

制御と分岐	
分岐、くりかえしなどの制御命令	
ラベル	[slot]@ラベル文字 ラベル文字列。"1:@label" でスロット指定
@ラベル	プログラムやデータの位置を示す名前
GOTO @ラベル	指定した 1:@ラベル名 の形式でプログラムスロット指定も可能
GOSUB @ラベル	指定した RETURN命令でGOSUBの次の命令に戻ってくる
RETURN	直前に使用したGOSUB命令の次の命令に処理を移す
ON 式 GOTO @ラベル0,@ラベル1...	式の値に対応したラベル行に分岐する / ラベル文字列は使えない
ON 式 GOSUB @ラベル0,@ラベル1...	式の値に対応したラベル行に分岐し、RETURN命令の使用でON GOSUBの次の命令に戻る / ラベル文字列は使えない
ON BREAK GOTO @ラベル	ユーザー 一度登録ラベルにジャンプすると、登録は解除される エラー、TRACE.STOP実行、+ボタン長押しによる強制停止、TOPMENUからのプログラム起動等によりプログラムが停止する場合はラベルを登録していてもジャンプしない
IF THEN ELSE	
IF 式 THEN 成立時処理1 ELSE 不成立時処理2 [ENDIF]	式の評価結果が真(1)の時に処理1を、偽(0)の時に処理2を実行する
THEN 処理	IF文で条件が成立した時の処理の前に書くキーワード
ELSE 処理	IFによる条件が成立しない時の処理の前に書くキーワード
ELSEIF 式 THEN 成立時処理 ENDIF	IFによる条件不成立時にさらに追加条件を指定する場合のキーワード
ENDIF	IFによる制御切替後処理が複数行になる場合にIF文を終了させる
IF 式 GOTO @ラベル [ELSE 不成立時処理]	式が真(1)の時@ラベルに分岐する
CASE WHEN ENDCASE	
CASE 式	CASEによる条件分岐を開始する
WHEN 式	CASE～ENDCASE内で分岐条件を指定する
OTHERWISE	CASE～ENDCASE内でどのWHENにも該当しなかった場合の処理を指定する
ENDCASE	CASEに、CASEで条件分岐を開始した後は必ずENDCASEを使用しなければならない
LOOP ENDLOOP	
LOOP	無限にループから抜けるには、BREAK命令を使用する必要がある
ENDLOOP	LOOP～ENDLOOPループの開始キーワード 対応するLOOPキーワード直後の命令に処理を移す
FOR TO NEXT	
FOR ループ変数=初期値 TO 終了値 [STEP 増分]	処理を指定回数繰り返す
TO 終了値	FOR～NEXTループの終了値を指定する
STEP 加算値	FOR～NEXTループでループ毎に加算する値を定義する
NEXT	FORループの終わりを示す命令
WHILE WEND	
WHILE 式	式の計算結果が偽(0)の場合、続くWENDの次の命令に処理を移す
WEND	直前のWHILE命令に処理を移し、再度WHILEから実行する
REPEAT UNTIL	
REPEAT	REPEAT～UNTILループの開始命令
UNTIL 式	式の評価結果が偽(0)の場合、対応するREPEAT命令に処理を移す
CONTINUE	強制的にLOOP～ENDLOOP、FOR～NEXT、WHILE～WEND、REPEAT～UNTIL内で使用可能
BREAK	ループをLOOP～ENDLOOP、FOR～NEXT、WHILE～WEND、REPEAT～UNTIL内で使用
制御	
END	プログラムを終了する
STOP [表示文字列]	実行中のCONT命令で継続続行可能
DEF	
[COMMON] DEF 定義名 [* 引数[引数...] [OUT { * V1 [V2...]]] / END	ユーザー定義命令
[COMMON] DEF 定義名 ([引数...]) / END	ユーザー定義命令
DEF 定義名	
DEF 定義名 引数[引数...]	
DEF 関数名([引数[引数...]])	
DEF 命令名[引数[引数...]] OUT V1[V2...]	
DEF 命令名 * [OUT *]	可変長の引数や可変長の返値を持つユーザー定義命令の定義
COMMON	
RETURN	スロットを超えて独自命令を定義する場合に利用するキーワード
END	DEFの返値を設定して呼び出し元に復帰する ユーザー関数、ユーザー命令のDEF定義を終了する
DEFARGC()	ユーザー定義命令・関数の引数の個数を調べる
DEFARG(引数番号)	ユーザー定義命令・関数の引数の値を調べる
DEFOUTC()	ユーザー定義命令・関数の返値の個数を調べる
DEFOUT 返値番号,値	ユーザー定義命令・関数の返値を設定する
CALL 命令名文字列 [引数...][OUT 変数1[変数2...]	指定名称 1:ユーザー定義命令名 の形式でプログラムスロット指定も可能
CALL(関数名[引数...])	指定名称を持つ関数を呼び出す
CALL SPRITE	SPFUNC IDはCALLIDX()で取得
CALL TEXT	TFUNCでIDはCALLIDX()で取得

数学

三角関数や対数などの数学計算用関数
整数 = INT(数値)
実数 = FLOAT(数値)
実数 = FLOOR(数値)
実数 = ROUND(数値)
実数 = CEIL(数値)
数値 = ABS(数値)
符号 = SGN(数値)
値 = MIN(数値配列)
値 = MIN(数値,数値[,数値…])
値 = MAX(数値配列)
値 = MAX(数値,数値[,数値…])
整数 = RND([シードID]最大値)
実数 = RND([シードID])
RANDOMIZE シードID[,シード値]
実数 = SQRT(数値)
EXP(数値)
LOG(数値[,底])
POW(数値,乗数)
ラジアン値 = RAD(角度)
度 = DEG(ラジアン値)
三角比 = SIN(ラジアン値)
三角比 = COS(ラジアン値)
三角比 = TAN(ラジアン値)
ラジアン値 = ASIN(三角比)
ラジアン値 = ACOS(三角比)
ラジアン値 = ATAN(三角比)
ラジアン値 = ATAN(座標Y,座標X)
値 = SINH(数値)
値 = COSH(数値)
値 = TANH(数値)
値 = CLASSIFY(数値)

整数型に変換する
実数型に変換する
小数点以下すべて0となるが、実数型となる。
小数部を小数点以下すべて0となるが、実数型となる。
小数点以下すべて0となるが、実数型となる。
絶対値を得る
符号を取 1: 正、0: 0、-1: 負
指定され 配列の型と同じ型が返る？
指定され 配列の型と同じ型が返る？
指定された数値配列内の一番大きい値を得る
指定された複数の数値から一番大きい値を得る
整数の乱シードID は 0～7
実数型のシードID は 0～7
乱数系列シードID は 0～7
正の平方根を求める
e(自然対 指数省略で定数#EXPの値を返す
対数を求める
べき乗を求める
角度の単位を度からラジアンに変換する
角度の単位をラジアンから度に変換する
サイン(正弦)を返す
コサイン(余弦)を返す
タンジェント(正接)を返す
アークサイン(SINの逆関数)を返す
アークコサイン(COSの逆関数)を返す
アークタンジェント(TANの逆関数)を返す(数値から)
アークタンジェント(TANの逆関数)を返す(XY座標から) / 原点からY,Xの値
ハイパボリックサイン(双曲線正接)を返す
ハイパボリックコサイン(双曲線余弦)を返す
ハイパボリックタンジェント(双曲線正接)を返す
通常数値 0=通常数値、1=無限大、2=NaN

文字列操作

文字列操作や変換用関数
文字コード = ASC(文字列)
文字列 = CHR\$(文字コード)
値 = VAL(文字列)
文字列 = STR\$(数値[,桁数])
文字列 = HEX\$(数値[,桁数])
文字列 = BIN\$(数値[,桁数])
文字列 = FORMAT\$(書式文字列,値,...)
値 = LEN([文字列 | 配列])
値 = LAST([文字列 | 配列])
文字列 = MID\$(文字列,開始位置,文字数)
文字列 = LEFT\$(文字列,文字数)
文字列 = RIGHT\$(文字列,文字数)
文字列 = INSTR([開始位置,]元文字列,検索する文字列)
文字列 = SUBST\$(元文字列,開始位置[,文字数,]置換文字列)
文字列 = DATE\$()
文字列 = TIME\$()

指定された文字列の先頭文字コードを取得する
指定された文字コードから文字を返す
文字列から数値を得る
数値を文字列に変換する
数値を16進数文字列に変換する
数値を2進数文字列に変換する
表示書式を使って値を整形し文字列化する
文字列の文字数、または配列の要素数を得る / 文字列
文字列または配列の末尾の位置を返す / LEN関数の返値から1を引いた値
文字列の指定位置から指定数分の文字列を得る
文字列の先頭から指定数分の文字列を得る
文字列の末端から指定数分の文字列を得る
文字列内から対象文字列を検索する / 検索結果の文字列が返る
文字列の一部を置き換える / 置き換えた文字列が返る
現在の日付を文字列として返す / “西暦/月/日”
現在の時刻を文字列として返す / “時:分:秒”

その他

その他分類分けできない各種命令

CONST #定数名=定数式[#定数名=定数式]
ENUM #定数名[-定数式][#定数名=-定数式]

DATA 定数式[定数式…]
READ 変数1[変数2…]
RESTORE [@ラベル | 文字列]

OPTION 機能名

WAIT [frame数=1]
VSYNC [frame数=1]

REM
,

TMREAD [時間文字列] OUT H.M.S
DTREAD [日付文字列] OUT Y.M.D[W]
値 = CHKLABEL[ラベル文字列 [フラグ=0]]

値 = CHKCALL(命令名文字列)
値 = CHKVAR(文字列)

DIALOG “表示文字列”[選択タイプ][キャプション文字列][タイムアウト時間]]

値 = DIALOG(“表示文字列”,ボタン種類[“キャプション文字列”[タイムアウト時間]] / ダイアログを表示しタッチやハードウェアボタンが押されるまで待つ

文字列 = DIALOG(“初期文字列”,“キャプション文字列”[最大文字数])

RESULT()

値 = CALLIDX()

CLIPBOARD 文字列
文字列 = CLIPBOARD()

KEY キー番号,“文字列”
文字列 = KEY(ファンクションキー番号)

FONTINFO [文字コード] 文字列,フォント種類 OUT X,Y

値 = FREEMEM()
FREEMEM OUT 総空き容量,連続空き容量

値 = MILLISEC()
値 = MAINCNT()

設定値 = SYSPARAM(設定項目名文字列)
SYSPARAM 設定項目名文字列 OUT 設定値1,設定値2
SYSPARAM 設定項目名文字列,設定値

PERFBEGIN 計測番号[色コード]
PERFEND 計測番号
値 = PERFEND(計測番号)

METALOAD
METAEDIT メタデータ項目,設定データ
METAEDIT メタデータ項目 OUT 設定データ

METASAVE

定数を定義する / 定数式には数値、文字列、他の定数、CHR\$,RGB,DEG,RADの各関数、及びそれらの演算が使用できる
定数を連番で定義する / CONSTと違い、整数定数のみ定義できる

READで読み込むデータの定義
DATA命令で定義した値を読み込む
READ命令で読み込むDATA文を変更する

プログラムの動作モードを設定 / STRICT 変数宣言が必須となる(宣言なし参照はエラー)

frame数経過後まで処理停止 / 必ず停止 ※frame=1/60sec
前回のVSYNC実行時からframe数経過するまで処理停止 / 停止しないこともある

コメント記述用の記号
コメント記述用の記号

時間文字列を数値に変換する / HH:MM:SS形式の時間文字列(省略時、現在の時間)
日付文字列を数値に変換 / YYYY/MM/DD形式の日付文字列(省略時現在の日時) / W曜日(日曜日を0とする数値)
指定文字列で参照できるラベルの存在確認
/ 0: DEF内だけを検索, 1: DEF内に無ければグローバルラベルを検索 / 戻り値 0=存在しない, 1=存在
指定文字列で参照できる命令・関数の存在確認 / 戻り値 0=存在しない, 1=存在
指定文字列で参照できる変数の存在確認 / 戻り値 0=存在しない, 1=存在

ダイアログを表示しボタンが押されるまで待つ / 結果は RESULT関数で取得
表示文字列 256文字まで / CHR\$(10) か CHR\$(13) で改行
選択タイプ 0: 了解(デフォルト), 1: いいえ/はい, 2: 戻る/次へ, 3: 中止/決定, 4: 中止/実行, 5: 次へ
キャプション文字列 46文字まで / 省略すると”DIALOG”
タイムアウト時間 秒数指定、マイナス値で frame数、0 で操作待ち

ボタン種類			
bit0	Xボタン	bit8	L
bit1	Bボタン	bit9	R
bit2	Yボタン	bit10	ZL
bit3	Aボタン	bit11	ZR
bit4	上ボタン	bit12	Lスティック押し込み
bit5	下ボタン	bit13	Rスティック押し込み
bit6	左ボタン	bit14	A.B.X.Yボタンのいずれか
bit7	右ボタン	bit15	上,下,左,右ボタンのいずれか
bit16	スティック押し込み以外のいずれかのボタン		
bit17	タッチパネル		

ファイル名入力専用のダイアログを表示する
初期: 入力初期値 / ファイル名に使えない文字を使用するとエラー
キャ: 39文字
最大: 32まで
戻り値入力値、ただし RESULT関数の返値が-1の場合はキャンセルで文字列は無効
一部命令の実行結果を取得する / RESULTを更新する命令の実行後に使用する
DIALOG, ファイル操作命令, XCTRLSTYLE等の、実行結果を取得する必要のある命令の実行結果を返す

コールバックと結びつけた番号を返す
戻り値
CALL SPRITEの場合コールバックを呼び出したスプライトの管理番号
CALL TEXTの場合コールバックを呼び出したテキストスクリーンのID
CALL SPRITE,CALL TEXT以外で呼び出すと無意味な値が返る

クリップボードの内容を設定する
クリップボードの内容を取得する

ファンクションキーの内容を設定する / ファンクションキーは Ctrl + F1〜F5 / キー番号: 1〜5, 文字列: 256文字まで
ファンクションキーの内容を取得する

フォントの格納座標を取得する / 文字コードにユーザ定義文字(BG)は指定できない、文字列は1文字目のみ対象、フォント種類は 16

文字列や配列で使用するメモリの空き容量を取得する / 総空き容量が返る
容量の単位はバイト。SmileBASICでは1文字2バイト、整数配列1個で4バイト、実数配列1個で8バイト消費
総空: 未使用メモリの合計サイズ
連続: 未使用メモリのうち、一度に使用できるサイズ

ブチコン4を起動してからの経過時間 ms を返す
ブチコン4を起動してからのフレームカウン(約1/60s) を返す / 画面更新回数を計測するために使用

各種情報を取得する
各種情報を取得する
各種情報を設定する

設定項目名文字列	
LANG	Switchに設定されている使用言語名(取得)
SYSBEEP	システムUIの効果音再生フラグ(設定・取得)
TABSTEP	エディタでTABキーを使用する際に挿入される空白の数(取得)
KEYREPEAT	設定メニューのキーリピート値(取得)(リピート開始、リピート間隔)
MOUSECURSOR	マウスカーソルの表示(設定・取得)

実行時間計測を開始する / 計測番号: 0〜7
実行時間計測を終えて、結果をパフォーマンスゲージに登録する
実行時間計測を終えて、結果をパフォーマンスゲージに登録する / 戻り値は μs

カレントプロジェクトのメタデータを読み込む / カレントプロジェクトのアイコンや説明文をメモリ上に読み込み、編集可能にする
メタデータ項目を設定する
現在設定されているメタデータ項目を取得する

メタデータを保存する / 保存ダイアログが表示される

メタデータ項目	設定データ
0	タイトル 最大24文字の文字列
1	説明文 最大240文字の文字列、CHR\$(10)で改行
2	アイコン 要素数1600(縦40,横40)の整数配列(色コード)

各種入力			
各種コントローラー・入力デバイスからの入力命令			
XCTRLSTYLE 最大接続数 操作スタイル[モーションセンサー使用フラグ]IRセンサー使用フラグ[接続画面表示フラグ]		サブプログラムからの呼び出しは無視	
コントローラーの最大接続数、操作スタイル、使用機能を設定する			
最大接続数		接続したい最大のコントローラー数を指定する	
		フル 1～2	
		2本持ち 1～2	
		横持ち 2～4	
		縦持ち 2～4	
操作スタイル			
		0	フル 携帯モード、Joy-Con2本持ち、Proコントローラー
		1	2本持ち Joy-Con2本持ち
		2	横持ち Joy-Con(L)/(R)横持ち、Proコントローラー
		3	縦持ち Joy-Con(L)/(R)縦持ち
モーションセンサー使用フラグ		1:使用、0:未使用	
IRセンサー使用フラグ		1:使用、0:未使用 / 操作スタイルを1(2本持ち)にする必要がある	
接続画面表示フラグ		コントローラー接続画面を必ず表示するかどうかを指定する	
		/ 1で必ず表示、0で設定変更のみ表示(省略時は0)	
XCTRLSTYLE 0		コントローラーの操作スタイルを起動時状態にリセットする / XCTRLSTYLE 1.0.0.0と指定するのと同じ	
XCTRLSTYLE OUT 最大接続数 操作スタイル		サブプログラムからの呼び出しは無視	
コントローラー種類 = CONTROLLER(コントローラーID)		現在設定されているコントローラー最大接続数と操作スタイルを取得する	
		コントローラーの種類や接続状況調べる	
		通常コントローラーのIDは接続したコントローラーのプレイヤーランプの点灯数に対応	
		デフォルトコントローラーは全接続コントローラーの入力を合成し、現在設定されている操作スタイルに関係なく、	
		常に携帯モードコントローラー相当の入力として扱う	
		コントローラーID1～4を使用する場合は事前にXCTRLSTYLE命令で使用したい操作スタイルを設定しておくことを強く推奨する	
		戻り値:コントローラー種類	
		0	未接続
		1	携帯モードコントローラー
		2	Proコントローラー
		3	Joy-Con 2本持ち
		4	Joy-Con(L)
		5	Joy-Con(R)
CONTROLLER コントローラーID OUT コントローラー種類,メイン色,サブ色		コントローラーの接続状況や種類を調べる / 色コードで返す / 未接続の場合は黒が返る	
CONTROLLER コントローラーID OUT コントローラー種類,左メイン色,左サブ色,右メイン色,右サブ色			
値 = BUTTON(コントローラーID,ボタンID[,機能ID])		コントローラーの特定ボタンの操作状態を取得する / 指定ボタンが指定機能IDの条件を満たしていれば1、そうでなければ0	
ボタン値 = BUTTON(コントローラーID)		コントローラの全ボタン状態をまとめて取得する	
ボタン値 = BUTTON(コントローラーID,-1[,機能ID])		コントローラの全ボタン状態をまとめて取得する	
		機能ID	
0		ボタンが押されている	
1		ボタンを押した瞬間(BREPEAT対応あり)	
2		ボタンを押した瞬間(BREPEAT対応なし)	
3		ボタンを離した瞬間	
		ボタンID	フル、2本持ち
		#B_RUP	Xボタン
		#B_RDOWN	Bボタン
		#B_RLEFT	Yボタン
		#B_RRIGHT	Aボタン
		#B_LUP	上ボタン
		#B_LDOWN	下ボタン
		#B_LLEFT	左ボタン
		#B_LRIGHT	右ボタン
		#B_L1	
		#B_SL	L SL SL
		#B_R1	
		#B_SR	R SR SR
		#B_L2	
		#B_S1	ZL LまたはR LまたはR
		#B_R2	
		#B_S2	ZR ZLまたはZR ZLまたはZR
		#B_LSTICK	Lスティック押し込 スティック押し込み スティック押し込み
		#B_RSTICK	Rスティック押し込み
		#B_RANY	A,B,X,Yボタンのいずれか
		#B_LANY	上,下,左,右ボタンのいずれか
		#B_ANY	スティック押し込み以外のいずれかのボタン
		・方向ボタンの上下左右は、コントローラーを持った方向に対するボタン位置。縦持ちの「上」は横持ちの「左」となる	
BREPEAT ボタンID,リピート開始時間,リピート間隔		ボタンのリピート速度を設定する。 / 指定は frame	
BREPEAT ボタンID OUT リピート開始時間,リピート間隔		ボタンのリピート速度を取得する。 / 戻りは frame	
値 = STICK コントローラーID[,スティックID] OUT X,Y		コントローラーのスティック状態を調べる / スティックID は 0: L, 1: R /	
		戻り値:横方向(X)には左<->右で-1.0<->1.0、縦方向(Y)には上<->下で-1.0<->1.0	
センサー			
ACCEL コントローラーID[,センサーID] OUT X,Y,Z		XCTRLSTYLEでモーションセンサーを有効にしていない場合は正常な値が返らない	
		加速度センサーの情報を取得する / 戻り値の単位は重力加速度。	
		戻り値 / コントローラーID 0 は指定できない / センサーID 1は2本持ちの場合のみ有り	
		Xは方向キー左右方向(右ボタン方向が正)	
		Yは方向キー上下方向(下ボタン方向が正)	
		Zはコントローラー表面に対して垂直方向(上向きが正)	
GYROV コントローラーID[,センサーID] OUT X,Y,Z		ジャイロセンサーの角加速度を取得する / 戻り値の単位はラジアン。	
GYROA コントローラーID[,センサーID] OUT X,Y,Z		ジャイロセンサーの角度を取得する / 戻り値の単位はラジアン。	
		戻り値	
		Xはピッチ	
		Yはロール	
		Zはヨー	
		それぞれ正の向きに対して右ネジの回転する向きが正の値となる	
GYROSYNC コントローラーID[,センサーID]		ジャイロセンサーの角度をリセットする	
		命令を呼び出した直後のコントローラー姿勢がX:0,Y:0,Z:0となる	
		実際の呼び出しが行われるまで最大で1Vカウント(V sync count → frame)の時間	
振動			
VIBRATE コントローラID[,左右指定],振動パターン番号		コントローラーを振動させる / サブプログラムからの呼び出しは無視 / 左右指定: 0: 左,1: 右	
VIBRATE コントローラID[,左右指定],低帯域周波数,低帯域振幅,高帯域周波数,高帯域振幅		振動,プリセットの振動パターン番号0～62 / 0を指定すると振動が停止する	
		周波数と振幅を指定してコントローラーを振動させる	
		周波[Hz]	0 で停止
		振幅 -128.0～127.0	0 で停止
値 = VIBRATE(コントローラID[,左右指定])		戻り値:振動パターン番号、周波数・振幅指定時は 65535、振動してないなら 0、左右指定省略時は 右状態値*65536+左状	
VIBRATE		全コントローラーの振動を停止する	
タッチ			
TOUCH [タッチID[,座標系変換フラグ]] OUT タッチ時間,X,Y[,タッチ面積]		タッチパネルのタッチ情報を返す/ TVモードではTOUCH関数は有効な値を返さない	
		タッチ0～9 タッチを開始したポイント順 / 省略した場合は 0 扱い	
		座標:0 で BASIC画面内座標、1 で液晶画面座標	
		タッチタッチを開始してから現在までのフレームカウント。タッチしていない場合 0	
		X,Y タッチされている場合の座標。タッチされていない場合は無意味な値が入る。	
		面積 タッチされている場合のタッチ面積。タッチされていない場合は無意味な値が返る	
マウス			
MOUSE [座標系変換フラグ] OUT X,Y[,ホイール]		マウスの座標情報を返す	
MBUTTON()		マウスの全ボタン状態をまとめて取得する	
MBUTTON(ボタンID[,機能ID])		マウスの特定ボタンの操作状態を取得する / 指定ボタンが指定機能IDの条件を満たしていれば1、そうでなければ0	
		ボタン 調べたいボタンのID	
		0	左ボタン
		1	右ボタン
		2	中央ボタン
		3	ボタン4
		4	ボタン5
		機能ID:知りたい状態の種類。省略すると0が指定される。	
		0	ボタンが押されている
		1	(使用不可)
		2	ボタンを押した瞬間
		3	ボタンを離した瞬間

IRカメラ

IRカメラ

IRSTART コントローラーID,IR認識タイプ

XCTRLSTYLEであらかじめIR使用フラグをONにしておく必要がある / サブプログラムからの呼び出しは無視される

IRカメラの使用を開始する

コントローラーID,IRカメラを使用するコントローラーのID,1～2 / コントローラーIDの詳細はCONTROLLER関数のヘルプを参照

IR認識タイプ

- 0 クラスタリングモードは明るい領域を矩形として認識する。同時に最大16個の領域を認識
1 撮影モード 撮影したデータはIRSPRITEで表示可能、データ取り出し不可、起動ちゅうは画面キャプチャ禁止
2 モーメントモード 撮像範囲を8x6の小領域に分割し、それぞれの認識情報を取得

IRSTART コントローラーID,0,[照明タイプ],[ゲイン],[露出],[最小クラスタ],[最大クラスタ],[最低輝度]

IRカメラの使用を開始する(クラスタリングモード詳細パラメーター設定型)

照明タイプ IRカメラの照明を指定する / 省略時の値は0

- 0 全て(遠+近)
1 遠距離用のみ
2 近距離用のみ
3 照明OFF

ゲイン IRカメラの撮影ゲイン,1～16 / 省略時の値は2

露出 IRカメラの撮影露出時間,7～600 / 省略時の値は200

最小クラスタ クラスタとして認識する最小の面積(ピクセル数),0～76800 / 省略時の値は3

最大クラスタ クラスタとして認識する最大の面積(ピクセル数),0～76800 / 省略時の値は76800

最低輝度 クラスタとして認識する最低の輝度,0～255 / 省略時の値は150

IRSTART コントローラーID,1,[照明タイプ],[ゲイン],[露出],[イメージサイズ]

IRカメラの使用を開始する(撮影モード詳細パラメーター設定型)

イメージサイズ 撮影画像のサイズを指定する / 320×240で最短4秒、以下サイズが小さくなるごとに転送時間は 4倍速くなる

- 0 320×240 転送 4秒
1 160×120 転送 1秒
2 80×60(省略時) 転送 1/4秒(4fps)
3 40×30 転送 1/16秒(16fps)
4 20×15 転送 1/64秒(60fps)

IRSTART コントローラーID,2,[照明タイプ],[ゲイン],[露出]

IRカメラの使用を開始する(モーメントモード詳細パラメーター設定型)

IRSTOP コントローラーID

IRカメラの使用を停止する

値 = IRSTATE(コントローラーID)

IRカメラの開始状況を取得する

- 返値 IRカメラの状況
0 使用していない
1 停止中
2 クラスタリング起動中
3 撮影モード起動中
4 モーメントモード起動中

IRREAD コントローラーID OUT サンプリング番号,ノイズレベル,データ数,データ配列

IRカメラの認識結果を読み取る

サンプリング番号 新しいデータを読み取るたびに1増えていく番号 / 同じ番号を連続して取得した場合は、新しいデータが未受信

ノイズレベル 認識中のノイズレベル / 0～2の値が返り、値が小さいほどノイズが少なく、正確な値が取れている事を示す

データ数 データ配列に含まれるデータの数

クラスタリングモードの場合、認識したクラスタの数を示す。配列には各クラスタの情報が入る。

モーメントモードの場合常に48。配列には撮影領域内を横8×縦6に分割した各ブロックの情報が入る。

データ配列 読み取った認識結果を格納した配列 / 幅8の2次元配列 / 高さは0～データ数-1までに意味のある値が入っている

・座標の範囲はX:(0～320) Y:(0～240)

データ配列 クラスタリング

- 0 クラスタX座標
1 クラスタY座標
2 クラスタ幅
3 クラスタ高さ
4 クラスタのピクセル数
5 クラスタの重心座標X
6 クラスタの重心座標Y
7 クラスタの平均輝度

データ配列 モーメントモード

- 0 ブロックの平均輝度
1 ブロックの重心座標X
2 ブロックの重心座標Y

IRSPRITE コントローラーID,スプライトID

撮影モードで取得したモーションIRカメラ画像をスプライトに貼り付ける / SPPAGEを設定した場合元に戻す事はできない

結果 = KEYBOARD(スキャンコード[機能ID])

USBキーボードの入力情報を取得する

KEYBOARD スキャンコード[機能ID] OUT 結果,モディファイヤキー情報

USBキーボードの入力情報を取得する

機能ID 知りたい状態の種類。省略すると0が指定される。

- 0 キーが押されている
1 使用不可(指定すると結果としてゼロが返る)
2 キーを押した瞬間
3 キーを離した瞬間

戻り結果 キーが機能IDで指定した状態であれば1、なければ0 / USBキーボード未接続は 0

モディファイヤモディファイヤキーの押下情報をビットで返す

- bit0 Windowsキー
bit1 Shiftキー
bit2 Ctrlキー
bit3 Altキー

Toy-Con

Toy-Conの接続設定をする

XCTRLSTYLE 最大接続数,100,Toy-Con種類左1,Toy-Con種類右1[接続画面表示フラグ]

XCTRLSTYLE 最大接続数,100,Toy-Con種類左1,Toy-Con種類右2,Toy-Con種類左2,Toy-Con種類右2[接続画面表示フラグ]

TOPIANO コントローラーID OUT 認識フラグ,キー・ボタン情報[ダイヤル種類[ダイヤル回転角速度]]

TCROBOT コントローラーID OUT 認識フラグ,手足情報配列[ピッチ,ロール,ヨー]

TCVISOR コントローラーID OUT 認識フラグ,ピッチ,ロール,ヨー

TCHOUSE コントローラーID OUT 認識フラグ,情報配列

TCFISHING コントローラーID OUT 認識フラグ,釣り竿ヨー,釣り竿ピッチ,リール速度[釣り竿ヨー角度,釣り竿ピッチ角度]

TCBIKE コントローラーID OUT 認識フラグ,ハンドル傾き,アクセル量[アクセル角度,ウイリー角度]

ファイル	
データのセーブ・ロード・ファイル操作命令	
FILES "[ファイル種類:] "[文字列配列]	ファイルの一覧をコンソール上に表示する / 文字列配列 指定でコンソールではなく配列に取得
FILES "[ファイル種類:]プロジェクト名" [文字列配列]	プロジェクトのファイルの一覧をコンソール上に表示する / 文字列配列 指定でコンソールではなく配列に取得
FILES "/" [文字列配列]	プロジェクトの一覧をコンソール上に表示する / 文字列配列 指定でコンソールではなく配列に取得
文字列配列 = FILES("[ファイル種類:]")	ファイル一覧が格納された配列を作成する
文字列配列 = FILES("[ファイル種類:]プロジェクト名")	プロジェクトのファイル一覧が格納された配列を作成する
文字列配列 = FILES("/")	プロジェクト一覧が格納された配列を作成する
LOAD "[ファイル種類:]ファイル名"[スロット]	プログラムを読み込む / TXTのみ指定可能 / スロット0～3 省略時 0
LOADG "[ファイル種類:]ファイル名",グラフィックページ[0X,OY] [OUT 幅,高さ]	画像ファイルを読み込む
内容 = LOADV("[ファイル種類:]ファイル名")	ファイルの内容を読み込んで文字列または配列として返す / 戻り値 内容: DAT,GRP: 配列(失敗時 要素0)、TXT: 文字列(失敗時空文)
LOADV "[ファイル種類:]ファイル名",数値配列	ファイルの内容を数値配列に読み込む / DAT, GRP 指定
保存	結果は RESULT() / 上書き時は @BACKUP 種類 を生成
SAVE "[ファイル種類:]ファイル名"[スロット]	プログラムをセーブする / TXT 指定、省略時もTXT
SAVEG "[ファイル種類:]ファイル名",グラフィックページ[始点X,始点Y,幅,高さ]	グラフィックページの内容をファイルに保存する / GRP指定、省略時もGRP
	グラフィックページ0～5
	始点X,始点Y 保存したい領域の左上の座標を指定する(X,Y共に0～2047) 省略時 0, 0
	幅,高さ 保存したい領域のサイズを指定する(幅,高さ共に1～2048) 省略時 2048, 2048
	/ 始点X*幅 > 2048、始点Y*高さ > 2048 でエラー
	変数の内容をファイルに保存する / TXT, DAT, GRP 指定 / 内容: DAT: 数値配列、GRP:2次元正数配列、TXT: 文字列
SAVEV "[ファイル種類:]ファイル名",内容	カレントプロジェクトを変更する
	カレントプロジェクトを取得する
PROJECT "プロジェクト名"	プログラムのロードと実行を行う / DIRECTモード実行不可 / TXT のみ指定可能
PROJECT()	指定スロットのプログラムを実行する / EXECで実行をはじめたスロットでEND命令を実行するとEXECの次の命令に戻ってくる
EXEC "[ファイル種類:]ファイル名文字列"[スロット]	指定したファイルが存在するかどうかチェックする / 戻り値: 1: 存在, 0: なし
EXEC スロット	ファイルを削除する / 結果は RESULT() で 成功時は1、失敗時は0、キャンセル時は-1 / ダイアログ表示
	ファイルの名前を変更する / 結果は RESULT() で 成功時は1、失敗時は0、キャンセル時は-1 / ダイアログ表示
値 = CHKFILE("[ファイル種類:]ファイル名文字列")	変更元種類 省略時 TXT
DELETE "[ファイル種類:]ファイル名"	変更先種類
RENAME "[変更元種類:]変更元ファイル名","[変更先種類:]変更先ファイル名"	変更元種類と違う場合エラー、省略時は変更元種類
スクリーン制御	
画面全体の表示制御命令	
ACLS	画面設定を全てクリアし、起動時状態に戻す
ACLS GRP保存フラグ,SPDEF保存フラグ,GRPF保存フラグ	画面設定を全てクリアし、起動時状態に戻す
	GRP保存1を指定するとフォントページを除いたグラフィックページをクリアせずに保存する
	SPDEF保1を指定するとSPDEFの内容を保存する
	GRPF保1を指定するとフォントページをクリアせずに保存する
XSCREEN 幅,高さ[サンプル倍率[合成モード[アスペクト比]]]	画面解像度を設定する
	コンソール(テキストスクリーン4)のサイズは幅,高さに合わせて変更され、それ以外のテキストスクリーン設定は変わらない
	グラフィック表示用スプライトの設定は幅,高さに合わせて変更される
	幅、高さ 画面の幅と高さをピクセル数で指定する / 4で割り切れる数を指定可能
	幅:128～1280
	高さ:128～720
	サンプル倍率 ピクセルをさらに細かく分割して、回転や拡大をよりなめらかに表現する事ができる / 省略時は1
	・1以上、幅にサンプル倍率をかけた物が1280以内、高さにサンプル倍率をかけた物が720以内の値まで指定可能
合成モード	BASIC画面を最終的に表示画面に合成する際の描画方法を指定する
0	バイリニア ピクセルの角がなめらかになるが、全体にぼやけた表示になる / 省略時の指定
1	スマートニアレストと2の間
2	ニアレストネイバー は全体的にくっきりした表示になるが、ピクセルの角がギザギザになる
XSCREEN OUT 幅,高さ[サンプル倍率[合成モード[アスペクト比]]]	画面解像度を取得する
ANIMDEF 定義番号,アニメ対象,データ配列[ループ]	アニメーションの定義 / SPANIMやTANIM命令でできる
	定義番号 アニメーション定義番号0～1023
	アニメ対象 変化させる要素を管理する数値または文字列
	0 XY XY座標
	1 Z Z座標
	2 R 回転角度
	3 S 倍率XY
	4 C 表示色
	5 V 変数(スプライト変数7の値)
	6 UV UV座標(定義元画像座標)
	7 I 定義番号
	文字列の末尾に"+"を付けるか、数値に8を加えるとアニメーション開始時点からの相対値となる
	文字列の末尾に"-"を付けるか、数値に16を加えるとアニメーション終了時にスプライトを削除する
データ配列	アニメデータが格納された1次元数値配列 / 時間1,項目1,[項目2],時間2,項目1,[項目2]… (最大32個まで)
ループ	ループ回数:1～ 0を指定することで無限ループとなる / 省略時 1
	アニメーションの定義(DATA命令で指定)
	アニメーションの定義(直接引数として指定)
	指定定義番号のアニメーションをクリアする
	アニメーション定義をすべてクリアする
BACKCOLOR 背景色	BASIC画面の背景色を設定する
FADE フェード色	画面にフェード効果进行
FADE フェード色,時間frame	フェードイン、フェードアウト効果进行 / アルファ成分が255だとフェードアウト効果、アルファ成分が0だとフェードイン効果
色 = FADE()	現在のフェード色を取得する
値 = FADECHK()	フェードイン・フェードアウト中かどうかを調べる / フェードイン・フェードアウト中であれば1、していない時は0

テキストスクリーン	
指定値	
色	ARGB各8ビットの色コード ARGB8888
スクリーンID	0～4、省略時は4 / 旧来の コンソールが 4, BG が 0～3
レイヤーID	0～7 / 0 が手前で 7 が後ろ、テキストスクリーン、SPRITEで使用される
フォント種類	[8]16]
Z	表示優先度: -4095～4095
テキストスクリーンへの文字表示・データ入力命令	
CLS [スクリーンID]	テキストスクリーンを消去する
COLOR [スクリーンID,]色コード	文字の描画色を設定する
COLOR([スクリーンID])	文字の描画色を取得する
LOCATE [スクリーンID,] [座標X][座標Y]	テキストスクリーンへの文字表示座標を指定する / X, Yは省略すると以前のXY座標を維持
LOCATE [スクリーンID] OUT 座標X座標Y	
PRINT [表示内容[{ }]表示内容…]	コンソールの文字表示 / 表示内容を複数指定時、: は改行無しで、はtab距離空けて表示 / 省略形 ?
TPRINT スクリーンID[表示内容[{ }]表示内容…]	
ATTR [スクリーンID,]水平反転,垂直反転,回転	テキストスクリーンに表示する文字の属性を設定する / 回転: 0: 0度, 1: 90度, 2: 180度, 3: 270度
ATTR [スクリーンID,]表示属性	
ATTR [スクリーンID] OUT 水平反転,垂直反転,回転	テキストスクリーンに表示する文字の属性を設定する / 表示属性 = 垂直反転*8 + 水平反転*4 + 回転
表示属性 = ATTR[[スクリーンID]	
SCROLL [スクリーンID,]文字数X,文字数Y	テキストスクリーンの表示内容をスクロールさせる / X: 正で左,負で右, Y: 正で上, 負で下
文字コード = CHKCHR([スクリーンID,]座標X,座標Y)	テキストスクリーンに表示されている文字の文字コードを調べる / 文字なしなら 0
CHKCHR [スクリーンID,]座標X,座標Y OUT 文字コード,表示属性	
INPUT [“ガイド文字列”]変数[変数2…]	キーボードから数値または文字列を入力する / ENTERキーの入力待ち / テキストスクリーン4 固定
INPUT [“ガイド文字列”]文字列変数	
INKEY\$()	
T系 / 旧来の BG と同じ扱い	コンソール(テキストスクリーン4)の広さと表示に使用するフォントの設定を行う
TSCREEN フォント種類[文字サイズ]	
TSCREEN スクリーンID,フォント種類,文字サイズ[幅,高さ]	幅と高さは画面サイズを文字サイズで割った物が設定される
TSCREEN スクリーンID,フォント種類,幅,高さ	
TSCREEN [スクリーンID] OUT フォント種類,文字サイズ,幅,高さ	テキストスクリーンの広さと表示に使用するフォントの設定を行う
TPAGE [スクリーンID,]グラフィックページ,参照原点X,参照原点Y	
TCOLOR スクリーンID,色コード	幅 x 高さが32768まで指定可能
TCOLOR(スクリーンID)	
TLAYER スクリーンID,レイヤーID	幅、高さを省略した場合、または0を指定した場合、現在の画面サイズを文字サイズで割った物が設定される
TLAYER(スクリーンID)	
TPUT スクリーンID,X,Y[文字列 文字コード] [表示属性]	テキストスクリーンの指定座標に文字を一文字書く / 文字コード0～65535、文字列は先頭の文字のみ有効
TFILL スクリーンID,始点X,始点Y,終点X,終点Y[文字列 文字コード] [表示属性]	
THOME スクリーンID,ホーム座標X,ホーム座標Y	テキストスクリーンのホーム座標(座標基準点)を設定する / TOFS命令の座標基準点、TROT(回転)やTSCALE(スケールリング)の中心
THOME スクリーンID OUT ホーム座標X,ホーム座標Y	
TOFS スクリーンID,X,Y[Z]	テキストスクリーンの表示座標を設定する / THOMEで設定したテキストスクリーン上の座標を、TOFSで設定した画面上の座標に持:
TOFS スクリーンID OUT X,Y[Z]	
TROT スクリーンID,角度	テキストスクリーンの回転角を設定する / 角度は rad ?
角度 = TROT(スクリーンID)	
TSCALE スクリーンID,倍率X,倍率Y	テキストスクリーンの拡大・縮小を行う / THOMEで設定したホームポジションを中心に拡大・縮小 / 2.0で2倍に拡大、0.5で半分に縮
TSCALE スクリーンID OUT 倍率X,倍率Y	
TSHOW スクリーンID	テキストスクリーンを表示する
値 = TSHOW(スクリーンID)	
THIDE スクリーンID	テキストスクリーンを隠す
値 = THIDE(スクリーンID)	
TBLEND スクリーンID,合成方法	テキストスクリーンの合成方法を設定する / 0で通常合成、1で加算合成
合成方法 = TBLEND(スクリーンID)	
TANIM スクリーンID,アニメ対象,データ配列[,ループ]	テキストスクリーンにアニメを設定する(配列で指定)
TANIM スクリーンID,アニメ対象,“@ラベル文字列”[,ループ]	
TANIM スクリーンID,アニメ対象,時間1,項目1[,項目2[,時間2,項目1[,項目2]]…[,ループ]	テキストスクリーンにアニメを設定する(直接引数として指定)
TANIM スクリーンID,アニメーション定義番号	
TANIM [スクリーンID]	テキストスクリーンにアニメを設定する(アニメーション定義番号で指定)
TSTOP [スクリーンID]	テキストスクリーンのアニメーションを停止させる
TSTART [スクリーンID]	
値 = TCHK(スクリーンID)	テキストスクリーンのアニメーションを開始させる
TVAR スクリーンID,変数キー[,値]	テキストスクリーン変数への書き込み / 値は数値または文字列、省略で削除 / 変数キー一名は文字列か数値
値 = TVAR(スクリーンID,変数キー)	
TCOPY コピー先スクリーンID[,コピー元スクリーンID],始点X,始点Y,終点X,終点Y,転送先X,転送先Y	テキストスクリーン内のキャラクタ情報をコピーする
TSAVE 読み出しスクリーンID[,X,Y,幅,高さ]転送先配列	テキストスクリーンのキャラクタ情報を配列へコピーする / XY幅高さ省略時:テキストスクリーン全体
TLOAD 書き込みスクリーンID[,X,Y,幅,高さ]転送元配列	/ 転送先配列の要素が不足する場合、1次元配列に限り自動追加
整数配列 = TARRAY(スクリーンID)	配列からテキストスクリーンにキャラクタ情報をコピーする / XY幅高さ省略時:テキストスクリーン全体
TUPDATE スクリーンID [,始点X,始点Y,終点X,終点Y]	
	テキストスクリーンの内容を表す整数配列を取得する
	TARRAYによる書き込み内容を表示に反映させる
	整数配列 の配列サイズは[スクリーン高さ, スクリーン幅x2], 1文字は2要素で表され
	、1要素目は上位16ビットに表示属性、下位16ビットに文字コード、2要素目は表示色が入る
	TARRAY で取得した配列に対して書き込みを行い、TUPDATE で画面に反映する。VRAM の参照を直接書き換えるイメージ
	TUPDATEを使用しなくても、更新座標付近に他の文字表示命令で書き込みを行うと自動的に反映が行われる
TFUNC スクリーンID[,“@ラベル”][“ユーザー定義命令名”]	テキストスクリーンごとにコールバック処理を割り当てる / 省略でコールバック処理をクリア
	CALL TEXT で全テキストスクリーンのコールバック処理を一度に実行
	コールバック先ではCALLIDX()関数で管理番号を取得できる
	旧来の BGFUNG と同様

グラフィック描画	
指定値	0～2047
X,Y	ARGB各8ビットの色コード ARGB8888
色	GCOLORで設定した色
描画時の省略可能な色コード	0 単純描画(省略時) #G_NORMAL
描画方法	1 透明度が0以外の時単純描画 #G_NORMAL2
	2 半透明描画 #G_ALPHA
	3 描画先の透明度も考慮して半透明描画 #G_ALPHA2
	4 加算描画 #G_ADD
	[8]16]
フォント種類	
グラフィックページに図形を描画する命令	
GTARGET 描画対象ページ	グラフィック描画対象ページを指定する / G系で描画されるページを変更する、他のGRPを更新する場合はこれと呼ぶ
描画対象ページ = GTARGET()	現在のグラフィック描画対象ページを取得する
GCOLOR 色コード	グラフィック描画色を指定する / 描画命令で色を省略した場合に使用される色を設定
色コード = GCOLOR()	グラフィック描画色を取得する
RGB/HSV	RGB値から色コードを計算する / A,R,G,Bともに 0～ 255, A省略時は 255
色コード = RGB([A],R,G,B)	色コードからRGBの各成分を得る
RGB 色コード OUT [A],R,G,B	実数のRGB値から色コードを計算する / A,R,G,Bともに 0～ 1.0, A省略時は 1.0
色コード = RGBF([A],R,G,B)	色コードからRGBの各成分を得る(実数版)
RGBF 色コード OUT [A],R,G,B	HSV値から色コードを計算する / H: 0～ 360 以後360毎に繰り返し数(0 赤→黄→緑→青→紫→赤 360), S,V 0～255
色コード = HSV(H色相,S彩度,V明度)	色コードをHSV値に変換する
HSV 色コード OUT 色相,彩度,明度	
CLIP	
GCLIP [始点X,始点Y,終点X,終点Y]	グラフィック画面の書き込みクリッピング領域を指定する (描画更新できる領域を指定する、それ以外は描画しても描画されない)
GCLIP OUT 始点X,始点Y,終点X,終点Y	グラフィック画面の書き込みクリッピング領域を取得する
描画	
GCLS [色コード]	グラフィックページの内容を指定色で塗りつぶす / GCLIPの設定範囲 が対象 / 値省略で 0 (透明色)で塗りつぶし
色コード = GPGET(座標X,座標Y)	グラフィック画面の指定座標の色を取得する / GRP範囲外の座標を指定すると0
GPSET 座標X,座標Y[色コード[描画方法]]	グラフィック画面に点を打つ
GLINE 始点X,始点Y,終点X,終点Y[色コード[描画方法]]	グラフィック画面に直線を引く
GCIRCLE 中心点X,中心点Y,半径[開始角,終了角,フラグ][色コード[描画方法]]	グラフィック画面に円・円弧を描く / 角 は 0～360, 時計の3時方向が 0 で時計回りに進む / フラグ: 0:円弧,1:扇型
GBOX 始点X,始点Y,終点X,終点Y[色コード[描画方法]]	グラフィック画面に四角形を描く
GFILL 始点X,始点Y,終点X,終点Y[色コード[描画方法]]	グラフィック画面に四角形を描いて塗りつぶす
GPAINT 開始点X,開始点Y[塗りつぶし色[境界色]]	グラフィック画面を塗りつぶす / 境界色:この色で囲まれている領域を塗る、省略時は開始点座標にある色の範囲を塗る
GCOPY [転送元ページ]始点X,始点Y,終点X,終点Y 転送先X,転送先Y,描画方法	他のグラフィックページから画像をコピーする
GCOPY 転送元ページ,始点X,始点Y,終点X,終点Y,転送先X,転送先Y,色コード,描画方法	他のグラフィックページから画像をコピーする / 色コード: コピー元のピクセルと乗算する色コード
GSAVE [転送元ページ][X,Y,幅,高さ]転送先配列	画像を配列へコピーする / XY幅高さ省略時はクリッピング領域を対象
GLOAD [X,Y,幅,高さ]画像配列,描画方法	画像データを配列からグラフィック画面にコピーする / 画像配列 は GSAVEで取得した形式
GLOAD [X,Y,幅,高さ]インデックス画像配列,パレット配列,描画方法	画像データをインデックス画像配列から色変換しながらグラフィック画面にコピーする / インデックス画像配列の値番目のパレット配列
GTRI X1,Y1,X2,Y2,X3,Y3[色コード[描画方法]]	グラフィック画面に三角形を描いて塗りつぶす
GPUTCHR X,Y[文字列 文字コード][フォント種類[色コード[描画方法]]]	グラフィック画面に文字を描く / 文字列は全部表示される? 折り返しは?
GPUTCHR X,Y[文字列 文字コード],フォント種類,スケールX,スケールY,色コード[描画方法]	グラフィック画面に文字を描く / スケール: 表示倍率:1～32767, 省略時 1
VRAM	
整数配列 = GARRAY(グラフィックページ)	グラフィックページの内容を表す整数配列を取得する
GUPDATE グラフィックページ[始点X,始点Y,終点X,終点Y]	GARRAYによる書き込み内容を表示に反映させる
	整数配列は2次元配列 / 2048x2048?
	GARRAY で取得した配列に対して書き込みを行い、GUPDATE で画面に反映する。VRAM の参照を直接書き換えるイメージ
	GUPDATEを使用しなくても、書き込み座標付近に他の描画命令で書き込みを行うと自動的に反映が行われる
	領域からはみでた部分が更新される事もある
GSAMPLE グラフィックページ,サンプリングタイプ	グラフィックページのサンプリング方法を設定する
サンプリングタイプ = GSAMPLE(グラフィックページ)	グラフィックページのサンプリング方法を取得する
	サンプリングタイプ
	0 ボーダー(0～2047以外のUV座標は透明色になる)
	1 リピート(グラフィックページが0～2047の外にもずっと繰り返しているように扱う)

スプライト		
指定値		
管理番号		作成するスプライトの番号:0～4095 SPSET 以外の 管理番号に対して操作・参照する場合、その管理番号のSPSET前に使うとエラー SPDEFで定義したテンプレートの定義番号:0～4095 1で表示、0で表示しない
定義番号		b00 ↑ 90度単位の回転(b00とb01の2ビットで指;省略時は 0 値
表示フラグ		b01 ↓ #A_ROT0、#A_ROT90、#A_ROT0180、#A_ROT270
アトリビュート		b02 横反転(0=OFF、1=ON)、#A_REVH b03 縦反転(0=OFF、1=ON)、#A_REVV b04 加算合成(0=OFF、1=ON)、#A_ADD 表示優先度:→4095～4095 -1～5 / 初期状態のスプライト用ページは4 0～7 / 0 が手前 で 7 が後ろ、テキストスクリーン、SPRITEで使用される
Z		
グラフィックページ		
レイヤーID		
グラフィックページ内の画像を高速・大量に表示する命令		
SPSET 管理番号,定義番号[表示フラグ]		スプライトを作成する(定義テンプレートを利用) スプライトを作成する(直接画像情報等を指定) / UVは参照元の GRP の座標(0～2048)、WHはその範囲
SPSET 管理番号U,V,W,H[アトリビュート][表示フラグ]		空いている管理番号を探してスプライトを作成し、管理番号を返す / 空きがなくな生成できなかった時は-1が返る
管理番号 = SPSET(定義番号[表示フラグ])		空いている管理番号を探してスプライトを作成し、管理番号を返す
管理番号 = SPSET(U,V,W,H,アトリビュート[表示フラグ])		空いている管理番号を探してスプライトを作成し、管理番号を返す
管理番号 = SPSET(開始番号,終了番号,U,V,W,H,アトリビュート[表示フラグ])		空いている管理番号を探してスプライトを作成し、管理番号を返す
管理番号 = SPSET(開始番号,終了番号,U,V,W,H,アトリビュート[表示フラグ])		
SPCLR [管理番号[管理番号2]]		指定スプライトを削除する / 2つ指定した場合はその範囲の管理番号を削除する / 省略時は全スプライトを削除する。グラフィック表示用スプライト(4095)は削除されない
値 = SPSHOW 管理番号		スプライトを表示させる / SPSET前に使うとエラー / 戻り値: 1:表示中, 0:非表示
SPHIDE 管理番号		スプライトの表示を隠す
値 = SPHIDE(管理番号)		スプライトが消えているかどうかを取得する / 戻り値: 1:非表示, 0:そうではない
SPHOME 管理番号,座標X,座標Y		スプライトのホーム座標(座標基準点)を指定する
SPHOME 管理番号 OUT HX,HY		スプライトのホーム座標を取得する / SPOFS命令の座標基準点、SPROT(回転)やSPSCALE(スケーリング)の中心点、衝突判定の中心座標として使用
SPOFS 管理番号[X][Y][Z]		スプライトを移動させる
SPOFS 管理番号[座標系フラグ] OUT X,Y,Z]		スプライトの表示座標を取得する / 座標系フラグ: 1:絶対座標, 0:相対座標(SPLINK時) 省略時 0 (SPLINK時は相対座標が取得できる
SPROT 管理番号,角度		スプライトを回転させる / 角度: 0～360(時計回り)
角度 = SPROT(管理番号)		スプライトの回転角度を取得する
SPSCALE 管理番号,倍率X,倍率Y		スプライトのスケール(表示倍率)を変更する / スケールを考慮した当たり判定は先にSPCOLを実行
SPSCALE 管理番号 OUT SX,SY		スプライトの表示倍率を取得する / 倍率は 2.0で2倍に拡大、0.5で半分に縮小
SPCOLOR 管理番号,色コード		スプライトの表示色を設定する / 実際の表示色は色コードに元のピクセル色を乗算した値
色コード = SPCOLOR(管理番号)		スプライトの表示色を取得する
SPCHR 管理番号,定義番号		スプライトのキャラクタ定義を変更する
SPCHR 管理番号[U][V][W][H][アトリビュート]		スプライトのキャラクタ定義を変更する
SPCHR 管理番号 OUT U,V,W,H,ATTR]		スプライトのキャラクタ定義情報を取得する
定義番号 = SPCHR(管理番号)		スプライトのキャラクタ定義番号を取得する
SPPAGE 管理番号,グラフィックページ		指定スプライトの参照グラフィックページを設定する / 初期状態は 4
SPPAGE(管理番号)		指定スプライトの参照グラフィックページを取得する
SPPAGE グラフィックページ		これから作成するスプライトに割り当てるグラフィックページを設定する
グラフィックページ		SPPAGE これから作成するスプライトに割り当てるグラフィックページを取得する
SPLAYER 管理番号,レイヤーID		指定スプライトの所属レイヤーを設定する
レイヤーID = SPLAYER(管理番号)		指定スプライトの所属レイヤーを取得する
SPLAYER レイヤーID		これから作成するスプライトが所属するレイヤーを設定する
レイヤーID = SPLAYER()		これから作成するスプライトが所属するレイヤーを取得する
SPDEF 定義番号,U,V[W,H[原点X,原点Y]][アトリビュート]		スプライトのキャラクタ定義用テンプレートを作成する
SPDEF 数値配列[定義番号オフセット][Uオフセット,Vオフセット]		スプライトのキャラクタ定義用テンプレートを配列から一括作成する / 1個分の要素はU,V,W,H,原点X,原点Y,アトリビュートの7つ
SPDEF “@ラベル文字列”[定義番号オフセット][Uオフセット,Vオフセット]		スプライトのキャラクタ定義用テンプレートをDATA列から一括作成する / 定義は上記と同じ
SPDEF 定義番号 OUT U,V[W,H,HX,HY][A]		スプライトのキャラクタ定義テンプレートの情報を取得する
SPDEF 定義番号,元になる定義番号,[U][V][W][H][[原点X],[原点Y],[アトリビュート]		スプライトキャラクタ定義用テンプレートをコピーする
SPDEF		スプライトキャラクタ定義用テンプレートを初期状態に戻す
SPLINK 管理番号,リンク先管理番号[リンクフラグ]		スプライトを別のスプライトにリンクさせる リンク元より大きい管理番号を指定するとエラー 管理番号が子、リンク先管理番号が親、親の動きにあわせて子も動く リンクフラグ 省略時は 0値
リンク先番号 = SPLINK(管理番号)		指定された管理番号スプライトのリンク先番号を取得する / 親を取得する
SPUNLINK 管理番号		スプライトのリンクを解除する
SPANIM 管理番号,アニメ対象,データ配列[ループ]		スプライトにアニメーションを設定する
SPANIM 管理番号,アニメ対象,”@ラベル文字列”[ループ]		スプライトにアニメーションを設定する(DATA命令で指定)
SPANIM 管理番号,アニメ対象,時間1,項目1[項目2[時間2,項目1[項目2]…][ループ]		スプライトにアニメーションを設定する(直接引数として指定)
SPANIM 管理番号,アニメーション定義番号		スプライトにアニメーションを設定する(アニメーション定義番号で指定)
SPANIM [管理番号]		スプライトアニメーションをクリアする / 管理番号を省略すると全スプライトのアニメーションをクリア
SPSTOP [管理番号]		スプライトのアニメーションを停止させる / 管理番号を省略すると全スプライトのアニメーションを停止
SPSTART [管理番号]		指定スプライトのアニメーションを再開する / 管理番号を省略すると全スプライトのアニメーションを開始
値 = SPCHK(管理番号)		スプライトのアニメーション状態を取得する
	返値	bit 0の時アニメ停止中 b00 XY座標 1 #CHKXY b01 Z座標 2 #CHKZ b02 回転 4 #CHKR b03 倍率XY 8 #CHKS b04 表示色 16 #CHKC b05 変数 32 #CHKV b06 UV座標 64 #CHKUV b07 定義番号 128 #CHKI
SPVAR 管理番号,変数キー-[値]		スプライト変数を設定する / 値は数値または文字列、省略で削除 / 変数キー名は文字列か数値
値 = SPVAR(管理番号,変数キー)		スプライト変数を読み出す / 未設定時は値は 0 / プログラム実行開始時、ACLS呼び出し時にクリアされる
SPCOL 管理番号[スケール対応][マスク]		スプライト衝突判定情報を設定する / マスク 0～&HFFFFFFF(32ビット) / スケール対応: 0: 無視(省略時), 1: SPSCALEに判定領域を同期、SPCOL命令後に設定したSPSCALEから有効
SPCOL 管理番号,始点X,始点Y,幅,高さ[スケール対応][マスク]		スプライト衝突判定結果を返す / 戻り値は衝突したスプライトの管理番号、衝突のないとき-1
SPCOL 管理番号 OUT 始点X,始点Y,幅,高さ[スケール対応][マスク]]		指定スプライトとのスプライト衝突判定結果を返す / 戻り値は 0=衝突なし、1=衝突 直前に設定した情報からスプライトの衝突判定結果を返す
SPCOLVEC 管理番号[移動量X,移動量Y]		スプライト衝突判定用移動速度を設定する / 移動量省略時、SPANIMのXYを線形補完で実行中:前フレームからの移動距離、それ以外の場合:0.0
衝突管理番号 = SPHITSP(管理番号[先頭ID,末尾ID])		スプライトの衝突判定結果を返す / 戻り値は衝突したスプライトの管理番号、衝突のないとき-1
値 = SPHITSP(管理番号,相手管理番号)		指定スプライトとのスプライト衝突判定結果を返す / 戻り値は 0=衝突なし、1=衝突 直前に設定した情報からスプライトの衝突判定結果を返す
衝突管理番号 = SPHITSP()		
衝突管理番号 = SPHITRC(始点X,始点Y,幅,高さ[マスク],移動量X,移動量Y])		動く四角形とすべてのスプライトの衝突判定結果を返す / 戻り値は衝突したスプライトの管理番号、衝突のないとき-1
値 = SPHITRC(管理番号,始点X,始点Y,幅,高さ[マスク],移動量X,移動量Y])		指定したスプライトと四角形の衝突判定結果を返す / 戻り値は 0=衝突なし、1=衝突
値 = SPHITRC(先頭ID,末尾ID,始点X,始点Y,幅,高さ[マスク],移動量X,移動量Y])		指定範囲のスプライトと四角形の衝突判定結果を返す / 戻り値は 0=衝突なし、1=衝突 直前に設定した情報からスプライトの衝突判定結果を返す
衝突管理番号 = SPHITRC()		
衝突時間 = SPHITINFO()		衝突判定 判定時の座標×速度×衝突時間が衝突座標と一致
SPHITINFO OUT TM,X1,Y1,X2,Y2		衝突判定結果の詳細情報取得(衝突時間と座標) / TM: 衝突時間, X1,Y1: 衝突時の物体1の座標, X2,Y2: 衝突時の物体2の座標
SPHITINFO OUT TM,X1,Y1,VX1,VY1,X2,Y2,VX2,VY2		衝突判定結果の詳細情報を取得する(衝突時間と座標と速度) / VX1,VY1: 衝突時の物体1の速度, VX2,VY2: 衝突時の物体2の速度
SPFUNC 管理番号[“@ラベル”] “ユーザー定義命令名”]]		スプライトごとにコールバック処理を割り当てる / 省略でコールバック処理をクリア CALL SPRITE で全スプライトのコールバック処理一度に実行 コールバック先ではCALLIDX()関数で管理番号を取得できる 旧来の SPFUNC と同様
値 = SPUSED(管理番号)		指定されたスプライトが使われているか調査する / 0=空き、1=使用中

レイヤー		
指定値		
レイヤーID		0～7 / 0 が手前で 7 が後ろ、テキストスクリーン、SPRITEで使用される
スプライトとテキストスクリーンの表示内容をさらに高度に加工する命令 LAYER レイヤーID[合成タイプ[合成色]]		
LFILTER レイヤーID,フィルター種類,フィルターパラメーター LFILTER レイヤーID[0]		レイヤーの設定を行う 合成タイプ 0 合成しないで上書き (省略値) 1 単純合成 2 加算合成 3 乗算合成 指定レイヤーにフィルターを適用させる フィルター種類が0(なし)の場合はフィルターパラメータを指定できない フィルターフィルターパラメーター 0 フィルターなし (省略値) 1 モザイク 整数(モザイクの大きさ) 2 ぼかし 整数(ぼかし量) 3 水平ラスター変形 実数配列(変形情報),整数(リピート) 4 垂直ラスター変形 実数配列(変形情報),整数(リピート) 5 色変換 実数(色相),実数(彩度),実数(明度) ・モザイクの大きさは大きいほど画像が粗くなる ・ぼかし量は大きいほど画像がぼける ・変形情報配列は2次元の実数配列で、2次元目の要素数が2または4である必要がある ・変形情報配列の要素数が2の場合、オフセットXスケール値が連続する ・変形情報配列の要素数が4の場合、XオフセットXスケール,YオフセットYスケールが連続する ・変形情報配列のオフセットは画面幅または高さを1.0として指定する ・ラスター変形のリピートは変形結果が画像外に出た時に反対側を参照するかどうかを指定する ・色変換の引数は描画色の各色要素に加算する値を指定する。HSV関数を参照
フィルター種類 = LFILTER(レイヤーID)		レイヤーに設定されているフィルター種類を取得する
LCLIP レイヤーID [左上X座標,左上Y座標,右下X座標,右下Y座標] LCLIP レイヤーID OUT 左上X座標,左上Y座標,右下X座標,右下Y座標		レイヤーのクリッピング設定を設定・解除する / 値省略で解除 レイヤーのクリッピング矩形座標を取得する
LMATRIX レイヤーID,ホーム座標X,ホーム座標Y[X,Y[倍率X,倍率Y[回転角]] LMATRIX レイヤーID,変換行列		レイヤー内の表示要素を描画する際に使用する変換行列を設定する レイヤー内の表示要素を描画する際に使用する変換行列を設定する / 要素数が16必要、単精度実数範囲内、行列はOpenGLの変換行列に準じた形式で、射影行列まで含んでいる必要が レイヤーの変換行列をリセットする
LMATRIX レイヤーID		
サウンド		
サウンド関連命令		
BEEP [効果音番号][音程][音量][パンポット]]] ハンドル = BEEP([効果音番号][音程][音量][パンポット]]])		単純な警告音・効果音の発生 単純な警告音・効果音の発生
		効果音番号 説明 0～155 プリセット音 256～383 BGM音源 音程 -32768～32767 (100で半音)、省略時 0 / ピッチ 音量 0～127、省略時 64 パンポット 0(左)～64(中央)～127(右)、省略時 64
BEEPPAN ハンドル,パンポット BEEPPIT ハンドル,ピッチ BEEPSTOP [ハンドル][フェード時間]] BEEPVOL ハンドル,音量		効果音のパンポットを変化させる 効果音の音程を変化させる 効果音停止 / ハンドル省略時は全効果音 効果音の音量を変化させる
BGMCLEAR [ユーザー定義曲番号] BGMCONT [トラック番号][フェード時間]]		ユーザー定義音楽の消去 / 曲番号:128～255(省略時すべての定義を消去) 一時停止中の音楽演奏を再開(一時停止にはBGMPAUSEを利用) / トラック番号:0～15(省略時は全トラック) / フェード時間: 秒(小数指定可能、0=即時再開、省略時0)
BGMPITCH [トラック番号,]ピッチ BGMPLAY [トラック番号,]曲番号[音量][フェード時間]		演奏している音楽の基準音程を変化させる 音楽演奏 / フェードインする時間: 0～32767 秒 曲番号 プリセット音源(0～45) ユーザー定義(128～255)
BGMPLAY MML文字列 BGMPAUSE [トラック番号][フェード時間]] BGMSET ユーザー定義曲番号,MML文字列 BGMSETD ユーザー定義曲番号,@ラベル文字列 BGMSTOP [トラック番号][フェード時間]]		音楽演奏(入力したMMLデータを再生) / MMLによる再生はトラック0、ユーザー定義曲番号255 がMMLによる曲に書き換わる 音楽演奏一時停止(再開にはBGMCONTを利用) ユーザー定義音楽の事前定義 MMLでのユーザー定義曲の事前定義 / 内部的にはRESTOREと同じ扱い / DATAの終端は数値で判断(DATA 0) 音楽演奏停止 / 省略時は全トラック
BGMVAR トラック番号,変数番号,値 BGMVAR トラック番号,変数番号 OUT 変数		BGMTトラック変数の値を設定 / 変数番号:0～7 BGMTトラック変数の値を取得 / 変数番号:0～7
BGMVOL [トラック番号,]音量 BGMWET [トラック番号,]ウェット		指定トラックの音量を設定 指定トラックのエフェクタ効果量を設定 / .0(ドライ)～127(ウェット)
EFCEN 状態 EFCEN OUT 変数 EFCSET プリセット番号 EFCSET 反射音モード,反射音音量,反射音時間,残響音モード,残響音音量,残響音時間,高音減衰率,残響音音色,入力音量 EFCWET BEEPウェット,BGMウェット,TALKウェット,PCMウェット		エフェクタの状態を設定 / 0: OFF, 1: ON エフェクタの状態を取得 / 0: OFF, 1: ON エフェクタのプリセット設定 / 0: #EFCOFF OFF, 1: #EFCBATH 風呂, 2:#EFCCAVE 洞窟, 3:#EFCSPACE 宇宙 エフェクタ詳細設定 エフェクタ ウェット設定
PCMCONT PCMPOS OUT 変数 PCMSTOP PCMSTREAM PCMデータ配列[サンプリング周波数] PCMSTREAM PCMデータ配列L,PCMデータ配列R[サンプリング周波数] PCMVOL [チャンネル,]音量		PCM再生再開 PCM再生位置取得 PCM再生停止 PCM再生開始 / サンプリング周波数(Hz):1～192000(省略時48000) / 1次元配列はモノラル、2次元配列はステレオ PCM再生開始 PCM音量設定 / 0 左, 1 右 / 音量:-32767～32767(負の値を指定すると位相が逆になる)
SNDMSBAL バランス[フェード時間] SNDMSBAL(バランス)		メインインタプリタ/サブインタプリタの音量バランスを設定する / サブプログラム内でのみ設定可能 メインインタプリタ/サブインタプリタの音量バランスを取得する / バランス:0(サブ側のみ)～64(メイン/サブ均等)～127(メイン側のみ)
SNDMVOL 音量[フェード時間] SNDMVOL OUT 音量 SNDSTOP		サウンド全体の音量を設定する / サブプログラムからの設定は無視される サウンド全体の音量を取得する すべての音の発生を停止
TALK 文字列		与えた文字列に従って音声合成 ひらがな/カタカナ/句読点などの文字列 又は \$ から始まる音楽文字列 a b c d e f g h i j k m n o p q r s t u v w x y z by ch cl(つはねる音) dy gy hy ky my ny py sh ts #0.1秒無音) <で始まり> で終わる制御文字列 <S数値> スピード: 0(速)～1024(遅) 標準:128 <P数値> ピッチ: 0(高)～1024(低) 標準:128 <V数値> 音量: 0(小)～128(大) 標準:128
TALKSTOP		音声合成を停止
WAVSET 楽器番号,アタック,ディケイ,サステイン,リリース,定義文字列[基準音程]		楽器音の波形設定 / 楽器番号:224～255、各パラメータ: 0～127 定義文字列 00～"Ff" を8ビットの1サンプルとして連続して記述(128が減算され -128～127 の値として扱われる) 16,32,64,128,2 基準音程 定義した波形の音程のノート番号を指定。省略時は69("O4A"オクターブ4のラ)
WAVSETA 楽器番号,アタック,ディケイ,サステイン,リリース,配列[基準音程],波形先頭,波形終端]		楽器音の波形設定 / 3号互換モード
WAVSETA 楽器番号,アタック,ディケイ,サステイン,リリース,配列,基準音程,ループ先頭,ループ終端,サ		楽器音の波形設定 サウンドのメモリ容量が許す限りの長さの波形を収容可能(サンプル周波数 48000Hz で 80秒程度)
BGMCHK [トラック番号] OUT 変数		指定したトラックの音楽の演奏状態を返す
CHKMML MML文字列 OUT 変数		MML文字列の内容が正常にMMLとして解釈できるかどうかを返す / -1:正常, 0:エラー(解釈エラーになった文字列位置)
TALKCHK OUT 変数		音声合成の状態調査 / FALSE=停止中、TRUE=再生中

高度な演算

高度な演算を高速に行う命令

BIQUAD 出力配列(OUT),入力配列(IN),フィルタ係数配列(FP)

BQPARAM フィルタ係数配列,フィルタ種別,サンプリング周波数,カットオフ周波数,Q値または帯域幅

FFT 出力実数部配列,出力虚数部配列,入力実数部配列,入力虚数部配列[,窓関数値配列]

IFFT 出力実数部配列,出力虚数部配列,入力実数部配列,入力虚数部配列

FTWFN 配列,窓関数種別

ARYOP 演算タイプ,結果格納配列,パラメータ1,パラメータ2[,パラメータ3]

ソープログラムからエディタ内のテキストを参照・編集するための命令

PRGEDIT ソープログラムスロット[,行番号]

行番号 = PRGEDIT()

PRGEDIT OUT [ソープログラムスロット,]行番号

PRGGET\$([自動シークフラグ])

PRGSEEK [移動方向]

値 = PRGSEEK([移動方向])

PRGSET 文字列

PRGINS 文字列[,フラグ]

PRGDEL [削除行数]

PRGSIZE([ソープログラムスロット[,取得する値のタイプ]])

PRGNAME\$([ソープログラムスロット])

サブプログラム

メイン・サブのプログラム間で状態やデータを交換する命令

サブプログラム内でのみ意味を持つ命令。

XSUBSCREEN 表示オフセットX,表示オフセットY,幅,高さ

XSUBSCREEN 0

XSUBSCREEN OUT 表示オフセットX,表示オフセットY,幅,高さ

ENVSTAT 状態

ステータス = ENVSTAT()

ENVSTAT OUT ステータス,実行スロット,編集スロット

ENVSTAT OUT ステータス,エラー番号,エラー発生引数番号,エラー発生スロット,エラー発生行,エラー発生文字位置

ENVLOAD 0,メインID,サブID

ENVLOAD 1,メインID,サブID

ENVSAVE 0,サブID,メインID

ENVSAVE 1,サブID,メインID

ENVINPUT\$([履歴番号])

タイプ = ENVTYPE()

ENVTYPE OUT タイプ,パラメーター

ENVFOCUS フラグ

フラグ = ENVFOCUS()

ENVPROJECT

PUSHKEY 文字列

HELPIFNO OUT 命令名文字列,数値配列

HELPIFNO 命令名文字列 OUT 数値配列

HELPGET 命令名文字列,ページ番号 OUT 行文字列配列,タグ配列,属性数値配列,コード例文字列

双二次フィルタを適用する

双二次フィルタのフィルタ係数を計算する

複素数配列に対してフーリエ変換を行う

複素数配列に対してフーリエ逆変換を行う

窓関数値配列を得る

配列内要素の一括演算を行う

演算タイプ

0 #AOPADD 加算(p1+p2)

1 #AOPSUB 減算(p1-p2)

2 #AOPMUL 乗算(p1*p2)

3 #AOPDIV 除算(p1/p2)

4 #AOPMAD 積和(p1*p2+p3)

5 #AOPLIP 線形補間(p1*p3+p2*(1-p3))

6 #AOPCLP クランプ(p1の値をp2<=x<=p3の範囲内に丸める)

エディタ上のソースコード操作を開始する

操作中のソースコード情報を取得する

カレント行1行分の文字列を取得する

カレント行を1行前後に移動させる

カレント行を1行前後に移動させる / 移動できた場合は1を、先頭行または最終行から先に移動しようとしてできなかった場合は0を返す

カレント行の内容を指定文字列に置き換える / PRGGET\$が空文字列を返す場合は行追加

カレント行へ文字列を1行挿入する / フラグ: 挿入先 1: 行の後方, 0: 行の前方

カレント行を削除する / 省略時1行、マイナス値を入れた場合全体を削除

ソースコードの行数を取得する / タイプ: 0: 行数(デフォルト), 1: 文字数, 2: 空文字数 / ソープログラムスロット省略した場合現在実行中

プログラムのファイル名を取得する / LOAD/SAVE命令で指定したファイル名が返る

サブプログラム画面の表示サイズを指定する

サブプログラム画面の表示をやめる

サブプログラム画面の表示オフセットを取得する

メインインタプリタの状況を設定する / 0のみ指定可能 現在の状態が3,4の場合は設定されない

メインインタプリタの状況を取得する

メインインタプリタの状況を取得する

メインインタプリタの状況を取得する

ダイレクトモードで待機中

実行中断中

エラーで実行停止

プログラム実行中

プログラム編集集中

メインインタプリタ上のプログラムスロットの内容をサブインタプリタ側のプログラムスロットにコピーする / ID は プログラムスロット

メインインタプリタ上のグラフィックページデータをサブインタプリタ側にコピーする / ID は グラフィックページ

サブインタプリタ上のプログラムスロットの内容をメインインタプリタ側にコピーする / ID は プログラムスロット

サブインタプリタ上のグラフィックページデータをメインインタプリタ側にコピーする / ID は グラフィックページ

メインインタプリタのコンソール入力行、もしくは入力履歴の内容を取得する

/ 履歴番号: 0: 入力中のコンソール行(省略時), 1:直近 ~32: 過去

現在実行中のプログラムがどのインタプリタで動いているかを取得する

/タイプ: 0: メインインタプリタ, 1: サブインタプリタ, 2: UIインタプリタ

/ 起動時に設定されたパラメーター。ユーザーが作成できるプログラムでは常に-1が返る

フォーカスを取得・譲渡する / 1でフォーカスを取得、0でフォーカスを譲渡

フォーカスの所持状況を取得する / 1で持っている、0で持っていない

サブインタプリタの作業用プロジェクトをメインインタプリタの作業用プロジェクトに同期させる

指定文字列をキーボード入力した事にする / 最大64文字

カーソル位置にあるヘルプデータの情報 / 内部実装用命令。使用禁止 / 命令名がない場合は「DEFAULTHELPKEY」が返る

命令名に対してのヘルプデータの情報 / 内部実装用命令。使用禁止

ヘルプデータの取得 / 内部実装用命令。使用禁止

画面仕様	
GRP0 - GRP5	
テキストスクリーン(通常文字)	GRP5 変更不可
テキストスクリーン(ユーザー定義文字)	テキストTPAGEで変更可
スプライト	スプライトSPPAGEで変更可
GRP5 が 旧来の GRPF (フォント定義)	
GRP4 が 旧来の SPRITE と BG のリソース	
テキストスクリーン(ユーザー定義文字) が旧来の BG、画面ごとに GRP 参照先を変えられる	
スプライトは旧来と同じ、スプライト単位で GRP 参照先を変えられる	
ユーザー定義フォントは文字コード&HE800から&HF7FFまでの4096文字	
ただし BG として使う場合 T系の命令で 旧来の BG系 と同じように使え、BGタイル指定も 0 から可能	