。

音色に関するコマンド	
@ 数値 @144番以降の音色は楽器音よりも音が大きく聞こえます。	音色 (@0～@255) 楽器の種類を変更。 0～127 GM互換楽器 128～129 ドラムセット 144～150 PSG 151 ノイズ 224～255 ユーザー定義波形
@E 値Attack, 値Decay, 値Sustain, 値Release ADSR数値は、小さいと遅く、大きいと速くなります。	エンベロープ指定 音の鳴り方を調節する機能。 各要素の値の範囲は0～127。 (例) "@E0,99,64,64" 
@ER	エンベロープ解除

特殊な制御に関するコマンド	
[	ループ開始 最大3段まで入れ子可能。
] 数値	ループ終了 (0～255) ● 直前のループ開始記号まで演奏位置を戻します。（0または回数指定が無い場合無限ループ）

MML登録時、ループ記号内に 音階・休符・N の何れかが含まれていない場合、エラーとなります。

マクロ定義に関するコマンド	
{ ラベル=MML }	マクロの定義 よく使うフレーズを定義します（ラベルは最大８文字まで）
{ ラベル }	マクロ定義の利用 あらかじめ定義されたMMLフレーズを利用します。

演奏表現に関するコマンド	
@MON	モジュレーションON @MA、@MPで事前に設定されているモジュレーションを始動。
@MOF	モジュレーションOFF モジュレーションの停止。
@MA 値Depth, 値Range, 値Speed, 値Delay	トレモロ指定（各値0～127） 音量の揺れを調節する機能。 (例) "@MA0,99,64,64"
@MP 値Depth, 値Range, 値Speed, 値Delay	ビブラート指定（各値0～127） 周波数の揺れを調節する機能。 (例) "@MP0,99,64,64"

@MAと@MPは同時に使うことができません。設定と同時に影響が出ます。

標準楽器(@0〜@127)

番号	楽器名
0	Acoustic Grand Piano
1	Bright Acoustic Piano
2	Electric Grand Piano
3	Honky-tonk Piano
4	Electric Piano 1
5	Electric Piano 2
6	Harpsichord
7	Clavi
8	Celesta
9	Glockenspiel
10	Music Box
11	Vibraphone
12	Marimba
13	Xylophone
14	Tubular Bells
15	Dulcimer
16	Drawbar Organ
17	Percussive Organ
18	Rock Organ
19	Church Organ
20	Reed Organ
21	Accordion
22	Harmonica
23	Tango Accordion
24	Acoustic Guitar (nylon)
25	Acoustic Guitar (steel)
26	Electric Guitar (jazz)
27	Electric Guitar (clean)
28	Electric Guitar (muted)
29	Overdriven Guitar
30	Distortion Guitar
31	Guitar Harmonics
32	Acoustic Bass
33	Electric Bass (finger)
34	Electric Bass (pick)
35	Fretless Bass
36	Slap Bass 1
37	Slap Bass 2
38	Synth Bass 1
39	Synth Bass 2
40	Violin

41	Viola
42	Cello
43	Contrabass
44	Tremolo Strings
45	Pizzicato Strings
46	Orchestral Harp
47	Timpani
48	String Ensembles 1
49	String Ensembles 2
50	Synth Strings 1
51	Synth Strings 2
52	Voice Aahs
53	Voice Oohs
54	Synth Voice
55	Orchestra Hit
56	Trumpet
57	Trombone
58	Tuba
59	Muted Trumpet
60	French Horn
61	Brass Section
62	Synth Brass 1
63	Synth Brass 2
64	Soprano Sax
65	Alto Sax
66	Tenor Sax
67	Baritone Sax
68	Oboe
69	English Horn
70	Bassoon
71	Clarinet
72	Piccolo
73	Flute
74	Recorder
75	Pan Flute
76	Blown Bottle
77	Shakuhachi
78	Whistle
79	Ocarina
80	Lead 1 (square)
81	Lead 2 (sawtooth)
82	Lead 3 (calliope)
83	Lead 4 (chiff)
84	Lead 5 (charang)

85	Lead 6 (voice)
86	Lead 7 (fifths)
87	Lead 8 (bass + lead)
88	Pad 1 (new age)
89	Pad 2 (warm)
90	Pad 3 (polysynth)
91	Pad 4 (choir)
92	Pad 5 (bowed)
93	Pad 6 (metallic)
94	Pad 7 (halo)
95	Pad 8 (sweep)
96	FX 1 (rain)
97	FX 2 (soundtrack)
98	FX 3 (crystal)
99	FX 4 (atmosphere)
100	FX 5 (brightness)
101	FX 6 (goblins)
102	FX 7 (echoes)
103	FX 8 (sci-fi)
104	Sitar
105	Banjo
106	Shamisen
107	Koto
108	Kalimba
109	Bag pipe
110	Fiddle
111	Shanai
112	Tinkle Bell
113	Agogo
114	Steel Drums
115	Woodblock
116	Taiko Drum
117	Melodic Tom
118	Synth Drum
119	Reverse Cymbal
120	Guitar Fret Noise
121	Breath Noise
122	Seashore
123	Bird Tweet
124	Telephone Ring
125	Helicopter
126	Applause
127	Gunshot

ドラム(@128〜@129)

@128 (または@129)で音色を設定すると、O1B〜O5Aまでの音階に応じた打楽器音が鳴ります。

音階	楽器名	
O1B	Acoustic Bass Drum 2	909BD
O2C	Acoustic Bass Drum 1	808BD
O2C#	Side Stick	←
O2D	Acoustic Snare	808SD
O2D#	Hand Clap	←
O2E	Electric Snare	909SD
O2F	Low Floor Tom	808Tom LF
O2F#	Closed Hi-hat	808CHH
O2G	High Floor Tom	808Tom HF
O2G#	Pedal Hi-hat	808CHH
O2A	Low Tom	808Tom L
O2A#	Open Hi-hat	808OHH
O2B	Low-Mid Tom	808Tom LM
O3C	High Mid Tom	808Tom HM
O3C#	Crash Cymbal 1	808Cymbal
O3D	High Tom	808Tom H
O3D#	Ride Cymbal 1	←
O3E	Chinese Cymbal	←
O3F	Ride Bell	←
O3F#	Tambourine	←
O3G	Splash Cymbal	←
O3G#	Cowbell	808Cowbell
O3A	Crash Cymbal 2	←
O3A#	Vibra-slap	←
O3B	Ride Cymbal 2	←
O4C	High Bongo	←
O4C#	Low Bongo	←
O4D	Mute Hi Conga	808Conga MH
O4D#	Open Hi Conga	808Conga OH
O4E	Low Conga	808Conga L
O4F	High Timbale	←
O4F#	Low Timbale	←
O4G	High Agogo	←
O4G#	Low Agogo	←
O4A	Cabasa	←
O4A#	Maracas	808Maracas
O4B	Short Whistle	←
O5C	Long Whistle	←
O5C#	Short Guiro	←
O5D	Long Guiro	←
O5D#	Claves	808Claves

O5D#	Claves	←Claves
O5E	Hi Wood Block	←
O5F	Low Wood Block	←
O5F#	Mute Cuica	←
O5G	Open Cuica	←
O5G#	Mute Triangle	←
O5A	Open Triangle	←

PSGと波形(@144～@255)

番号	楽器名
144	デューティー比　12.5%　のPSG
145	デューティー比　25.0%　のPSG
146	デューティー比　37.5%　のPSG
147	デューティー比　50.0%　のPSG
148	デューティー比　62.5%　のPSG
149	デューティー比　75.0%　のPSG
150	デューティー比　87.5%　のPSG
151	ノイズ

224 : 255	※BGMPRG命令で生成されたユーザー波形

TALKコマンド

コマンド	説明
カナ文字	そのまま文字を発音
’	アクセント
/	アクセント区切り
|	フレーズ区切り
_(ｱﾝﾀﾞｰｽｺｱ)	一時停止
.	文末
?	文末(疑問)
!	文末(驚き)
%	直前母音の無効化
@H 数値	イントネーション変更（H0～H3）
	H0=標準語
	H1=ギャル風
	H2=外人風
	H3=関西弁風
@N 数値	ピッチ制御(N0～N3500)
	イントネーションや話者感情などの設定によって正確な音程では発音されない可能性があります。
	＜参考値＞
	N1447=C3
	N1510=C#3
	N1559=D3
	N1626=D#3
	N1686=E3
	N1745=F3
	N1792=F#3
	N1858=G3
	N1916=G#3
	N1971=A3
	N2031=A#3
	N2093=B3
	N2137=C4
@T 数値	発声速度（T0～T1000）
@V 数値	ボリューム（V0～V80）
@S 数値	話者変更（S0～S11）
	S0=若い男性
	S1=若い女性
	S2=男性
	S3=女性
	S4=年老いた男性
	S5=年老いた女性
	S6=男の子
	S7=女の子
	S8=若い女性姉妹
	S9=アニメキャラクタ
	S10=宇宙人
	S11=大男

@E 数値 先に@S話者設定を行ってください	感情変更（E0～E16） E0=怒ったように E1=ビジネスライクに E2=穏やかに E3=落胆したように E4=興奮したように E5=裏声で E6=幸せそうに E7=大きな声で E8=単調に E9=元気に E10=静かに E11=皮肉っぽく E12=おびえたように E13=叫んだように E14=緊張したように E15=ささやき声で E16=歌うように

11 BASIC基本仕様

BASIC基本仕様

SmileBASICの基本仕様や制限事項について。

基本要素

文字	マルチバイト文字で管理
文字種類	数字、英数字、カナ、記号
数の表現	32ビット固定小数(四捨五入) 4096を1.0として扱う。 整数部は、±524287の範囲。 範囲外の数値は扱えません。
16進表記	&H
2進表記	&B
変数	英字で始まる最大16文字まで、ただし _(ｱﾝﾀﾞｰｽｺｱ) は受け付ける。文字列型の変数は、名前の最後に＄をつける。 (例) ANS=75:C$="TEXT"
ラベル文字列	@で始まるラベル名が入った文字列型変数
配列	要素数の合計は262144個、2次元配列まで対応。括弧は () 又は []。添字は0から始まる。 (例) DIM NO(10)の範囲は、 NO(0)～NO(9)まで。 文字列用の変数は最大4096個までしか利用できません。配列としての定義と実際に利用できる個数が異なります。最大数を超えるとエラーとなります。
複数命令	可能（ ：(ｺﾛﾝ)で区切る）
サブルーチンとネスト	制限無し、FOR～NEXT も同様（メモリーの範囲内）
ファイル制御構造	ファイル読み書き中は専用ダイアログが表示されて終わるまで強制待機。
ファイル読み書き単位	リソース単位（ユーザーによる自由な読み書きはできません）
ファイル名	英字で始まる8文字以内 （ 'A'～'Z' , '_' , '0'～'9' ）

演算子(算術・比較・ビット)

◆算術演算子は、以下の5種類

+	加算 (A+B)
-	減算 (A-B)
*	乗算 (A＊B)
/	除算 (A/B) ※0除算はエラー
%	剰余 (A%B) ※0除算はエラー

文字列変数は加算と乗算が利用できます。

(例) 文字列の足し算

A$="ABC":B$="XYZ":PRINT A$+B$

結果→ ABCXYZ

(例) 文字列の掛け算(乗算)

A$="ABC"*4:PRINT A$

結果→ ABCABCABCABC

◆比較演算子は、以下の6種類

＞	左辺が右辺より大きい (A＞B)
＜	左辺が右辺より小さい (A＜B)
＞＝	左辺が右辺より大きいか等しい (A＞=B) ※=＞は禁止
＜＝	左辺が右辺より小さいか等しい (A＜=B) ※=＜は禁止
＝＝	両辺が等しい (A==B)
！＝	両辺が等しくない (A!=B)

◆ビット演算子は、以下の4種類

AND	論理積 (A AND B)
OR	論理和 (A OR B)
XOR	排他的論理和 (A XOR B)
NOT	否定 (NOT A)

◆真偽を反転する演算子

!	真偽反転記号 ※ !TRUE は、FALSEと同じ ※ !FALSEは、TRUEと同じ

◆演算子優先順位

1	（） [] で囲まれた部分
2	マイナス　NOT　！
3	関数
4	＊　／　％（乗除余）
5	＋　－
6	==　!=　＜　＜=　＞　＞=
7	AND　OR　XOR

編集機能

テキスト用のメモリーは約52万文字程度の文字を扱うことができます。これ以上の文字を入力した場合は、最大行数（行内の最大文字数）以内でも登録できなくなります。

エディタ	「編集モード」に搭載される行単位のテキストエディタ。ENTERキーの入力で自動的に行番号が挿入される。テキストの折り返しは行わない。
行の範囲	1～9999（1行の文字数が多い時、最大行数まで使えない）
行番号の扱い	テキストエディタの行の番号として扱う。改行で自動的に増加し、GOTOやGOSUBなどの命令は、行番号指定はできない仕様となっている。分岐命令については、行番号ではなく「@ラベル名」を利用する。
1行の文字数	1行100文字まで、画面に表示できない範囲は横方向へのスクロールで確認。（テキスト用のメモリーが不足すると1行内の文字数に満たない状態でも入力できなくなります）

入力装置

キーボード	ソフトウェアキーボード （英字、カナ、記号）
ハードウェアボタン	利用可能（セレクトボタンはESCキーとして利用、Lボタンとスタートボタンは実行時のみ開放）
タッチパネル	システム変数 TCHST、TCHX、TCHY で利用可能

表示関係

<table>
<tr><td>表示優先順位</td><td>ユーザーSPRITE及びグラフィック面は、ユーザー用BG面に対して優先順位設定可能。</td></tr>
<tr><td>前

上画面
表示階層

奥</td><td>コンソール画面
ユーザー用BG面（前）
ユーザー用SPRITE面
ユーザー用BG面（奥）
グラフィック面
背景面</td></tr>
<tr><td>前

下画面
表示階層

奥</td><td>キーボードおよびパネル
ユーザー用BG面（前）
ユーザー用SPRITE面
ユーザー用BG面（奥）
グラフィック面
背景面</td></tr>
<tr><td>解像度</td><td>256x192ドット</td></tr>
<tr><td>コンソール文字数</td><td>32文字x24行（最終行で改行すると1行スクロールする）</td></tr>
<tr><td>アニメ機能</td><td>SPRITE命令による簡易アニメ再生</td></tr>
<tr><td>キャラ機能</td><td>8ｘ8ドット単位の画像。SPRITEやBG専用のリソース(CHR形式)。<table><tr><td>BGU</td><td>ユーザー用4バンク(BGU0～BGU3)</td></tr><tr><td>SPU</td><td>ユーザー用8バンク(SPU0～SPU7)</td></tr></table></td></tr>
<tr><td>同時発色数</td><td>上画面と下画面に独立した色の指定が可能。下記色数の合計。<table><tr><td>BG</td><td>透明含む16色x16種</td></tr><tr><td>SP</td><td>透明含む16色x16種</td></tr><tr><td>GRP</td><td>256色(0番は透明色)</td></tr></table></td></tr>
<tr><td>パレット</td><td>SPRITEとBG用の色は、透明1色を含む16色を1パレットと呼びパレット単位で色番号を指定する。</td></tr>
<tr><td>文字色</td><td>文字用の色は、SPRITEとBG用に定義されている各パレット内の色番号15番目を利用している。この色を変更した場合、画面内の文字色も変化する。</td></tr>
<tr><td>背景色</td><td>BG用のパレット0番の0番目の色は、背景色として利用されます。</td></tr>
</table>

サウンド関係

同時発音数	BGMと効果音と音声合成を合わせて16音(PSG含まず)。
効果音	プリセット音70種類。周波数、音量、パンポットの変更。同時に8音まで再生。
音声合成	カタカナ文字列で音声を発声。話者、感情、ピッチ等の変更。
BGM用音源	128種類の楽器音。 2種類のドラムセット。 PSGとノイズ音源。 ユーザー定義簡易波形音源。
BGM用波形定義	32種類の波形定義に対応。
BGM	プリセット30曲。 ユーザー定義128曲。 最大8曲同時に演奏が可能。

エラー番号表

BASICを実行中に発生するエラーや警告表現には以下の要素があります。エラー発生時、システム変数ERRにエラー番号、ERLに行番号が記録されます。

1	Syntax error	おかしな文法が含まれています。
2	Out of range	数値が有効範囲を超えました。
3	Out of memory	メモリーが不足しています。
4	Undefined label	分岐先が見つかりません。
5	NEXT without FOR	FORに対応しないNEXTがあります。
6	Out of DATA	READで取得するためのDATAが不足しています。
7	Illegal function call	関数や命令の機能の指定方法が間違っています。
8	Duplicate definition	配列や変数を二重に定義しました。
9	Can't continue	CONTでプログラムを再開できません。
10	Missing operand	パラメータが不足しています。
11	Duplicate label	ラベルを二重に定義しました。
12	Illegal resource type	指定されたリソース種類文字列は存在しません。
13	Illegal character type	指定された種類のキャラは存在しません。
14	String too long	文字列が長すぎます。ラベルは8文字、文字列は256文字以内で収めてください。
15	Division by zero	0による除算を行いました。
16	Overflow	演算結果が許容範囲を超えました。
17	Subscript out of range	配列変数の添字が範囲外です。
18	Type mismatch	変数の型が一致しません。
19	Fomula too complex	式が複雑すぎます（括弧が多すぎるなど）。
20	RETURN without GOSUB	GOSUBがないのにRETURNがあります。
21	FOR without NEXT	NEXTに対応しないFORがあります。
22	Illegal MML	MMLに間違いがあります。

12 表示と文字

表示と文字

画面や色などの概念と文字やキーボードに関する情報。

表示画面構成

上画面

画面最奥から、背景、グラフィック画面、BGスクリーン後、BGスクリーン前、コンソール画面の5つの画面で構成されています。

ユーザープログラム実行中、SPSETおよびSPCHR命令でグラフィック画面、BG後、BG前の前後にSPRITEの表示優先順位を変更することができます。

背景色	全ての画面の後ろに表示される単色面
グラフィック画面	線や円を描いたり色を塗ったりすることができる画面
ユーザー用BGスクリーン後	後ろに表示されるBGスクリーン面 (8x8キャラを64x64個分管理)
ユーザー用BGスクリーン前	手前に表示されるBGスクリーン面 (8x8キャラを64x64個分管理)
コンソール画面	コンソール画面をBG2枚で表現

下画面

画面最奥から、背景、グラフィック画面、BGスクリーン後、BGスクリーン前、キーボード・パネル部品の5つの画面で構成されています。普段はキーボード表示用画面。プログラム実行時にPNLTYPE命令でBGやグラフィックを利用することができます。

背景色	全ての画面の後ろに表示される単色面
グラフィック画面	線や円を描いたり色を塗ったりすることができる画面
ユーザー用BG前後	ユーザー用BGスクリーン
キーボード、パネル部品	キーボード、操作パネル、およびファイルダイアログ等の表示

カラーパレット

本製品には、コンソールとBGスクリーン、およびSPRITEとグラフィックという表示用要素がありますが、

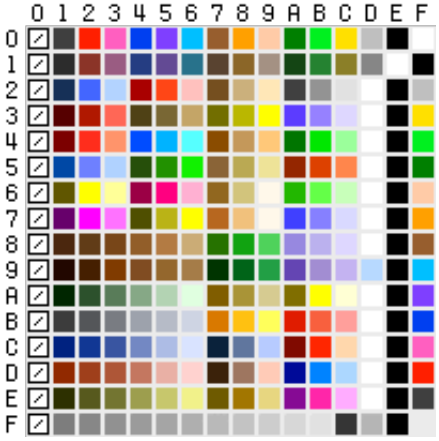
- コンソールの文字とBGスクリーン
- SPRITE
- グラフィック

それぞれに独立したカラーパレットという色の管理構造があります。文字とBGおよびSPRITEの色は、16色（0番は透明）単位で利用することができますが、グラフィックでは256色利用可能です。

BGやSPRITEで使う色は16色ずつに区切って利用し、横方向を色番号、縦をパレット番号と呼びます。

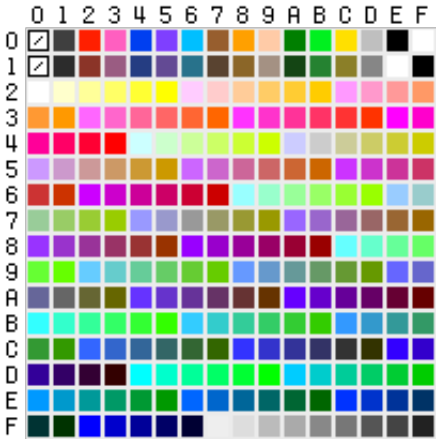
COLSET命令で好きな色に変更することができます。基本的に、左側に濃い色、右に行くほど明るい色を定義しておくことでパレット切り替え時に極端な色の乱れを防ぐことができます。

◆コンソール文字とBG及びSPRITE用



用途ごとに3色ずつ割り当てますが、色の定義は、暗い・普通・明るいになるようにします。

◆グラフィック用

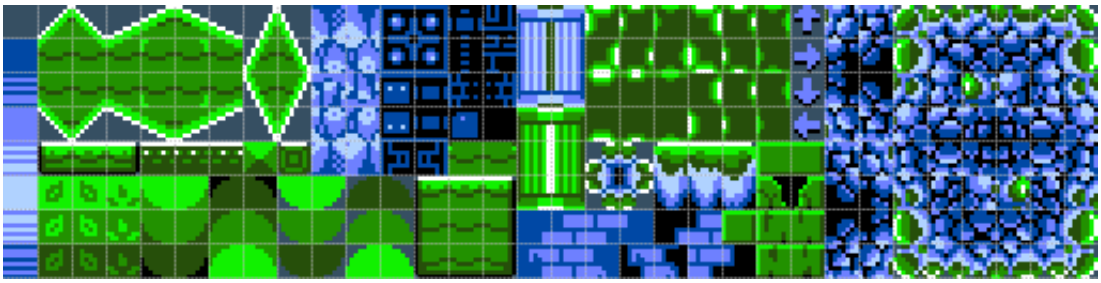


ユーザー用BGキャラ

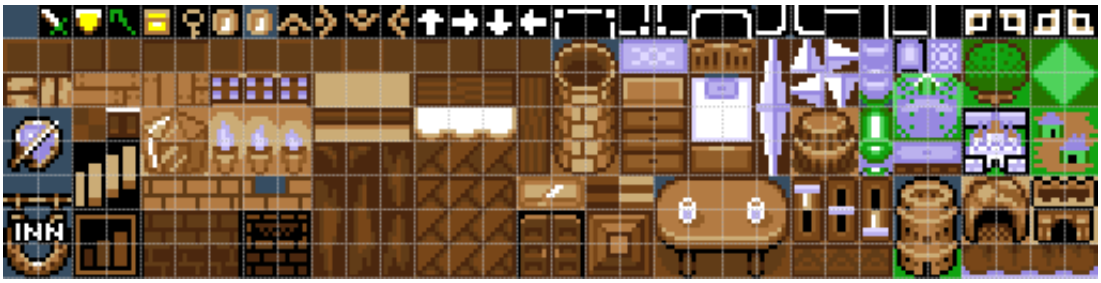
LOADで引数を指定し、画面のBGキャラクタとして利用します。CHRINITで初期状態に戻すことができます。

BGU0

BGU1



BGU2



BGU3

ユーザー用SPRITEキャラ

SPRITE画面上で4方向にキャラを移動するアニメーションを表示する場合、上下左右の各方向に動くSPRITEキャラ番号をSPUリソースから指定します。

たとえば、SPU1リソース上には、SPRITEキャラ番号64～67(右)、68～71(下)、72～75(左)、76～79(上)という番号に時計回り順に移動アニメーション画像が並んでいます。SPANIM命令で簡単なアニメーション再生が可能です。また、キャラ画像はユーザーが自由に書き換えることもできます。

キーボード表

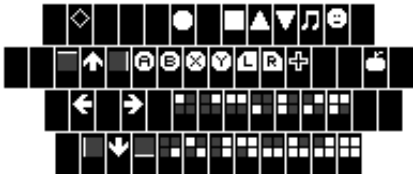
文字列をタッチするとコンソール画面内に対象となる文字列が流し込まれます。SHIFTキー（または本体Lボタン）を押すと、入力できる文字列の配置が換わります。

◆アルファベット

◆アルファベット+SHIFT (または本体Lボタン)

◆記号

◆記号+SHIFT (または本体Lボタン)



◆カナ

◆カナ+SHIFT (または本体Lボタン)

文字コード表

内蔵されている文字は以下の256種類。

既存BASICのように、0番～31番までのコントロールコードは存在しません。

13 ファイルとリソース

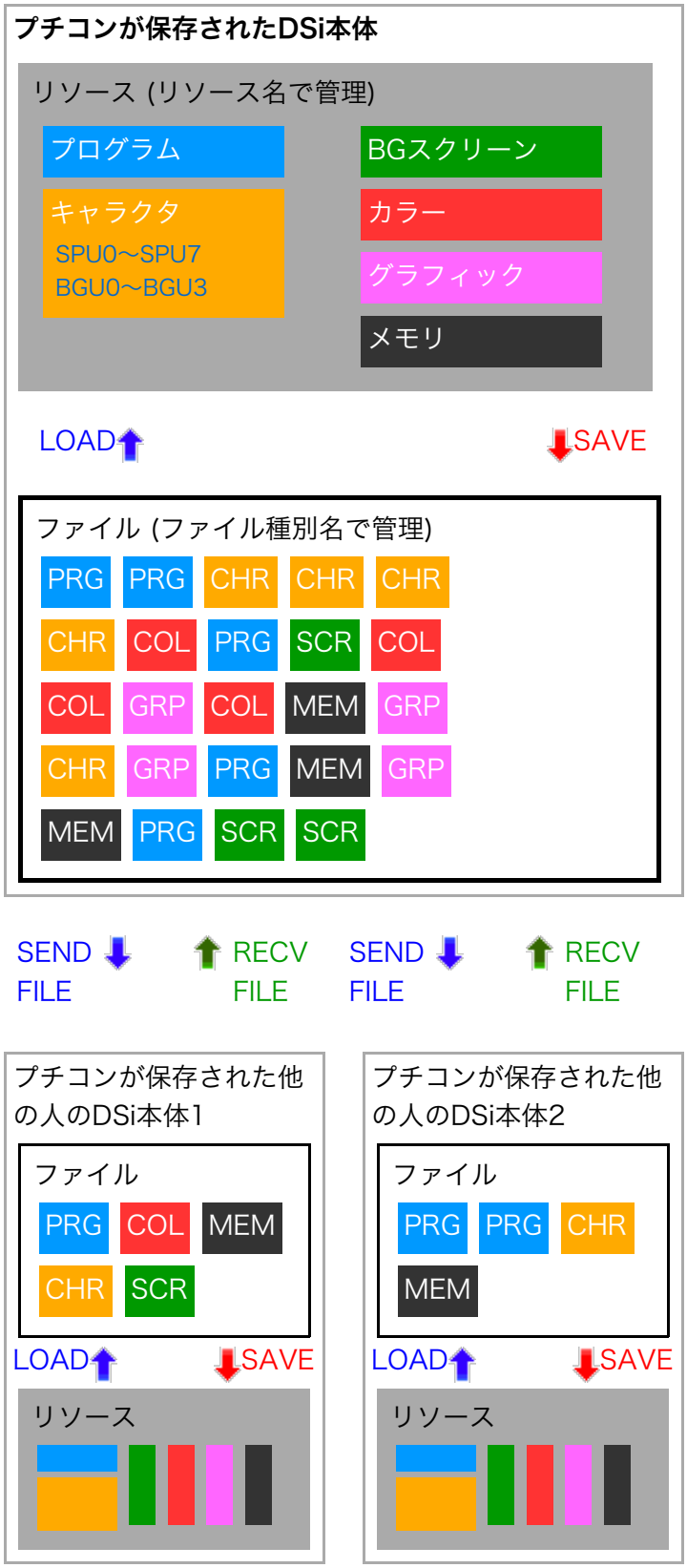
ファイルとリソース

プログラム内でファイルをアクセスする場合、およびプチコンが保存された他のDSi本体への送受信に関しての遷移。

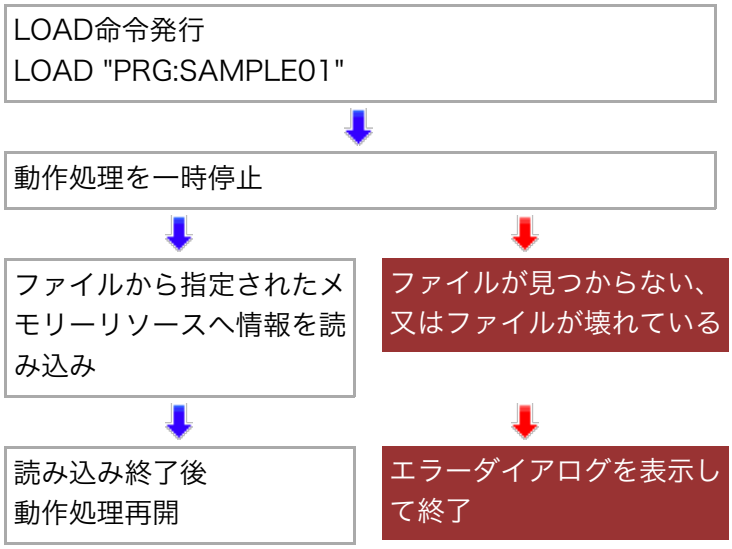
対象となる命令
LOAD、SAVE、SENDFILE、RECVFILE

ファイルとリソースの関係

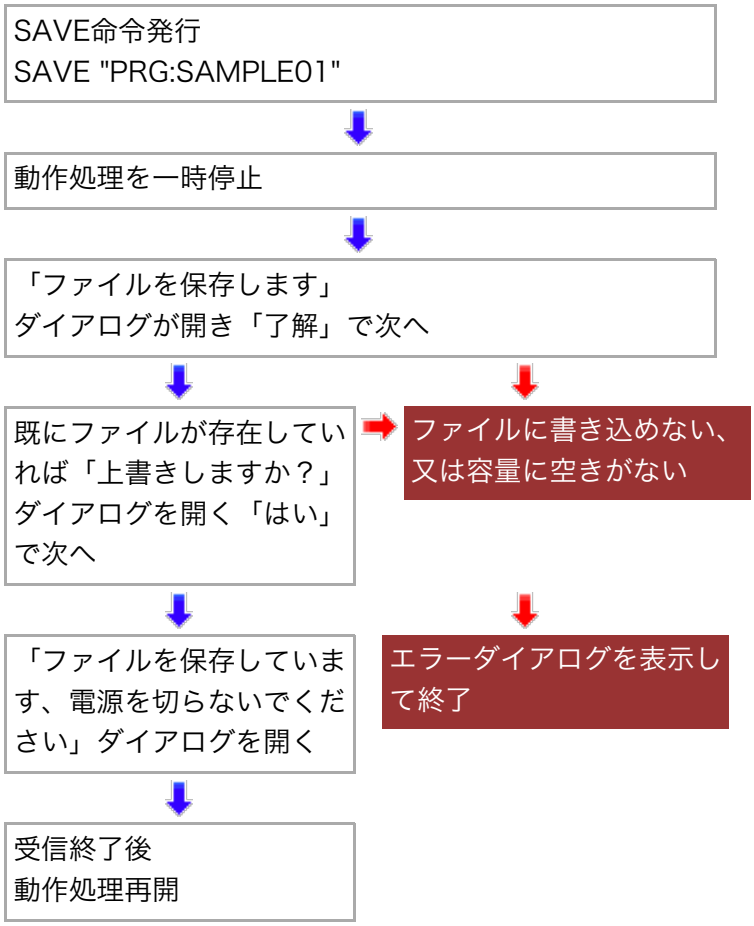
LOAD・SAVEは、内部リソースとファイル間の読み書きを行い、RECVFILE や SENDFILEは、プチコンが保存された他のDSi本体との通信によるファイルへの読み書きに使用します。



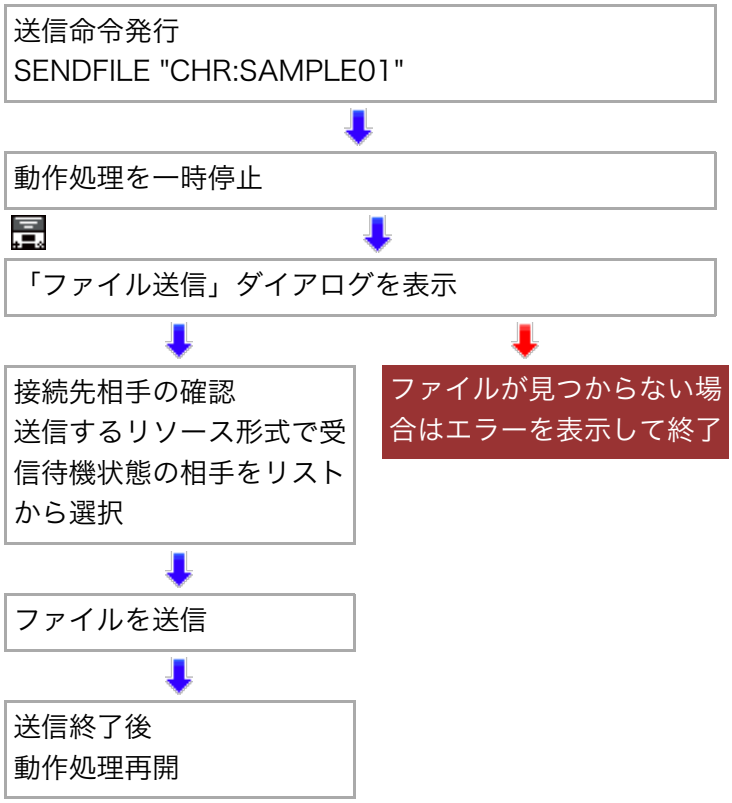
LOADの流れ



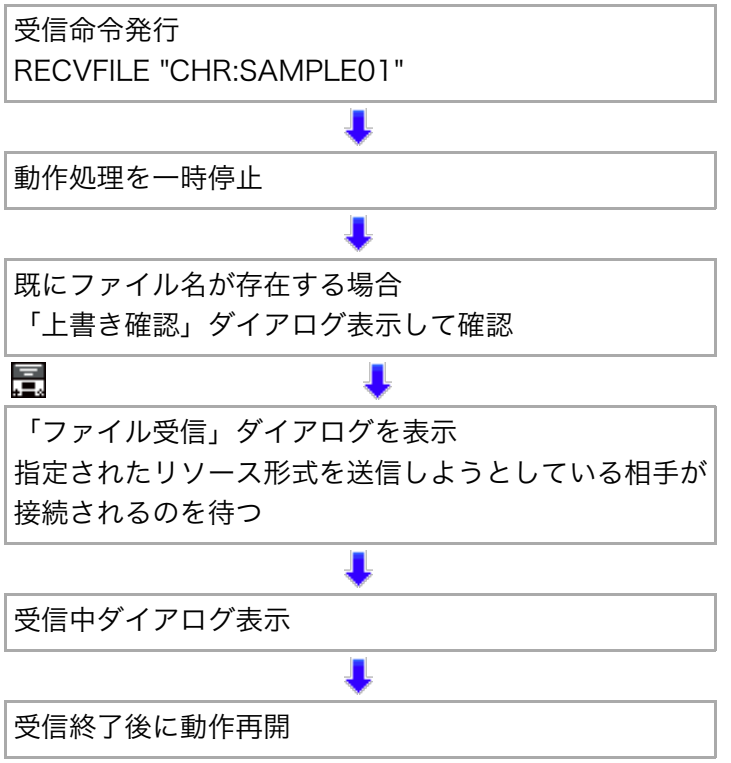
SAVEの流れ



SENDFILE（送信）の流れ



RECVFILE (受信) の流れ



14 命令・関数に関する規則

命令・関数に関する規則

BASICに搭載されている様々な命令や関数について表記ルールのまとめ。

命令方針

コンピュータに対する指示を出すコマンドとして統一した書式を優先。

◎基本書式

命令　[引数] [, 引数] [, 引数] ...

命令の独自書式方針

既存BASICにある命令ごとに変化する書式は引き継がず独自書式を優先。

◎既存BASICの書式(視認性を重視)

LINE(x,y)-(x2,y2),color

◎独自書式

GLINE x,y,x2,y2,color

関数方針

独自書式によって命令は '(' なしで実現させたため、関数は命令直後に必ず '(' を付ける仕様とした。

◎関数

変数 = COLGET$("種類名", 番号)

複数の値を取得する命令の方針

関数と命令の中間的位置づけ

複数の情報を取得する関数は、C言語であればポインタ渡しが使えますが、BASICでは、ポインタの概念がないため受け取る情報ごとに関数が必要。本製品では、READの処理系を応用して複数の情報渡しを導入した。

◎複数値を取得する命令

TMREAD("時間文字列") ,H ,M ,S

値を取得する場合、引数が必要なケースが多いため引数渡しも可能。受け取る引数の型や、引数の数が変わるような書式には対応していない。

15 アルファベット順簡易表

アルファベット順簡易表

SmileBASIC に内蔵されている命令およびシステム変数等の予約語を、アルファベット順に並べたもの。単純な命令の名称については、キーボードから1文字入力することで候補が表示されますが、引数の数や内容はこの表をご利用ください。

A, B, C, D

A
変数=ABS(数値)
ACLS
AND
APPEND "ﾌｧｲﾙ名"
変数=ASC(文字)
変数=ATAN(ﾗｼﾞｱﾝ値) 変数=ATAN(目的y-y, 目的x-x)

B
BEEP [波形番号 [,ﾋﾟｯﾁ [,音量 [,ﾊﾟﾝﾎﾟｯﾄ]]]]
変数=BGCHK(ﾚｲﾔｰ)
BGCLIP 始点x,始点y,終点x,終点y
BGCLR [ﾚｲﾔｰ]
BGCOPY ﾚｲﾔｰ,始点x,始点y,終点x,終点y,転送先x,転送先y
BGFILL ﾚｲﾔｰ,始点x,始点y,終点x,終点y,ｷｬﾗ番号,ﾊﾟﾚｯﾄ番号,横反転,縦反転 BGFILL ﾚｲﾔｰ,始点x,始点y,終点x,終点y,ｽｸﾘｰﾝﾃﾞｰﾀ BGFILL ﾚｲﾔｰ,始点x,始点y,終点x,終点y,"ｽｸﾘｰﾝﾃﾞｰﾀ文字列"
変数=BGMCHK([ﾄﾗｯｸ番号])
BGMCLEAR [曲番号]
変数=BGMGETV(ﾄﾗｯｸ番号,変数番号)
BGMPLAY 曲番号 BGMPLAY ﾄﾗｯｸ番号,曲番号[,ﾄﾗｯｸ音量] BGMPLAY "MML文字列"[,"MML文字列"...]
BGMPRG 波形番号,A,D,S,R,"波形文字列"
BGMSET 曲番号,"MML文字列"[,"MML文字列"]
BGMSETD 曲番号,@ラベル BGMSETD 曲番号,変数$
BGMSETV ﾄﾗｯｸ番号,変数番号,値
BGMSTOP [ﾄﾗｯｸ番号[,ﾌｪｰﾄﾞ時間]]
BGMVOL [ﾄﾗｯｸ番号,] 音量
BGOFS ﾚｲﾔｰ,x,y [,補間時間]
BGPAGE 画面
BGPUT ﾚｲﾔｰ,x,y,ｷｬﾗ番号,ﾊﾟﾚｯﾄ番号,横反転,縦反転 BGPUT ﾚｲﾔｰ,x,y,ｽｸﾘｰﾝﾃﾞｰﾀ BGPUT ﾚｲﾔｰ,x,y,"ｽｸﾘｰﾝﾃﾞｰﾀ文字列"
BGREAD (ﾚｲﾔｰ,x,y),CHR,PAL,H,V BGREAD (ﾚｲﾔｰ,x,y),SC BGREAD (ﾚｲﾔｰ,x,y),SC$
BREPEAT ﾎﾞﾀﾝID [,開始時間,ｲﾝﾀｰﾊﾞﾙ]
変数=BTRIG()
変数=BUTTON([種別])

C
CANCEL
変数=CHKCHR(x座標,y座標)
変数$=CHR$(文字ｺｰﾄﾞ)
CHRINIT "ｷｬﾗ名"
CHRREAD ("ｷｬﾗ名",ｷｬﾗ番号),C$
CHRSET "ｷｬﾗ名",ｷｬﾗ番号,"画像文字列"
CLEAR
CLS
COLINIT "色ﾊﾞﾝｸ名",色番号
COLOR 文字色 [,背景色]
COLREAD ("色ﾊﾞﾝｸ名",色番号),R,G,B
COLSET "色ﾊﾞﾝｸ名",色番号,"色情報文字列"
CONT
変数=COS(ﾗｼﾞｱﾝ値)
CSRX
CSRY

D
DATA 数値,数値,... DATA "文字列","文字列",... DATA 123,345,56,"SAMPLE"
DATE$
変数=DEG(ﾗｼﾞｱﾝ値)
DELETE "ﾌｧｲﾙ種別名:ﾌｧｲﾙ名"
DIM pos[4] DIM sample[10,5]
DTREAD("日付文字列"),YEAR,MON,DAY

E, F, G, H

E
ELSE
END
ERL
ERR
EXEC "PRG:ﾌｧｲﾙ名"
変数=EXP(数値)

F
FALSE
FILES [ﾌｧｲﾙ種別名 [ﾌｧｲﾙ種別名 ...]]
変数=FLOOR(数値)
FOR 変数=初期値TO終了値[STEP増加量]
FREEMEM
FREEVAR
FUNCNO

G
GBOX 始点x,始点y,終点x,終点y[,色]
GCIRCLE x,y,半径 [,色[,開始角,終了角]]
GCLS [色]
GCOLOR 色番号
GCOPY [転送ﾍﾟｰｼﾞ,] 始点x,始点y,終点x,終点y,転送先x,転送先y,ｺﾋﾟｰﾓｰﾄﾞ
GDRAWMD 状態
GFILL 始点x,始点y,終点x,終点y [,色]
GLINE 始点x,始点y,終点x,終点y [,色]
GOSUB @ﾗﾍﾞﾙ GOSUB 変数$
GOTO @ﾗﾍﾞﾙ GOTO 変数$
GPAGE 画面 [,描画ﾍﾟｰｼﾞ][,表示ﾍﾟｰｼﾞ]
GPAINT x,y [,色 [,境界色]]
GPSET x,y [,色]
GPRIO 番号
GPUTCHR x,y,"ｷｬﾗ名",番号,ﾊﾟﾚｯﾄ番号,ｽｹｰﾙ
変数=GSPOIT(x,y)

H
変数$=HEX$(数値[,桁数])

I, J, K, L

I
数値=ICONCHK()
ICONCLR [ｱｲｺﾝ位置]
ICONPAGE
ICONPMAX
ICONPUSE
ICONSET ｱｲｺﾝ位置,ｱｲｺﾝ番号
IF 条件式 GOTO @ﾗﾍﾞﾙ IF 条件式 GOTO @ﾗﾍﾞﾙ ELSE 不成立時実行される命令 IF 条件式 GOTO @ﾗﾍﾞﾙ ELSE @ﾗﾍﾞﾙ
IF 条件式 THEN 成立時に実行される命令 IF 条件式 THEN @ﾗﾍﾞﾙ IF 条件式 THEN 成立時に実行される命令 ELSE 不成立時に実行される命令 IF 条件式 THEN @ﾗﾍﾞﾙ ELSE @ﾗﾍﾞﾙ
変数$=INKEY$()
INPUT ["文字列";]受け取る変数 INPUT ["文字列";]受け取る変数$ INPUT ["文字列";]受け取る変数,受け取る変数$
変数=INSTR([開始位置],"文字列","検索対象文字列")

J

K
KEY 番号,"文字列"
KEYBOARD

L
@ﾗﾍﾞﾙ名
変数$=LEFT$("文字列",文字数)
変数=LEN(文字列)
LINPUT ["文字列";]受け取る変数$
LIST LIST@ﾗﾍﾞﾙ LIST行番号
LOAD "ﾘｿｰｽ名:ﾌｧｲﾙ名" [,表示制御]
LOCATE x,y
変数=LOG(数値)

M, N, O, P

M
MAINCNTL
MAINCNTH
MEM$
変数$=MID$(文字列,開始位置,文字数)

N
NEW
NEXT [変数名]
NOT

O
ON 変数 GOSUB @ﾗﾍﾞﾙ0,@ﾗﾍﾞﾙ1,@ﾗﾍﾞﾙ2...
ON 変数 GOTO @ﾗﾍﾞﾙ0,@ﾗﾍﾞﾙ1,@ﾗﾍﾞﾙ2...
OR

P
PACKAGE$
変数=PI()
PNLSTR x座標,y座標, "文字列"[,ﾊﾟﾚｯﾄ番号]
PNLTYPE "ﾊﾟﾈﾙ名"
変数=POW(数値,べき乗数値)
PRGNAME$
PRINT "文字列" PRINT 変数 PRINT 変数$ PRINT 変数;変数$;"文字列" PRINT "文字列",変数,変数$

Q, R, S, T

Q

R
変数=RAD(角度)
READ 取得変数1[,取得変数2...] READ A READ B$ READ X,Y,Z,G$
REBOOT
RECVFILE "ﾌｧｲﾙ種別名:ﾌｧｲﾙ名"
REM ｺﾒﾝﾄｷﾊｺﾒﾝﾄﾃﾞｽ 'commenttext
RENAME "ﾌｧｲﾙ種別名:ﾌｧｲﾙ名","新しい名前"
RESTORE @ﾗﾍﾞﾙ RESTORE 変数$
RESULT
RETURN
変数$=RIGHT$("文字列",文字数)
変数=RND(最大値)
RSORT 開始位置,要素数,配列1[,配列2...]
RUN

S
SAVE "ﾘｿｰｽ名:ﾌｧｲﾙ名"
SENDFILE "ﾌｧｲﾙ種別名:ﾌｧｲﾙ名"
変数=SGN(変数)
変数=SIN(ﾗｼﾞｱﾝ値)
SORT 開始位置,要素数,配列1[,配列2…]
SPANGLE 管理番号,角度 [,補間時間,変化方向]
SPANIM 管理番号,枚数,時間 [,ﾙｰﾌﾟ]
変数=SPCHK(管理番号)
SPCHR 管理番号,ｽﾌﾟﾗｲﾄｷｬﾗ番号 [,ﾊﾟﾚｯﾄ番号,横反転,縦反転,優先順位]
SPCLR [管理番号]
SPCOL 管理番号,x,y,w,h,ｽｹｰﾙ対応 [,ｸﾞﾙｰﾌﾟ]
SPCOLVEC 管理番号 [,移動量x,移動量y]
変数=SPGETV(管理番号,変数番号)
変数=SPHIT(管理番号 [,判定開始管理番号])
SPHITNO
変数=SPHITRC(管理番号,x,y,w,h [,移動量x,移動量y])
変数=SPHITSP(管理番号,相手管理番号)
SPHITT
SPHITX
SPHITY
SPHOME 管理番号,x,y
SPOFS 管理番号,x,y [,補間時間]
SPPAGE 画面
SPREAD(管理番号),X,Y [,A,S,C]
SPSCALE 管理番号,ｽｹｰﾙ [,補間時間]
SPSET 管理番号,ｽﾌﾟﾗｲﾄｷｬﾗ番号,ﾊﾟﾚｯﾄ番号,横反転,縦反転,優先順位 [,幅,高さ]
SPSETV 管理番号,変数番号,数値
変数=SQR(数値)
STEP
STOP
変数$=STR$(数値)
変数$=SUBST$("文字列",開始位置,文字数,"置換文字列")
SWAP 変数,変数 SWAP 変数$,変数$
SYSBEEP

T
TABSTEP
TALK "音声文字列" [,"音声文字列"...]
変数=TALKCHK()
TALKSTOP
変数=TAN(ﾗｼﾞｱﾝ値)
TCHST
TCHTIME
TCHX
TCHY
THEN
TIME$
TMREAD ("時間文字列"),HOUR,MIN,SEC
TO
TRUE

U, V, W, X, Y, Z

U

V
変数=VAL(文字列)
VERSION
VISIBLE ｺﾝｿｰﾙ,ﾊﾟﾈﾙ,BG0,BG1,SPRITE,ｸﾞﾗﾌｨｯｸ
VSYNC ﾌﾚｰﾑ数

W
WAIT ﾌﾚｰﾑ数

X
XOR

Y

Z

16 システム変数

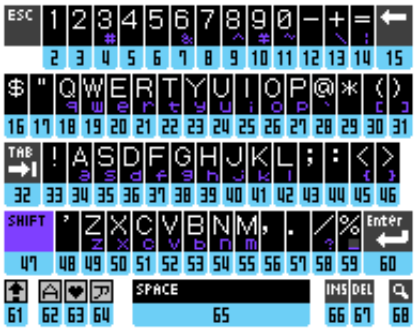
数値型システム変数

	R	W	(R=読み込み、W=書き込み)
CSRX	○	×	現在のカーソル横方向位置
CSRY	○	×	現在のカーソル縦方向位置
FREEMEM	○	×	残りユーザー用のメモリー容量
VERSION	○	×	システムのバージョン &HAABBCCDD 16進数表記
ERR	○	×	直前のエラー番号
ERL	○	×	エラー発生行番号
RESULT	○	×	ファイル系命令の実行結果
TCHX	○	×	タッチされたx座標
TCHY	○	×	タッチされたy座標
TCHST	○	×	タッチ状態 (TRUE=触られた)
TCHTIME	○	×	タッチされ続けている時間 （フレーム数）
MAINCNTL	○	×	起動時からの経過フレーム時間（最大145分）
MAINCNTH	○	×	起動時からの経過フレーム時間（145分以上の情報）
TABSTEP	○	○	TABによる移動量(0～16)
TRUE	○	×	必ず1
FALSE	○	×	必ず0
CANCEL	○	×	必ず-1
ICONPUSE	○	○	FALSE=使わない TRUE=使う
ICONPAGE	○	○	ユーザー用システムアイコンのページ番号（実行モードでは常に0が入ります）
ICONPMAX	○	○	ユーザー用システムアイコンのページ最大値(実行モードでは無効)
FUNCNO	○	×	押されているファンクションキーの番号(1～5、0=押されていない)
FREEVAR	○	×	登録可能な変数の数
SYSBEEP	○	○	システム効果音制御（TRUE=あり、FALSE=なし）
KEYBOARD	○	×	キースキャンコード
SPHITNO	○	×	SPRITE衝突判定の結果 （-1=なし、0～99=衝突）
SPHITX	○	×	SPRITE衝突時のx座標
SPHITY	○	×	SPRITE衝突時のy座標
SPHITT	○	×	SPRITE衝突時の時刻

文字列型システム変数

	R	W	(R=読み込み、W=書き込み)
TIME$	○	×	現在の時刻を文字列として取得（HH:MM:SS）
DATE$	○	×	現在の日付を文字列として取得（YYYY/MM/DD）
MEM$	○	○	ファイル保存可能な文字列
PRGNAME$	○	×	直前にPRG形式のファイルが読み込まれた場合にファイル名が格納される
PACKAGE$	○	×	直前に読み込まれたファイルのパッケージ情報が格納される

キースキャンコード



システム変数 KEYBOARD は、INKEY$では取得できないキーボードの情報を取得するための特別な変数です。得られる値は文字に対応する番号ではなくキーボードのボタンに対応する数値となります。何も入力が無い場合、KEYBOARD変数には0が入ります。

17 実行モード専用命令

実行モード専用命令

プログラムの実行、再開など「実行モード」のみで使用できる命令です(プログラム内記述不可)
NEW、LIST、RUN、CONT、FILES、REBOOT

NEW

プログラムを消す

書式	NEW	
引数	なし	
戻り	なし	
エラー		

LIST

「編集モード」への切り替えと編集開始

書式	LIST LIST @ラベル LIST 行番号	
引数	行番号	ソースが表示される行の指定
	@ラベル	ソースが表示される行の指定
戻り	なし	
エラー	行番号やラベルが存在しない時	

RUN

プログラムの実行

書式	RUN
引数	なし
戻り	なし
エラー	

CONT

STOP命令で停止中のプログラムを再開

書式	CONT
引数	なし
戻り	なし
エラー	再開できない状態で実行した時

FILES

ファイル一覧をコンソール上に表示

書式	FILES [ﾌｧｲﾙ種別名 [, ﾌｧｲﾙ種別名 ...]]
引数	ファイル種別名 (対象ファイルのリソース種類ごとに割り当てられた文字列)
	PRG プログラム(省略可能)
	MEM メモリー
	COL カラー
	GRP グラフィック
	SCR ユーザー用スクリーン
	CHR ユーザー用キャラ
戻り	なし
エラー	

REBOOT

BASICを終了してホームメニューに戻る

編集中のプログラムや加工したキャラ、色等の設定は失われます。後から使う可能性がある場合、SAVE命令で保存して下さい。

書式	REBOOT
引数	なし
戻り	なし
エラー	

18 宣言・代入命令

宣言・代入命令

メモリーの初期化、変数宣言などを行う以下の命令があります。
CLEAR、＝、DIM、REM、@、KEY

CLEAR

変数名やBASIC内部のメモリーを初期化

書式	CLEAR
引数	なし
戻り	なし
エラー	

=(LET)

代入(LET命令自体を省略)

書式	ABC=123 TEXT$="ABCDE"
引数	なし
戻り	なし
エラー	

DIM

配列宣言
要素数は2次元まで、合計262144個まで定義可能

書式	DIM pos(4),size(4) DIM sample(10, 5) DIM MSG$(15)
引数	なし
戻り	なし
エラー	

'(REM)

注釈(コメント)
この記号以降改行までは無視される。

書式	REM コノサキハコメントデス ' comment text
引数	なし
戻り	なし
エラー	

@(ラベル定義)

行に対する名前の宣言

必ず行の先頭に記述すること。@以降の文字列が名前として登録される。GOTOやGOSUBなどで分岐先として利用する。最大16文字まで。

書式	@ラベル名	
引数	ラベル名	文字列とは異なり""で囲む必要はない
戻り	なし	
エラー	英字・'_'・数字以外の文字や記号を割り当てた、およびラベル名が指定されていない時	

KEY

ファンクションキーに文字列を割り当て

ユーザープログラム内で利用する場合、押されたファンクションキーに登録された文字列がキー情報として内部的に流し込まれる。

書式	番号	1～5
引数	文字列	割り当てる文字列 (表示上は4文字までだが、内部的には256文字まで確保、表示しきれない場合、最後の文字を '.' として表示)
戻り	なし	
エラー	存在しない番号を指定した時	

19 変数操作・ウエイト命令

変数操作・ウエイト命令

変数の操作や、ウエイトなどの基本的な以下の命令があります。
SWAP、SORT、RSORT、VSYNC、WAIT

SWAP

2つの変数の内容を交換

書式	SWAP 変数1, 変数2 SWAP 変数$1, 変数$2	
引数	変数1	1つ目の変数
	変数2	2つ目の変数
戻り	なし	
エラー		

SORT

1次元配列を昇順(1→99)で並び替え

書式	SORT 開始位置, 要素数, 配列1 [,配列2 ...]	
引数	開始位置	並び替える最初の位置(0～
	要素数	並び替える個数
	配列1	対象となる配列名 ※()は不要
	[配列2]...	配列1の結果を元に並び替える配列(続けて配列名を書くと全ての配列を配列1の結果で並び替えます)
戻り	なし	
エラー		
例	配列Vの内容を元に配列IXの内容を昇順で並び替え CLEAR DIM IX(10),V(10) FOR I=0 TO 9 IX(I)=I : V(I)=RND(100) PRINT I;"=";IX(I);":";V(I) NEXT SORT 0,10, V, IX PRINT FOR I=0 TO 9 PRINT I;"=";IX(I);":";V(I) NEXT	

RSORT

1次元配列を降順(99→1)で並び替え

書式	RSORT 開始位置, 要素数, 配列1 [,配列2 ...]
引数	SORTを参照
戻り	なし
エラー	

VSYNC

画面更新周期との同期(描画更新待ち)

書式	VSYNC フレーム数	
引数	フレーム数	直前のVSYNC呼び出しからの経過フレーム数を指定(0=無視)
戻り	なし	
エラー		

WAIT

単純な時間待ち

書式	WAIT フレーム数	
引数	フレーム数	指定フレーム数分待機(フレーム数を60にすると1秒待つ)
戻り	なし	
エラー		

20 分岐系命令

分岐系命令

分岐条件、ルーチン呼び出しなどを行う以下の命令があります。
ON～GOTO、ON～GOSUB、GOTO、GOSUB、RETURN、STOP、END

ON～ GOTO

数値内容によって分岐

書式	ON 変数 GOTO 変数=0の時の分岐先@ラベル, 変数=1の時の分岐先@ラベル, 変数=2...	
引数	変数	分岐番号(0～
戻り	なし	
エラー		

ON～ GOSUB

数値内容によってサブルーチン呼び出し

書式	ON 変数 GOSUB 変数=0の時の分岐先@ラベル,変数=1の時の分岐先@ラベル, 変数=2...	
引数	変数	分岐番号(0～
戻り	なし	
エラー		

GOTO

プログラムの強制分岐
@ラベルの代わりに、文字型変数によるラベル文字列を利用することができます。

書式	GOTO @ラベル GOTO 変数＄	
引数	@ラベル	分岐先名
戻り	なし	
エラー		

GOSUB

サブルーチン呼び出し
@ラベルの代わりに、ラベル名が代入された文字列変数を利用することができます。

書式	GOSUB @ラベル GOSUB 変数＄	
引数	@ラベル	分岐先名
戻り	なし	
エラー		

RETURN

サブルーチンからの復帰
必ず GOSUB とセットで利用します。

書式	RETURN
引数	なし
戻り	なし
エラー	

STOP

実行中のプログラムを強制的に停止

書式	STOP
引数	なし
戻り	なし
エラー	

END

プログラムの終了

書式	END
引数	なし
戻り	なし
エラー	

21 くりかえし・比較命令

くりかえし・比較命令

指定回数の繰り返し、条件判断などを行う以下の命令があります。
FOR～TO～STEP、NEXT、IF～THEN～ELSE、 IF～GOTO～ELSE

FOR～ TO～ STEP

指定回数の繰り返し
STEPを省略した場合は、STEP1として扱います。増加量がプラスで初期値よりも終了値が小さい時、FOR命令をスキップしてNEXT以降の命令を実行します。

書式	FOR 変数=初期値 TO 終了値 [STEP 増加量]	
引数	変数	回数管理用
	初期値	始まりの数
	終了値	終わりの数
	増加量	1度に足す値
戻り	なし	
エラー		

NEXT

繰り返しの終わり
必ず FOR 命令とセットで利用します。

書式	NEXT [変数名]	
引数	変数名	繰り返す変数
戻り	なし	
エラー		

IF～ THEN～ ELSE

条件比較と分岐
IF命令は、複数行にまたがる記述はできません。

書式	IF 条件式 THEN 成立時命令 IF 条件式 THEN @ラベル IF 条件式 THEN 成立時命令 ELSE 不成立時命令 IF 条件式 THEN @ラベル ELSE @ラベル	
引数	条件式	比較する内容
	成立時	命令・分岐先
	不成立時	命令・分岐先
戻り	なし	
エラー		

IF～ GOTO～ ELSE

条件比較と分岐

IF命令は、複数行にまたがる記述はできません。

書式	IF 条件式 GOTO @ラベル IF 条件式 GOTO @ラベル ELSE 不成立時命令 IF 条件式 GOTO @ラベル ELSE @ラベル	
引数	条件式	比較する内容
	成立時	命令・分岐先
	不成立時	命令・分岐先
戻り	なし	
エラー		

22 読み込み系命令

読み込み系命令

データの読み込みなどを行う以下の命令があります。
READ、DATA、RESTORE、TMREAD()、DTREAD()

READ

DATAの情報を読み込む

書式	READ 取得変数1 [, 取得変数2 ...] READ A,B,C READ X$,Y$ READ X,Y,Z,G$	
引数	取得変数...	DATA から読み込む情報を格納する変数(複数指定可)
戻り	なし	
エラー	読み込むDATAが不足した時	

DATA

READで読み込むデータの定義
数値と文字は混在可能。

書式	DATA 数値, 数値,... DATA "文字列", "文字列",... DATA 123,"SAMPLE",...	
引数	情報	数値や文字列を ',' で区切って並べる
戻り	なし	
エラー		

RESTORE

READが読み込むDATAの位置を変更
@ラベルの代わりに、文字型変数によるラベル文字列を利用することができます。

書式	RESTORE @ラベル RESTORE 変数$	
引数	@ラベル	取得位置
戻り	なし	
エラー		

TMREAD()

時間文字列を数値に変換

書式	TMREAD("時間文字列"), HOUR, MIN, SEC	
引数	時間文字列	HH:MM:SS 形式の時間文字列
	HOUR	時間を受け取る変数
	MIN	分を受け取る変数
	SEC	秒を受け取る変数
戻り	なし	
エラー		

DTREAD()

日付文字列を数値に変換

書式	DTREAD("日付文字列"), YEAR, MON, DAY	
引数	日付文字列	YYYY/MM/DD 形式の時間文字列
	YEAR	年を受け取る変数
	MON	月を受け取る変数
	DAY	日を受け取る変数
戻り	なし	
エラー		

23 コンソール基本命令

コンソール基本命令

コンソールへの文字表示・変更などを行う以下の命令があります。
CLS、COLOR、LOCATE、PRINT、CHKCHR()、ACLS、VISIBLE

CLS

コンソール画面を消去

書式	CLS
引数	なし
戻り	なし
エラー	

COLOR

コンソールに表示する文字の色指定

書式	COLOR 文字色 [,背景色]	
引数	文字色	0～15(BG用パレット16種類の15番目を利用) 
	背景色	0～15(0=透明)
戻り	なし	
エラー		

LOCATE

コンソール上の文字表示位置を指定

書式	LOCATE x座標,y座標	
引数	x座標	0～31
	y座標	0～23
戻り	なし	
エラー		

PRINT

コンソールへの文字表示

書式	PRINT "文字列" PRINT 変数 PRINT 変数$ PRINT 変数;変数$;”文字列” PRINT "文字列",変数,変数＄	
引数	;	複数の要素を続けて表示する場合に利用
	,	複数の要素を続けて表示するがTAB位置補正がかかる
戻り	なし	
エラー		

CHKCHR()

コンソール上の文字番号を調べる

書式	変数=CHKCHR(x座標,y座標)	
引数	x座標	0～31
	y座標	0～23
戻り	数値	0～255=文字コード(-1=範囲外)
エラー		

ACLS

描画環境を初期化

書式	ACLS
引数	なし
戻り	なし
エラー	

コンソール・SPRITE・BG・グラフィックの描画状態や色等の設定が、以下のプログラムを実行した状態となります。

```
VISIBLE 1,1,1,1,1,1: ICONCLR
COLOR 0: CLS: GDRAWMD FALSE
FOR P=1 TO 0 STEP -1
 GPAGE P,P,P: GCOLOR 0: GCLS: GPRIO 3
 BGPAGE P: BGOFS 0,0,0: BGOFS 1,0,0
 BGCLR: BGCLIP 0,0,31,23
 SPPAGE P: SPCLR
NEXT
FOR I=0 TO 255
 COLINIT ”BG”, I: COLINIT ”SP”, I
 COLINIT ”GRP”,I
NEXT
```

VISIBLE

画面表示要素の制御

書式	VISIBLE コンソール,パネル,BG0,BG1,SPRITE,グラフィック	
引数	コンソール	0=OFF, 1=ON
	パネル	0=OFF, 1=ON
	BG0	0=OFF, 1=ON
	BG1	0=OFF, 1=ON
	SPRITE	0=OFF, 1=ON
	グラフィック	0=OFF, 1=ON
戻り	なし	
エラー		

24 コンソール入力系命令

コンソール入力系命令

ボタン状態や文字列の取得などを行う以下の命令および関数があります。
INKEY$()、INPUT、LINPUT、BUTTON()、BTRIG()、BREPEAT

INKEY$()

キーボードから1文字取得
TABキーはスペースに変換され、BSキーは取得することができません。これらのキーを利用したい時は、KEYBOARDシステム変数をご利用ください。

書式	変数$=INKEY$()	
引数	なし	
戻り	文字変数	キーボードからの1文字(入力が無い場合"")
エラー		

INPUT

数値または文字列の取得

書式	INPUT ["文字列";] 受け取る変数 INPUT ["文字列";] 受け取る変数$ INPUT ["文字列";] 受け取る変数, 受け取る変数2$	
引数	文字列	入力用の説明
	受け取る変数	キーボードからの入力を受け取るための、数値または文字列変数(','で区切れば複数指定可能)
戻り	なし	
エラー		

LINPUT

文字列の取得
INPUTでは入力できない ',' 等も受け付ける。

書式	LINPUT ["文字列";] 受け取る変数$	
引数	文字列	入力用の説明
	受け取る変数	キーボードから1行分の文字列を受け取る変数
戻り	なし	
エラー		

BUTTON()

各ボタンの押下状態取得

ビット単位でボタンの同時押し状態を取得します。例えば、上と右が同時に押された場合、9が返ります。

書式	変数=BUTTON([種別])
引数	種別 (省略時0相当) 0 押されている 1 押した瞬間(連射機能付き) 2 押した瞬間 3 放された瞬間
戻り	ボタンに対応するビット数値 1 十字ボタンの上 2 十字ボタンの下 4 十字ボタンの左 8 十字ボタンの右 16 Aボタン 32 Bボタン 64 Xボタン 128 Yボタン 256 Lボタン 512 Rボタン 1024 スタートボタン
エラー	

BTRIG()

ボタンを押した瞬間の状態を取得

書式	変数=BTRIG()	
引数	なし	
戻り	変数	ボタンに対応するビット数値 ※BUTTON()参照
エラー		

BREPEAT

ボタン連射情報の設定

ボタンの連射機能は通常OFF状態です。有効にする場合は、この命令を利用してください。ボタンIDのみを指定した場合、対象となるボタンの連射機能はOFFとなります。なお、時間単位は1=1/60秒です。

<table>
<tr><td>書式</td><td colspan="2">BREPEAT ボタンID [,開始時間, インターバル]</td></tr>
<tr><td>引数</td><td colspan="2">ボタンID(管理番号)<table><tr><td>0</td><td>十字ボタンの上</td></tr><tr><td>1</td><td>十字ボタンの下</td></tr><tr><td>2</td><td>十字ボタンの左</td></tr><tr><td>3</td><td>十字ボタンの右</td></tr><tr><td>4</td><td>Aボタン</td></tr><tr><td>5</td><td>Bボタン</td></tr><tr><td>6</td><td>Xボタン</td></tr><tr><td>7</td><td>Yボタン</td></tr><tr><td>8</td><td>Lボタン</td></tr><tr><td>9</td><td>Rボタン</td></tr><tr><td>10</td><td>スタートボタン</td></tr></table></td></tr>
<tr><td></td><td>開始時間</td><td>0～</td></tr>
<tr><td></td><td>インターバル</td><td>1～(0=停止)</td></tr>
<tr><td>戻り</td><td colspan="2">なし</td></tr>
<tr><td>エラー</td><td colspan="2"></td></tr>
</table>

25 パネル・アイコン命令

パネル・アイコン命令

パネルの種類変更やユーザー用システムアイコンの表示設定、状態確認などを行う以下の命令があります。
PNLTYPE、PNLSTR、ICONSET、ICONCLR、ICONCHK()

PNLTYPE

パネルの種類変更

書式	PNLTYPE "パネル名"
引数	パネル名 下画面に表示する種類を決める文字列 OFF パネルがない状態 PNL キーボードがない状態 KYA 英語のキーボード KYM 記号のキーボード KYK カナのキーボード
戻り	なし
エラー	

PNLSTR

下画面への文字列表示
上画面のコンソールのように最終行に表示しても自動的に改行は発生しません。

書式	PNLSTR x座標,y座標, "文字列" [, パレット番号]	
引数	x座標	0～31
	y座標	0～23
	文字列	表示させたい文字列
	パレット番号	0～15
戻り	なし	
エラー		

ICONSET

ユーザー用システムアイコンの表示キャラクタ設定(および表示開始)

書式	ICONSET アイコン位置, アイコン番号	
引数	アイコン位置	0～3
	アイコン番号	0～63
戻り	なし	
エラー		

ICONCLR

ユーザー用システムアイコンの表示解除

書式	ICONCLR [アイコン位置]	
引数	アイコン位置	0～3(省略時すべてのアイコン)
戻り	なし	
エラー		

ICONCHK()

ユーザ用システムアイコンの状態確認

書式	数値=ICONCHK()	
引数	なし	
戻り	数値	0～3(アイコン位置),-1=押されていない
エラー		

26 ファイル・通信命令

ファイル・通信命令

ファイルの読み込み、保存、送受信等を行う以下の命令があります。これらの命令を実行すると確認用のダイアログが表示されます。
LOAD、SAVE、DELETE、RENAME、RECVFILE、SENDFILE

LOAD

ファイルの読み込み

<table>
<tr><td>書式</td><td colspan="2">LOAD"リソース名:ファイル名" [,表示制御]</td></tr>
<tr><td>引数</td><td colspan="2">リソース名
読み込む対象となるリソースに割り当てられた文字列
<table><tr><td>PRG</td><td>プログラム(省略可能)</td></tr><tr><td>MEM</td><td>メモリー</td></tr><tr><td>COL0
COL1
COL2</td><td>BG用の色
SPRITE用の色
グラフィック用の色</td></tr><tr><td>GRP0
:
GRP3</td><td>グラフィック (4ページ分)</td></tr><tr><td>SCU0
SCU1</td><td>BGスクリーン手前
BGスクリーン奥</td></tr><tr><td>BGU0
:
BGU3</td><td>ユーザー用BGキャラ
(4バンク分)</td></tr><tr><td>SPU0
:
SPU7</td><td>ユーザー用SPRITEキャラ
(8バンク分)</td></tr><tr><td>SPS</td><td>下画面用SPRITEキャラ</td></tr><tr><td>BGF</td><td>フォント用BGキャラ</td></tr></table></td></tr>
<tr><td></td><td>表示制御</td><td>FALSE を入れると読み込み時のダイアログ表示を省略</td></tr>
<tr><td>戻り</td><td colspan="2">なし</td></tr>
<tr><td>エラー</td><td>RESULT</td><td>FALSE=失敗
TRUE=成功
CANCEL=中止</td></tr>
</table>

SAVE

ファイルの保存
内蔵されているサンプルで使われているファイル名は利用することができません。

書式	SAVE"リソース名:ファイル名"	
引数	リソース名	※LOADを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

DELETE

ファイルの消去
内蔵されているサンプルファイルは削除できません。

書式	DELETE"ﾌｧｲﾙ種別名:ファイル名"	
引数	ﾌｧｲﾙ種別名	※FILESを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

RENAME

ファイル名を変更
内蔵されているサンプルファイルは変更できません。

書式	RENAME"ファイル種別名:ファイル名", "新しい名前"	
引数	ﾌｧｲﾙ種別名	※FILESを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

RECVFILE

プチコンが保存された他の人のDSi本体からファイルを受信
内蔵されているサンプルファイルと同名のファイルは受信できません。

書式	RECVFILE"ファイル種別名:ファイル名"	
引数	ﾌｧｲﾙ種別名	※FILESを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

SENDFILE

プチコンが保存された他の人のDSi本体へファイルを送信
内蔵されているサンプルファイルは送信できません。

書式	SENDFILE "ファイル種別名:ファイル名"	
引数	ﾌｧｲﾙ種別名	※FILESを参照
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

27 ファイル命令(上級者)

ファイル命令(上級者)

ファイルやリソースについて理解している上級者向けの命令。説明を読んで内容が理解できない場合は、使用しないでください。
APPEND、EXEC、SAVE(パッケージ型保存)

APPEND

プログラムの結合(実行モード専用)
編集中のプログラムの最後尾に他のプログラムを結合する。理解しないままAPPEND命令を実行すると、編集中のプログラムの後ろに他のプログラムが結合して作成したプログラムが動かなくなる可能性があります。十分に動作を理解した上でご利用ください。

書式	APPEND "ファイル名"	
引数	ファイル名	結合するプログラムファイル名
戻り	なし	
エラー	RESULT	TRUE=成功 FALSE=失敗

EXEC

プログラム内から他のプログラムを読み込んで実行

書式	EXEC "ファイル名"	
引数	ファイル名	実行するプログラムファイル名
戻り	なし	
エラー	RESULT	TRUE=成功 FALSE=失敗

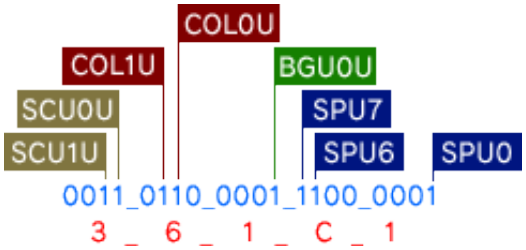
SAVE(パッケージ型)

プログラムとリソースをまとめて保存
パッケージ型で出力したファイルは極端に大きなサイズになる可能性があります。パッケージ型SAVEを利用する際には十分に空きがあることをご確認の上ご利用ください。なお、パッケージ型で保存されたファイルはLOAD命令で自動的に認識し含まれるリソースも読み込まれます。

書式	SAVE"リソース名:ファイル名", "パッケージパラメータ文字列"	
引数	パッケージパラメータ文字列	同時に保存するリソースを指定するための情報
戻り	なし	
エラー	RESULT	FALSE=失敗 TRUE=成功 CANCEL=中止

パッケージパラメータ文字列

リソースの種類を選択するためのビット単位の情報を16進文字列として表現
同時に保存したいリソースに対応するビットを1にすることで、1つのファイルにリソースを含めて保存することが可能。
(例) 以下のリソースを含めたSAVE
・上画面のユーザーSPRITE0,6,7と色
・上画面のユーザーBG0とBGスクリーン2枚と色
上記リソース構成を2進数及び16進では、



パッケージパラメータとして16進文字列でSAVE。
SAVE"TEST","361C1"
なお、HEX$を利用する場合、最大20ビット分の値までしか対応できませんのでご注意ください。

b00	SPU0 上画面用ユーザーSPRITEキャラ0
b01	SPU1 上画面用ユーザーSPRITEキャラ1
b02	SPU2 上画面用ユーザーSPRITEキャラ2
b03	SPU3 上画面用ユーザーSPRITEキャラ3
b04	SPU4 上画面用ユーザーSPRITEキャラ4
b05	SPU5 上画面用ユーザーSPRITEキャラ5
b06	SPU6 上画面用ユーザーSPRITEキャラ6
b07	SPU7 上画面用ユーザーSPRITEキャラ7
b08	BGU0U 上画面用ユーザーBGキャラ0
b09	BGU1U 上画面用ユーザーBGキャラ1
b10	BGU2U 上画面用ユーザーBGキャラ2
b11	BGU3U 上画面用ユーザーBGキャラ3
b12	BGUFU 上画面用フォント
b13	COL0U 上画面用BG用の色
b14	COL1U 上画面用SPRITE用の色
b15	COL2U 上画面用グラフィック用の色
b16	SCU0U 上画面用BGスクリーン手前
b17	SCU1U 上画面用BGスクリーン奥
b18	GRP0 グラフィックページ0
b19	GRP1 グラフィックページ1
b20	GRP2 グラフィックページ2
b21	GRP3 グラフィックページ3
b22	MEM メモリー文字列
b23	システム予約
b24	システム予約

b25	システム予約
b26	システム予約
b27	BGU0L 下画面用ユーザーBGキャラ0
b28	BGU1L 下画面用ユーザーBGキャラ1
b29	BGU2L 下画面用ユーザーBGキャラ2
b30	BGU3L 下画面用ユーザーBGキャラ3
b31	BGUFL 下画面用フォント
b32	COL0L 下画面用BG用の色
b33	COL1L 下画面用SPRITE用の色
b34	COL2L 下画面用グラフィック用の色
b35	SCU0L 下画面用BGスクリーン手前
b36	SCU1L 下画面用BGスクリーン奥
b37	システム予約
b38	システム予約
b39	システム予約
b40	システム予約
b41	システム予約
b42	システム予約
b43	システム予約
b44	システム予約
b45	未使用
b46	未使用
b47	未使用

28 数学基本関数

数学基本関数

整数部の取得、乱数、絶対値や符号の取得など、数学に関する以下の関数があります。
FLOOR()、RND()、ABS()、SGN()

FLOOR()

整数部を取り出す。
1バイト範囲の整数取出しには、AND 命令も利用可能です。（例）A=A AND &HFF

書式	変数=FLOOR(数値)	
引数	数値	元になる数値
戻り	数値	求めた結果
エラー		

RND()

指定値までの乱数を得る。

書式	変数=RND(最大値)	
引数	最大値	生成したい数値の最大値
戻り	数値	0～最大値までの乱数(最大値は含まれない)
エラー		

ABS()

絶対値を得る。

書式	変数=ABS(数値)	
引数	数値	元になる数値
戻り	数値	絶対値
エラー		

SGN()

符号取得。

書式	変数=SGN(数値)	
引数	数値	元になる数値
戻り	数値	0または、±1
エラー		

29 数学指数対数関数

数学指数対数関数

指数や対数に関する以下の関数があります。
SQR()、EXP()、LOG()、POW()

SQR()

平方根値を返す

書式	変数=SQR(数値)	
引数	数値	元になる数値
戻り	数値	求めた結果
エラー	数値が負数の時	

EXP()

指数値を求める

書式	変数=EXP(数値)	
引数	数値	元になる数値
戻り	数値	求めた結果
エラー		

LOG()

自然対数を求める

書式	変数=LOG(数値)	
引数	数値	元になる数値
戻り	数値	求めた結果
エラー		

POW()

べき乗を求める

書式	変数=POW(数値,べき乗数値)	
引数	数値	元になる数値
	べき乗数値	べき乗する値
戻り	数値	求めた結果
エラー	数値が負数でべき乗数値が整数では無い時	

30 数学三角関数

数学三角関数

サイン値、コサイン値の取得など、数学に関する以下の関数があります。
PI()、RAD()、DEG()、SIN()、COS()、TAN()、ATAN()

PI()

パイを返す

書式	変数=PI()	
引数	なし	
戻り	数値	円周率の値
エラー		

RAD()

角度情報からラジアン値を得る

書式	変数=RAD(角度)	
引数	角度	0～360
戻り	数値	角度から求めたラジアン数値
エラー		

角度とラジアン値の関係は以下の通り

角度	ラジアン値
0度	0
90度	PI() x 0.5
180度	PI()
360度	PI() x 2.0

DEG()

ラジアン値から角度情報を得る

書式	変数=DEG(ラジアン値)	
引数	ラジアン値	0～2π
戻り	数値	ラジアン値から求めた角度
エラー		

SIN()

サイン値を返す

書式	変数=SIN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

COS()

コサイン値を返す

書式	変数=COS(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

TAN()

タンジェント値を返す

書式	変数=TAN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

ATAN()

アークタンジェント値を返す

書式	変数=ATAN(ラジアン値)	
引数	ラジアン値	角度のラジアン値
戻り	数値	求めた結果
エラー		

引数を2つ（ Y, X ）渡すと移動量から方向を求める関数としても利用可能です。

書式	変数=ATAN(目的y-y,目的x-x)	
引数	目的y-y	目的yとの差
	目的x-x	目的xとの差
戻り	数値	確度のラジアン値
エラー		

31 文字基本関数

文字基本関数

数値から文字列を生成したり、文字番号を取得するなど、文字に関する以下の命令や関数があります。
ASC()、CHR$()、VAL()、STR$()、HEX$()

ASC()

指定された文字のASCIIコードを返す

書式	変数=ASC(文字)	
引数	文字	文字1つ
戻り	数値	指定された文字の文字コード
エラー		

CHR$()

指定された文字コードから文字を返す

書式	変数$=CHR$(文字コード)	
引数	文字コード	文字ごとに対応する番号
戻り	文字	文字コードに対応する文字
エラー		

VAL()

文字列から数値を得る

書式	変数=VAL(文字列)	
引数	文字列	数値の入った文字列
戻り	数値	文字列から取出した数値
エラー		

STR$()

数値から文字列を得る

書式	変数$=STR$(数値)	
引数	数値	文字列にしたい数値
戻り	文字	数値から生成した文字列
エラー		

HEX$()

数値から16進文字列を得る

書式	変数$=HEX$(数値 [,桁])	
引数	数値	16進文字列にしたい数値
	桁	1～5(桁数に満たない場合0で埋める)
戻り	文字	数値から生成した16進文字列
エラー		

32 文字検索置換関数

文字検索置換関数

文字数の取得や文字列から一部分を取出す等の以下の命令や関数があります。
LEN、MID$、RIGHT$、LEFT$、INSTR()、SUBST$

LEN()

文字列内の文字数を得る

書式	変数=LEN(文字列)	
引数	文字列	文字数を調べたい文字列
戻り	数値	文字の数(全ての文字を1文字として数える)
エラー		

MID$()

文字列の指定位置から指定数分の取り出し

書式	変数$=MID$(文字列, 開始位置, 文字数)	
引数	文字列	元になる文字列
	開始位置	0～文字単位の開始位置
	文字数	取出す文字数
戻り	文字	切り出した文字列
エラー		

RIGHT$()

文字列右端から指定数分の文字列取り出し

書式	変数$=RIGHT$(文字列, 文字数)	
引数	文字列	元になる文字列
	文字数	取出す文字数
戻り	文字	切り出した文字列
エラー		

LEFT$()

文字列左端から指定数分の文字列取り出し

書式	変数$=LEFT$(文字列, 文字数)	
引数	文字列	元になる文字列
	文字数	取出す文字数
戻り	文字	切り出した文字列
エラー		

INSTR

文字列内に検索対象文字列を探す

書式	変数=INSTR([開始位置,] 文字列, 検索文字列)	
引数	開始位置	0～文字単位の開始位置
	文字列	元になる文字列
	検索文字列	探す文字列
戻り	数値	0～文字列内の位置(-1=なし)
エラー		

SUBST$

文字列の置換

書式	変数$=SUBST$(文字列, 開始位置, 文字数, 置換文字列)	
引数	文字列	元になる文字列
	開始位置	0～元になる文字列の置換開始位置
	文字数	置換文字数
	置換文字列	置換する文字列
戻り	文字	置換後の文字列
エラー		

33 グラフィック基本命令

グラフィック基本命令

操作対象グラフィック画面の指定、消去などを行う以下の命令があります。 (色を省略した場合GCOLORで指定した色が使われます)
GPAGE、GCOLOR、GCLS、GSPOIT()

GPAGE

操作対象グラフィック画面の指定

書式	GPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

GCOLOR

グラフィック命令の省略時描画色を指定

書式	GCOLOR 色番号	
引数	色番号	0～255
戻り	なし	
エラー		

GCLS

操作対象グラフィック画面の消去

書式	GCLS [色]	
引数	色	0～255
戻り	なし	
エラー		

GSPOIT()

指定位置の色を調べる

書式	変数=GSPOIT(x座標, y座標)	
引数	x座標	0～255
	y座標	0～191
戻り	色	0～255(範囲外の場合-1)
エラー		

34 グラフィック描画命令

グラフィック描画命令

線や円の描画、塗りつぶし等を行う以下の命令があります (色を省略した場合GCOLORで指定した色が使われます)
GPSET、GPAINT、GLINE、GBOX、GFILL、GCIRCLE

GPSET

点を打つ

書式	GPSET x座標, y座標 [,色]	
引数	x座標	0～255
	y座標	0～191
	色	0～255
戻り	なし	
エラー		

GPAINT

指定位置から塗りつぶし
境界色を指定した場合、境界色に囲まれた範囲を塗りつぶします。省略時は指定位置の色に隣接する同一色が対象。XOR表示モードには対応しません。

書式	GPAINT x座標, y座標 [,色 [, 境界色]]	
引数	x座標	0～255
	y座標	0～191
	色	0～255
	境界色	0～255
戻り	なし	
エラー		

GLINE

線を引く

書式	GLINE 始点x, 始点y, 終点x, 終点y [,色]	
引数	始点x	0～255
	始点y	0～191
	終点x	0～255
	終点y	0～191
	色	0～255
戻り	なし	
エラー		

GBOX

箱を描く

書式	GBOX 始点x, 始点y, 終点x, 終点y [,色]	
引数	始点x	0〜255
	始点y	0〜191
	終点x	0〜255
	終点y	0〜191
	色	0〜255
戻り	なし	
エラー		

GFILL

矩形で塗りつぶす

書式	GFILL 始点x, 始点y, 終点x, 終点y [,色]	
引数	始点x	0〜255
	始点y	0〜191
	終点x	0〜255
	終点y	0〜191
	色	0〜255
戻り	なし	
エラー		

GCIRCLE

円を描く

書式	GCIRCLE x座標, y座標, 半径 [,色 [, 開始角, 終了角]]	
引数	x座標	0〜255
	y座標	0〜191
	半径	0〜255
	色	0〜255
	開始角	0〜360
	終了角	0〜360
戻り	なし	
エラー		

35 グラフィック命令(上級者)

グラフィック命令(上級者)

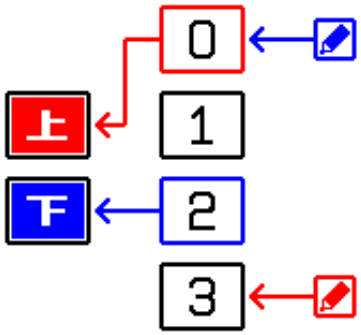
特殊なページ指定、画面のコピー、グラフィック画面へのキャラ表示等に関する以下の命令があります。
GPAGE、GDRAWMD、GPRIO、GCOPY、GPUTCHR

GPAGE(上級者)

操作対象グラフィック画面と描画用と表示用に割り当てる内部画像領域の指定
初期状態及びACLS実行後は、GPAGE 0,0,0 及び GPAGE 1,1,1 が割り当てられています。

書式	GPAGE 画面 [,描画ページ ,表示ページ]	
引数	画面	0=上画面 1=下画面
	描画ページ	0～3
	表示ページ	0～3
戻り	なし	
エラー		

上下2画面に4枚のページから好みの1枚を割り当てられます。上画面・下画面のどちらを操作対象とするかを決めて、対象画面に表示させるページを割り当てます。GPAINT等の描画命令は、描画ページとして割り当てられているページへ描画を行うため画面に表示されていないページで画面を準備してから表示を切り替えることも可能です。



GPRIO

グラフィック画面の表示優先順位の変更
表示優先順位は、常にスプライトに対して後ろ側となり、優先順位の数値はスプライトの仕様に準拠。

書式	GPRIO 番号	
引数	番号	0～3
戻り	なし	
エラー		

GDRAWMD

描画色のXOR表示モードの指定

XOR表示を行うことで、同じ場所に2回描画すると描画内容を消すことができます。

書式	GDRAWMD 状態	
引数	状態	FALSE=通常色 TRUE=XOR
戻り	なし	
エラー		

GCOPY

グラフィック画面のコピー

書式	GCOPY 転送元ページ, 始点x, 始点y, 終点x, 終点y, 転送先座標x, 転送先座標y, ｺﾋﾟｰﾓｰﾄﾞ	
引数	転送元ページ	0～3
	始点x	0～255
	始点y	0～191
	終点x	0～255
	終点y	0～191
	転送先座標x	0～255
	転送先座標y	0～191
	ｺﾋﾟｰﾓｰﾄﾞ	色0番のコピー FALSE=しない TRUE=する
戻り	なし	
エラー		

GPUTCHR

指定されたキャラクタの画像情報をグラフィック画面に表示

この命令を利用すると指定したパレット番号の情報がグラフィック用のパレットにコピーされます。定義される場所は、パレット番号 x 16番目の色からの16色分です。

書式	GPUTCHR x座標, y座標, "キャラ名", キャラ番号, パレット番号, スケール	
引数	x座標	0～255
	y座標	0～191
	キャラ名	※CHRINIT参照
	キャラ番号	0～255
	パレット番号	0～15
	スケール	1,2,4,8
戻り	なし	
エラー		

36 カラー・キャラ命令

カラー・キャラ命令

画面表示要素の制御、色指定および情報取得、キャラクタの定義や情報取得、初期化を行う以下の命令があります。
COLINIT、COLSET、COLREAD()、CHRINIT、CHRSET、CHRREAD()

COLINIT

色を初期状態に戻す

書式	COLINIT "色バンク名", 色番号	
引数	色バンク名 対象を指定する文字列 BG BGスクリーン用 SP SPRITE用 GRP グラフィック用	
	色番号	0～255
戻り	なし	
エラー		

COLSET

色を変更
BGの0番は、背景色として扱われます。

書式	COLSET "色バンク名", 色番号, "色情報文字列"	
引数	色バンク名	※COLINITを参照
	色番号	※COLINITを参照
	色情報文字列	16進文字列 (RRGGBBの順) 各要素00～FF (例) "FF00AA"
戻り	なし	
エラー		

COLREAD()

指定した色の情報取得(各要素0～255)

書式	COLREAD("色バンク名", 色番号), R, G, B	
引数	色バンク名	※COLINITを参照
	色番号	※COLINITを参照
	R	赤の色要素を受け取る変数
	G	緑の色要素を受け取る変数
	B	青の色要素を受け取る変数
戻り	なし	
エラー		

CHRINIT

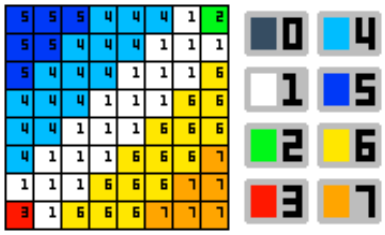
指定したキャラクタを初期状態に戻す

書式	CHRINIT "キャラ名"	
引数	キャラ名 対象キャラクタを指定するための文字列	
	BGU0 : BGU3	ユーザー用BG用キャラ
	SPU0 : SPU7	ユーザー用SPRITE用キャラ
戻り	なし	
エラー		

CHRSET

キャラクタを1つ定義(8x8ドット単位)

書式	CHRSET "キャラ名", キャラ番号, "画像文字列"	
引数	キャラ名	※CHRINIT参照
	キャラ番号	0～255
	画像文字列	16色8x8ドット分のキャラクタ情報を16進文字列とする
戻り	なし	
エラー		



(例) 上記のキャラから16進文字列生成
"5554441255444111544411164441116644111666411166671116667731666777" (1文字1ドット)

CHRREAD()

指定キャラクタの定義情報を取得

書式	CHRREAD("キャラ名", キャラ番号), C$	
引数	キャラ名	※CHRINIT参照
	キャラ番号	※CHRSET参照
	C$	画像文字列を受け取る変数 ※CHRSET参照
戻り	なし	
エラー		

37 SPRITE基本命令

SPRITE基本命令

SPRITEの活動開始や停止などを行う以下の命令があります。
SPPAGE、SPSET、SPCLR、SPHOME

SPPAGE

操作対象SPRITE画面の指定
下画面を選択できますが、下画面は基本的にキーボード用として使われるエリアであり、ユーザー用に用意されたキャラクタを表示させることはできません。あらかじめ用意された簡易の画像表示に利用します。

書式	SPPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

SPSET

SPRITEの定義(活動開始)

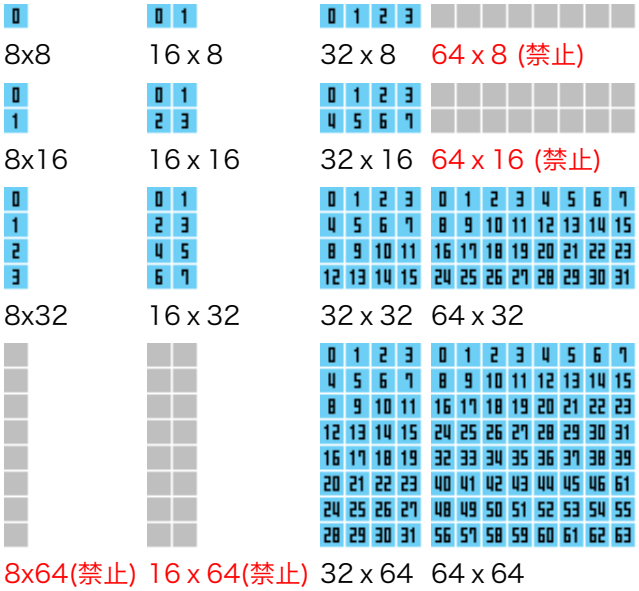
指定管理番号のスプライトが活動を開始します。座標は0,0へ初期化されます。SPSETによって活動を開始した後にキャラ番号だけを変更したい場合は、SPCHR命令を利用します。

<table>
<tr><td>書式</td><td colspan="2">SPSET 管理番号,キャラ番号, パレット番号, 横反転, 縦反転, 優先順位 [,幅, 高さ]</td></tr>
<tr><td rowspan="7">引数</td><td>管理番号</td><td>0～99</td></tr>
<tr><td>SPRITEキャラ番号</td><td>0～511
(下画面の場合0～117)</td></tr>
<tr><td>パレット番号</td><td>0～15</td></tr>
<tr><td>横反転</td><td>0=OFF　　1=ON
  </td></tr>
<tr><td>縦反転</td><td>0=OFF　　1=ON
  </td></tr>
<tr><td>優先順位</td><td><table><tr><td>0</td><td>コンソールの前</td></tr><tr><td>1</td><td>手前のBGの前</td></tr><tr><td>2</td><td>2枚のBGの間</td></tr><tr><td>3</td><td>奥のBGの後</td></tr></table>SPRITE間の表示優先順位は、管理番号が小さい物が手前に表示されます。</td></tr>
<tr><td>幅</td><td>8,16,32,64
(省略時16)</td></tr>
<tr><td>高さ</td><td>8,16,32,64
(省略時16)</td></tr>
<tr><td>戻り</td><td colspan="2">なし</td></tr>
<tr><td>エラー</td><td colspan="2"></td></tr>
</table>

幅と高さには、8x64、16x64、64x8、64x16の組合せは指定できません。

SPRITEキャラの格納ルール

SPSET命令によるサイズ指定を行った場合、サイズと表示されるキャラ番号の関係は以下のようになります。通常定義されているキャラは16x16単位で登録されているためサイズの異なるキャラを使う場合は、キャラの配置も調整が必要になります。



SPCLR

SPRITEの消去(活動禁止にする)

書式	SPCLR [管理番号]	
引数	管理番号	0～99(省略時すべてのSPRITEを消去)
戻り	なし	
エラー		

SPHOME

SPRITEの表示原点の指定

この命令を省略した場合、表示原点は左上(0,0)になります。

書式	SPHOME 管理番号 ,x ,y	
引数	管理番号	0～99
	x	0～63
	y	0～63
戻り	なし	
エラー		

38 SPRITE制御命令

SPRITE制御命令

SPRITEの座標変更やアニメ表示の指定などを行う以下の命令があります。
SPOFS、SPCHR、SPANIM、SPANGLE、SPSCALE

SPOFS

SPRITE座標の変更

書式	SPOFS 管理番号, x座標, y座標 [,補間時間]	
引数	管理番号	0～99
	x座標	±1024
	y座標	±1024
	補間時間	現在の状態と新しい値の間を自動補間する時間 (1=1/60秒)
戻り	なし	
エラー		

SPCHR

SPRITEのキャラ番号の変更

書式	SPCHR 管理番号,キャラ番号 [, パレット番号, 横反転, 縦反転, 優先順位]	
引数	管理番号	0～99
	SPRITEキャラ番号	0～511(下画面への表示の場合0～117)
	パレット番号	0～15
	横反転	0=なし,1=反転
	縦反転	0=なし,1=反転
	優先順位	0～3
戻り	なし	
エラー		

SPANIM

SPRITEによるアニメ表示

現在指定されているキャラ番号から、この命令で指定された枚数の範囲で指定時間ごとにキャラ番号を切り替える。

書式	SPANIM 管理番号, 枚数, 時間 [,ループ]	
引数	管理番号	0～99
	枚数	1～
	時間	1コマを表示する時間(1=1/60秒)
	ループ	0=無限ループ 1～ループ回数
戻り	なし	
エラー		

SPANGLE

SPRITE角度の変更

初期状態での回転は左上を原点として実行されます。回転中心を変更したい場合は、SPHOME命令をご利用ください。

書式	SPANGLE 管理番号, 角度 [,補間時間, 変化方向]	
引数	管理番号	0～31
	角度	0～360
	補間時間	現在の状態と新しい値の間を自動補間する時間 (1=1/60秒)
	変化方向	1=時計回り -1=反時計回り (省略時時計回り)
戻り	なし	
エラー		

SPSCALE

SPRITEスケールの変更

スケールを200%にした状態でSPANGLEによる回転を行うと、回転の結果200%相当の矩形範囲からはみ出た部分は表示が切られてしまいます。

書式	SPSCALE 管理番号, スケール [, 補間時間]	
引数	管理番号	0～31
	スケール	0～200(パーセント相当)
	補間時間	現在の状態と新しい値の間を自動補間する時間 (1=1/60秒)
戻り	なし	
エラー		

39 SPRITE情報取得命令

SPRITE情報取得命令

SPRITEの様々な情報を取得する以下の命令があります。
SPCHK()、SPREAD()、SPSETV、SPGETV()

SPCHK()

自動補間の状態取得

書式	変数 = SPCHK(管理番号)	
引数	管理番号	0～99
戻り	数値	0の時補間終了 b00 位置 b01 角度 b02 スケール b03 アニメ再生
エラー		

SPREAD()

SPRITE情報の読込み
補間動作で移動中の座標や角度等の情報を取得できます。

書式	SPREAD(管理番号) ,X ,Y [,A ,S ,C]	
引数	管理番号	0～99
	X	X座標を受け取る変数
	Y	Y座標を受け取る変数
	A	角度を受け取る変数
	S	スケールを受け取る変数
	C	キャラ情報を受け取る変数
戻り	なし	
エラー		

SPSETV

各SPRITEごとに用意されたユーザー用変数への書き込み
この変数はユーザーが自由に利用することができます。例えば、体力や攻撃力や防御力等の情報を配列を使わずにSPRITEごとに保持する等の使い方を想定しています。最初に初期値を代入して利用して下さい。

書式	SPSETV 管理番号, 変数番号, 値	
引数	管理番号	0～99
	変数番号	0～7
	値	数値
戻り	なし	
エラー		

SPGETV()

各SPRITEごとに用意されたユーザー用変数からの読込み

書式	変数=SPGETV(管理番号, 変数番号)	
引数	管理番号	0～99
	変数番号	0～7
戻り	数値	変数の内容
エラー		

40 SPRITE衝突判定命令

SPRITE衝突判定命令

SPRITEの当たり判定に関する以下の命令があります。
SPCOL、SPCOLVEC、SPHIT()、SPHITSP()、SPSPHITRC()

SPCOL

SPRITE衝突判定用の矩形の定義
この命令を実行しない状態では、オフセットは0,0、サイズはSPSETで指定されたサイズとなります。

書式	SPCOL 管理番号, x ,y, 横幅, 縦幅, スケール対応 [,グループ]	
引数	管理番号	0～99
	x	±64
	y	±64
	横幅	1～64
	縦幅	1～64
	スケール対応	FALSE=無視 TRUE=同期
	グループ	0～255(ビット単位でグループ設定
戻り	なし	
エラー		

グループには、ビット単位で好みの情報を割り当てることができます。例えば下位4ビットでプレイヤー側の情報、上位4ビットで敵側の情報に分けて、プレイヤーと敵との衝突判定では、グループに&HF0を入れます。

b00	プレイヤー
b01	プレイヤーの弾
b02	エフェクト
b03	アイテム
b04	敵
b05	敵の弾
b06	ボス
b07	背景の障害物

SPCOLVEC

SPRITE衝突判定用の移動速度の定義
強制的に移動量を指定する場合に利用します。解除する場合は、移動量を省略して実行してください。
なお、この命令を実行していない状態では、SPOFSで指定された時間と目的地から求めた値で自動的に計算されます。SPOFSを移動時間指定無しで実行した場合は、移動量は0となります。

書式	SPCOLVEC 管理番号 [,移動量x ,移動量y]	
引数	管理番号	0～99
	移動量x	±16.0
	移動量y	±16.0
戻り	なし	
エラー		

SPHIT()

SPRITEの衝突判定
同じグループ同士での衝突判定を行います。
衝突時、システム変数に相手の情報を残します。

書式	SPHIT(管理番号 [,開始管理番号])	
引数	管理番号	0～99
	開始管理番号	0～99(省略時自分以外すべて)
戻り	結果	TRUE=衝突発生
エラー		

SPHITSP()

SPRITE同士の衝突判定
指定した相手との衝突判定を行います。
衝突時、システム変数に相手の情報を残します。

書式	SPHITSP(管理番号, 相手管理番号)	
引数	管理番号	0～99
	相手管理番号	0～99
戻り	結果	TRUE=衝突発生
エラー		

SPHITRC()

SPRITEと矩形の衝突判定

指定した矩形サイズとの衝突判定を行います。

衝突時、システム変数に相手の情報を残します。

書式	SPHITRC(管理番号, 始点x, 始点y, 横幅, 縦幅 [,移動量x, 移動量y])	
引数	管理番号	0～99
	始点x	±1024
	始点y	±1024
	横幅	1～
	縦幅	1～
	移動量x	±16.0
	移動量 y	±16.0
戻り	結果	TRUE=衝突発生
エラー		

41 BG基本命令

BG基本命令

操作対象BG画面の指定、表示オフセット変更、BGスクリーンへの書き込みなどを行う以下の命令があります。
BGPAGE、BGCLR、BGCLIP、BGOFS、BGPUT、BGFILL、BGREAD()

BGPAGE

操作対象BG画面の指定

書式	BGPAGE 画面	
引数	画面	0=上画面 1=下画面
戻り	なし	
エラー		

BGCLR

BG画面の消去(キャラ0番で埋め尽くす)

書式	BGCLR [レイヤー]	
引数	レイヤー	0=手前,1=奥 (省略時両方)
戻り	なし	
エラー		

BGCLIP

表示範囲の指定(すべてのレイヤーが対象)

書式	BGCLIP 始点x, 始点y, 終点x, 終点y	
引数	始点x	0〜31
	始点y	0〜23
	終点x	0〜31
	終点y	0〜23
戻り	なし	
エラー		

BGOFS

BGスクリーンの表示オフセットを変更

書式	BGOFS レイヤー, x座標, y座標 [,補間時間]	
引数	レイヤー	0=手前,1=奥
	x座標	0〜511
	y座標	0〜511
	補間時間	現在の状態と新しい値の間を自動補間する時間 (1=1/60秒)
戻り	なし	
エラー		

BGPUT

指定位置のBGスクリーンに書き込み

書式	BGPUT レイヤー, x座標, y座標, キャラ番号, パレット番号, 横反転, 縦反転	
引数	レイヤー	0=手前,1=奥
	x座標	0〜63
	y座標	0〜63
	キャラ番号	0〜1023
	パレット番号	0〜15
	横反転	0=なし,1=反転
	縦反転	0=なし,1=反転
戻り	なし	
エラー		

BGFILL

BGスクリーンの矩形範囲塗りつぶし

書式	BGFILL レイヤー, 始点x, 始点y, 終点x, 終点y, キャラ番号, パレット番号, 横反転, 縦反転	
引数	レイヤー	0=手前,1=奥
	始点x	0〜63
	始点y	0〜63
	終点x	0〜63
	終点y	0〜63
	キャラ番号	0〜1023
	パレット番号	0〜15
	横反転	0=なし,1=反転
	縦反転	0=なし,1=反転
戻り	なし	
エラー		

BGREAD()

指定位置のBGスクリーンから情報を取得

書式	BGREAD(レイヤー, x座標, y座標), CHR, PAL, H, V	
引数	レイヤー	0=手前,1=奥
	x座標	0～63
	y座標	0～63
	CHR	キャラ番号を受け取る変数
	PAL	パレット番号を受け取る変数
	H	横反転情報を受け取る変数
	V	縦反転情報を受け取る変数
戻り	なし	
エラー		

42 BG命令(上級者)

BG命令(上級者)

BGの状態取得、特殊な形式での描画や塗りつぶしなどを行う以下の命令があります。
BGCHK()、BGCOPY、BGPUT、BGFILL、BGREAD()

BGCHK()

BGOFSで変化しているBGの状態取得

書式	BGCHK(レイヤー)	
引数	レイヤー	0=手前,1=奥
戻り	数値	0の時補間終了 b00 位置
エラー		

BGCOPY

BGスクリーンの矩形範囲コピー

書式	BGCOPY レイヤー, 始点x, 始点y, 終点x, 終点y, 転送先x, 転送先y	
引数	レイヤー	0=手前,1=奥
	始点x	0～63
	始点y	0～63
	終点x	0～63
	終点y	0～63
	転送先x	0～63
	転送先y	0～63
戻り	なし	
エラー		

スクリーンデータ仕様

◆スクリーンデータ

BGスクリーンのキャラ番号とパレット番号、および横反転や縦反転情報を1つにまとめた形式。16ビット値。

b00	▲
b01	:
b02	:
b03	:
b04	: 10ビットでキャラ番号 (0～1023)
b05	:
b06	:
b07	:
b08	:
b09	▼
b10	横反転 (0=OFF,1=ON)
b11	縦反転 (0=OFF,1=ON)
b12	▲
b13	: 4ビットでパレット番号 (0～15)
b14	:
b15	▼

◆スクリーンデータ文字列

スクリーンデータ形式を4桁の16進文字列に変換した形式。

(例)

スクリーンデータが &H103F の時、スクリーンデータ文字列は、"103F"

BGPUT(上級者用)

◆スクリーンデータを書く

書式	BGPUT レイヤー, x座標, y座標, スクリーンデータ	
引数	レイヤー	0=手前,1=奥
	x座標	0～63
	y座標	0～63
	ｽｸﾘｰﾝﾃﾞｰﾀ	16進4桁の数値
戻り	なし	
エラー		

◆スクリーンデータ文字列を書く

書式	BGPUT レイヤー, x座標, y座標, スクリーンデータ文字列	
引数	レイヤー	0=手前,1=奥
	x座標	0～63
	y座標	0～63
	ｽｸﾘｰﾝﾃﾞｰﾀ文字列	16進4桁文字列
戻り	なし	
エラー		

BGFILL(上級者用)

◆スクリーンデータで矩形塗り

書式	BGFILL レイヤー, 始点x, 始点y, 終点x, 終点y, スクリーンデータ	
引数	レイヤー	0=手前,1=奥
	始点x	0～63
	始点y	0～63
	終点x	0～63
	終点y	0～63
	ｽｸﾘｰﾝﾃﾞｰﾀ	16進4桁の数値
戻り	なし	
エラー		

◆スクリーンデータ文字列で矩形塗り

書式	BGFILL レイヤー, 始点x, 始点y, 終点x, 終点y, スクリーンデータ文字列	
引数	レイヤー	0=手前,1=奥
	始点x	0～63
	始点y	0～63
	終点x	0～63
	終点y	0～63
	ｽｸﾘｰﾝﾃﾞｰﾀ文字列	16進4桁文字列
戻り	なし	
エラー		

BGREAD(上級者用)

◆スクリーンデータを取得

書式	BGREAD(レイヤー, x座標, y座標), SC	
引数	レイヤー	0=手前,1=奥
	x座標	0～63
	y座標	0～63
	SC	ｽｸﾘｰﾝﾃﾞｰﾀを受け取る変数
戻り	なし	
エラー		

◆スクリーンデータ文字列を取得

書式	BGREAD(レイヤー, x座標, y座標), SC$	
引数	レイヤー	0=手前,1=奥
	x座標	0～63
	y座標	0～63
	SC$	ｽｸﾘｰﾝﾃﾞｰﾀ文字列を受け取る変数
戻り	なし	
エラー		

43 サウンド基本命令

サウンド基本命令

効果音とBGMに関する以下の命令があります。MMLを使った作曲に関する命令は上級者用のページをご覧ください。
BEEP、BGMPLAY、BGMSTOP、BGMCHK()、BGMVOL

BEEP

単純な警告音(効果音)の発声

書式	BEEP [波形番号 [,ピッチ [,音量 [,パンポット]]]]	
引数	波形番号	0～69(省略時は0番)
	ピッチ	-8192で2オクターブ下、0で原音、8192で2オクターブ上
	音量	0=無音 127=最大
	パンポット	0=左から 64=中央から 127=右から
戻り	なし	
エラー		

●ピッチと音階の計算方法
1オクターブは、4096段階の分解能を持ち、半音分のピッチPは、P=4096/12で求めることができる。この値を使った音程計算は以下のようになる。
＜音程変化＞
C　=P*0
C＃=P*1
D　=P*2
D＃=P*3
E　=P*4
F　=P*5
F＃=P*6
G　=P*7
G＃=P*8
A　=P*9
A＃=P*10
B　=P*11

BGMPLAY

音楽演奏開始(同時に8曲再生可能)
トラック音量を指定する場合は、トラック番号は省略できません。曲番号128以降は作曲した曲を演奏。

書式	BGMPLAY 曲番号 BGMPLAY トラック番号, 曲番号 [,トラック音量]	
引数	トラック番号	0～7(省略時0)
	曲番号	0～29, 128～255
	トラック音量	0～127
戻り	なし	
エラー		

BGMSTOP

音楽演奏停止
フェード時間（秒単位）を指定することで、演奏中の曲を少しずつ聞こえなくすることができます。なお、トラック番号を省略した場合は、BEEPによる効果音と全てのトラック演奏が停止します。

書式	BGMSTOP [トラック番号 [,フェード時間]]	
引数	トラック番号	0～7
	フェード時間	終了までの時間(0=即時停止)
戻り	なし	
エラー		

BGMCHK()

音楽の演奏状態調査

書式	変数=BGMCHK([トラック番号])	
引数	トラック番号	0～7(省略時0)
戻り	数値	FALSE=停止中 TRUE=演奏中
エラー		

BGMVOL

トラックごとの音量調整

書式	BGMVOL [トラック番号,] 音量	
引数	トラック番号	0～7(省略時0)
	トラック音量	0～127
戻り	なし	
エラー		

44 サウンド命令(上級者)

サウンド命令(上級者)

MMLを使って作曲を行ったりMMLとの情報交換などを行う以下の命令があります。
BGMSET、BGMSETD、BGMCLEAR、BGMPLAY、BGMSETV、BGMGETV()、BGMPRG

BGMSET

MMLによる曲データの登録

書式	BGMSET 曲番号, MML文字列 [,MML文字列2...]	
引数	曲番号	128～255
	MML文字列	演奏情報を文字列として記述
	MML文字列2	1つの文字列で収まらない場合、複数の文字列に分割可能
戻り	なし	
エラー		

BGMSETD

DATAとして用意したMMLの登録
MMLを記述したDATA文の最後に、MMLの終了の印としてDATA 0 が必要です。

書式	BGMSETD 曲番号, @ラベル BGMSETD 曲番号, 変数＄	
引数	曲番号	128～255
	@ラベル	MMLが登録されたDATAを指すラベル名
戻り	なし	
エラー		
例	@MMLDATA DATA "CDEFGAB＜C＞" DATA "CCCEEEGGG" DATA 0 '--- BGMSETD 128, @MMLDATA BGMPLAY 128	

BGMCLEAR

登録された曲データをクリア

書式	BGMCLEAR [曲番号]	
引数	曲番号	128～255(省略時すべての曲データ)
戻り	なし	
エラー		

BGMPLAY(上級者用)

MMLを直接渡して音楽を演奏
MMLによる演奏の場合は、演奏トラックは0に固定されます。トラック音量の指定はできません。（トラック番号１～７を指定するとエラーになります）

書式	BGMPLAY MML文字列 [,MML文字列2...]	
引数	MML文字列	演奏情報を文字列として記述
	MML文字列2	1つの文字列で収まらない場合、複数の文字列に分割可能
戻り	なし	
エラー		

BGMSETV

MML内変数への書込み

書式	BGMSETV トラック番号,変数番号,値	
引数	トラック番号	0～7
	変数番号	0～7(MML変数の$0～$7)
	値	0～255
戻り	なし	
エラー		

BGMGETV()

MML内変数からの読込み
MML内の変数代入が反映される周期は1/60秒ごとです。最低でもVSYNC 1以上の間を空けてください。

書式	変数=BGMGETV(トラック番号, 変数番号)	
引数	トラック番号	0～7
	変数番号	0～7(MML変数の$0～$7)
戻り	数値	-1=演奏停止中
エラー		

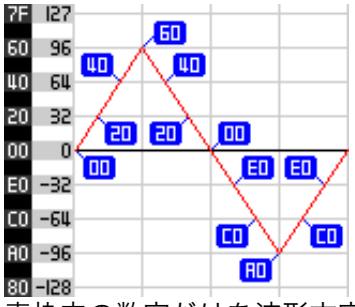
BGMPRG

MMLで演奏できる楽器として波形を登録

波形に対するエンベロープを設定するA,D,S,Rについては、MMLコマンド@Eもご覧ください。

書式	BGMPRG 楽器番号, A ,D ,S ,R ,波形文字列	
引数	楽器番号	224～255
	A 立ち上がり	0～127
	D 減衰	0～127
	S 維持	0～127
	R 消えるまで	0～127
	波形文字列	16進文字列(64文字または128文字必要)
戻り	なし	
エラー		

波形文字列は、符号付8ビット数値を2文字の16進数表記にして管理しています。



上図の青枠内の数字だけを波形文字列にすると以下のようになります。

"00204060402000E0C0A0C0E0"

このようにして求めた波形文字列64文字分で1周期の波形として扱われます。128文字分用意すると少し複雑な波形も定義することができます。

生楽器音サンプリングのような複雑な波形は登録できませんが、チープな音が鳴る簡易シンセサイザーとして利用できます。

45 音声合成命令

音声合成命令

音声合成に関する以下の命令があります。
TALK、TALKSTOP、TALKCHK()

TALK

入力された音声文字列を発声します
TALK命令を実行すると内部的に音声合成の処理が実行されるため音声文字列が長い場合、発声までに時間がかかることがあります。

書式	TALK 音声文字列 [,音声文字列2 ...]	
引数	音声文字列	カナで書かれた文字列 ※詳細はTALKコマンド参照
	音声文字列2	1つの文字列で収まらない場合、複数の文字列に分割可能
戻り	なし	
エラー		

TALKSTOP

音声合成の再生停止

書式	TALKSTOP
引数	なし
戻り	なし
エラー	

TALKCHK()

音声合成の再生状態調査

書式	変数=TALKCHK()	
引数	なし	
戻り	数値	FALSE=停止中 TRUE=再生中
エラー		