

キーワード	種別	読み	書式	概要	利用例
'	コマンド	シングルクォーテーション	'コメント文	コメント文記述 コメント文を記述する。REMコマンドの省略形。	10 'sample 01
-	二項演算子	マイナス	数値1 - 数値2	減算演算子 数値1から数値2を引きその値を返す。	A=5-1
!	単項演算子	感嘆符、びっくりマーク	!数値	論理否定演算子 数値が0なら1、0以外なら0を返す。	A=!(5-1)
!=	二項演算子	ノットイコール	数値1 != 数値2	不一致判定 数値1と数値2が不一致なら1、一致なら0を返す。	IF X!=10 Y=30 A=5!=B
"	補助記号	ダブルクォーテーション	"文字列"	文字列、ラベルの定義 文字列、ラベルはダブルクォーテーションで囲んで記述する。	10 "LOOP" 20 PRINT "Hello,World" 30 GOTO "LOOP"
#	補助記号	シャープ	1)PRINT #桁,文字列(または数値、文字列関数)[;文字列;...][;] 2)数値1 # 数値2	1)PRINT文内数値桁指定 PRINT文内の指定以降の数値の出力の桁数を指定する。 数値が正の場合、空白文字で桁合わせ、数値が負の場合、'0'文字で桁合わせする。 2)剰余演算子(%と同じ) 数値1 を 数値2 で割った余りを返す。 ※%の利用を推奨、tinyBASICとの互換性を保つために用意	PRINT 1;" ";22;" ";33 1 22 33 PRINT #3,1;" ";22;" ";33 1, 22, 33 PRINT #-3,1;" ";22;" ";33 001,022,033
\$	補助記号	ドル、ダラー	\$16進数	16進数表記記号 数値定数を16進数で記述する(1から4桁)。	A=\$1234 B=\$ABCD
%	二項演算子	パーセント	数値1 % 数値2	剰余演算子 数値1 を 数値2 で割った余りを返す。	?5%3 2
&	二項演算子	アンパサンド	数値1 & 数値2	ビット単位のAND演算 数値1と数値2のビット単位のAND演算を行いその対を返す。	IF A&1 ?"キスウ" ?bin\$(`1100&`0100,4) 0100
(補助記号	左括弧	1) 関数() 2) (式)	左括弧 1)関数の引数の指定する。 2)式の優先計算を行う。	A=ABS(B) B=(1+3+5)*(2+4+8)/(3+6+9)
)	補助記号	右括弧	関数() (式)	右括弧 関数の引数の指定または、式の計算優先度の指定	A=ABS(B) B=(1+3+5)*(2+4+8)/(3+6+9)
*	二項演算子	アスタリスク	数値1 * 数値2	乗算演算子 数値1 と 数値2の掛け算を行い、その値を返す。	A=B*10 ?9*9
,	補助記号	カンマ、コンマ	コマンド 引数1,引数2	区切り記号、項目の継続 コマンドや関数の引数の区切り、項目の継続指定 PRINT文では';'と同じ連結・改行抑制動作として機能する。	10 CLS:LOCATE 0,5:A=5:B=10 20 PRINT "(A,B)=(30 PRINT A,";",B,")"
/	二項演算子	スラッシュ	数値1 / 数値2	除算演算子 数値1 を 数値2で割った値を返す。	A=8/2
:	補助記号	コロ	コマンド文1:コマンド文2	コマンド区切り 1行に複数のコマンド文を記述する際の区切りとして利用する。	10 X=5:Y=5:CLS 20 LOCATE X,y:?"Hello"
;	補助記号	セミコロ	文字列1;文字列2;	PRINT文中の文字列の連結、改行抑制(行の継続)	10 CLS:LOCATE 0,5:A=5:B=10 20 PRINT "(A,B)=(30 PRINT A;";";B;")"
?	コマンド	疑問符、クエスチョン、ハテナ	? ? [#桁,]文字列(または数値、文字列関数)[;文字列;...][;] ※[]は省略可能を示す。 ※..は以下同様に繰り返し可を示す。	文字列・数値の表示 PRINTの省略形である。機能はPRINTに同じ。	10 H=20:M=30:S=0 20 ?"TIME="; 30 ?#-2,H;";";M;";";S RUN TIME=20:30:00
@	補助記号	アットマーク	@(配列番号)	配列記述子 配列変数を利用するための記号。配列番号は0〜99の指定が可能	@(0)=@(1)+@(2)+@(3) PRINT "@(0)=";@(0)
^	二項演算子	キャレット、ハット	数値1 ^ 数値2	ビット単位のXOR演算子(排他的論理和演算子) 数値1と数値2の各ビットのXOR(ビットが同値なら0、異なれば1)を行い、その値を返す。	A=`1111^^0101 ?BIN\$(A,4) 1010
`	補助記号	バッククォート	`2進数	2進数表記記号 数値定数を2進数で記述する(1から16桁)	A=`11110000 B=`1000000011111111
	二項演算子	パイプ、縦棒	数値1 数値2	ビット単位のOR演算 数値1と数値2の各ビットごとのORを行い、その値を返す。	A=B C
~	単項演算子	チルド、チルダー、によろ	~数値	ビット単位の反転演算 数値のビット単位の反転を行い、その値を返す。	A=~ 1010101010101010 ?BIN\$(A,16)
+	二項演算子	プラス	数値1 + 数値2	加算演算子 数値1と数値2の足し算を行い、その値を返す。	A=5+10 B=A+(1+5)*2
<	二項演算子	小なり、レスザン、レフトアングル	数値1 < 数値2	小なり判定(未満判定) 数値1が数値2よりも小さい場合は1、そうでない場合0を返す。	IF X<50 X=X+1 A=10<20
<<	二項演算子	小なり小なり、左シフト	数値1 << 数値2	左シフト演算子 数値1を数値2指定のビット数分左シフトし、その値を返す。	?BIN\$(`000000000000001<<5,16) A=B<<1
<=	二項演算子	小なりイコール	数値1 <= 数値2	小なりイコール判定(以下判定) 数値1が数値2以下の場合は1、そうでない場合は0を返す。	A=10<=20 IF X <= 49 X=X+1
<>	二項演算子	ノットイコール	数値1 <> 数値2	不一致判定 数値1と数値2が等しくない場合は1、等しい場合は0を返す。	IF X<>5 ? "Not 5" Y=Y+(S<>10)+(T<>20)
=	二項演算子、 代入演算子	イコール	1)数値1 = 数値2 2)変数 = 値	1)一致判定 数値1と数値2が等しい場合は1、等しくない場合は0を返す。 2)代入演算子 変数または、配列変数に値を代入する。	1) IF X=100 ? "GOAL!" 2) A=135/5
>	二項演算子	大なり	数値1 > 数値2	大なり判定(より大きい判定) 数値1が数値2よりも大きい場合は1、そうでない場合0を返す。	IF X > 50 X=X-1 A=B>50
>=	二項演算子	大なりイコール	数値1 >= 数値2	大なりイコール判定(以上判定) 数値1が数値2以上の場合は1、そうでない場合0を返す。	IF X >= 51 X=X-1 A=B>=51

キーワード	種別	読み	書式	概要	利用例
>>	二項演算子	大なり大なり、右シフト	数値1 >> 数値2	右シフト演算子 数値1を数値2指定のビット数分右シフトし、その値を返す。	?BIN\$(`1000000000000000>>5,16) A=B>>1
ABS	関数	エー・ビー・エス	ABS(数値)	絶対値の取得 指定した数値の絶対値を返す。	?ABS(-123) 123
ANA	関数	アナ、エー・エヌ・エー	ANA(ピン番号)	GPIO アナログ入力 指定ピンのアナログ入力値を読み取り、その値を返す。 アナログ入力を行う場合は事前にGPIOコマンドによる機能設定(アナログ入力)が必要。	GPIO PB1,ANALOG ?ANA(PB1)
ANALOG	定数	アナログ	ANALOG	GPIOモード設定定数 アナログ入力(=3) GPIOコマンドのモード指定に利用	GPIO PB1,ANALOG ?ANA(PB1)
AND	二項演算子	アンド	数値1 AND 数値2	論理AND演算 数値1、数値2がともに真(0以外) なら1、そうでない場合0を返す。	IF (X>0) AND (X<100) S=S+1 A=5 AND 10
ARRAY	定数	アレイ	ARRAY	配列変数領域先頭アドレス(=\$1AA0) 配列変数領域(@0)~@99) 200バイトの先頭アドレスを返す。	10 POKE ARRAY+50*2,123,0 20 ?@50
ASC	関数	エー・エス・シー	ASC(変数 [,位置]) ASC(文字列 [,位置])	文字列コード取得 指定した変数に格納されている文字列、または指定した文字列の先頭または指定位置の文字コードを返す。	10 ?ASC("A") 20 S="こんにちは":?HEX\$(ASC(S,3),4)
ATTR	コマンド	エー・ティ・ティ・アール	ATTR 属性	文字の表示属性の設定 シリアルコンソール画面の文字表示属性を設定する。 属性 0:通常、1: 下線、2:反転 3:部リンク、4:ボールド	ATTR 1
BANK	コマンド	バンク	BANK プログラム番号	PRG2参照領域バンク切り替え プログラム保存番号0~5のうちの指定したプログラム番号をPRG2で参照するプログラム保存領域として設定	BANK 5
BIN\$	文字列関数	ピン・ダラー	BIN\$(数値[,桁数]) ※[]は省略可能を示す。	数値から2進数文字列への変換 PRINT文中で利用し、数値を2進数に変換して出力する。桁を指定した場合、桁数に満たない場合0で埋める。	?BIN\$(123,16) 000000001111011 OK
BITMAP	コマンド	ビットマップ	BITMAP 横座標,縦座標,仮想アドレス,インデックス,幅,高さ [,倍率 [,色 [,モード]]]	ビットマップ画像の描画 指定した座標にビットマップ画像を描画する。仮想アドレスにはビットマップ画像の格納アドレスを指定する。 インデックスを指定することで、連続的に格納している画像を切り替えて表示することが出来る。幅、高さはビットマップ画像のサイズを指定する。倍率が表示する倍率を指定する。 ビットマップ画像は、1ドットを1ビットの 0or 1の連続データで定義する。 TFT版では1ドットに16ビットを割り当てた16ビット色の利用も可能である。 16ビット色を利用する場合、引数のモードに1を指定する。	10 CLS 20 POKE MEM+0,\$10,\$40,\$48,\$90,\$5F,\$D0,\$77,\$70 30 POKE MEM+8,\$3F,\$E0,\$1F,\$C0,\$10,\$40,\$20,\$20 40 POKE MEM+16,\$10,\$40,\$08,\$80,\$1F,\$C0,\$37,\$60 50 POKE MEM+24,\$7F,\$F0,\$5F,\$D0,\$50,\$50,\$0D,\$80 60 FOR A=0 TO 20 70 BITMAP A,0,MEM,A%2,12,8,4 80 WAIT 200 90 NEXT A
BLOAD	コマンド	ビー・ロード	BLOAD "ファイル名", 格納アドレス, バイト数 [,ファイル内位置]	SDカードからバイナリデータ読込 SDカードから指定したファイル内の指定位置のデータを、指定したバイト数分指定したアドレスに読み込む。 ファイル内位置を指定しない場合は、先頭から読み込む。	BLOAD "IMAGE.DAT",MEM,256,1024
BSAVE	コマンド	ビー・セーブ	BSAVE "ファイル名", 格納アドレス, バイト数 [,ファイル内位置]	SDカードへのバイナリデータ保存 指定アドレスに格納されいてるバイナリデータを指定したバイト数分、SDカードの指定ファイルの指定ファイル位置に保存する。ファイル内位置の指定がない場合は、先頭から保存する。 ファイル内位置は既存ファイルのサイズ以上の位置の指定は出来ない。書き込みエラーとなる。	BSAVE "IMAGE.DAT",MEM,256,1024
BYTE	関数	バイト	BYTE(変数) BYTE(文字列)	文字列のバイト数の取得 文字列、変数で参照してる文字列のバイト数をカウントし、その値を返す。全角文字1文字は2バイトとしてカウントする。文字数を取得にはLEN関数を利用のこと。	10 ?BYTE("1あ2い34う5678") 20 A="1あ2い34う5678" 30 ?BYTE(A)
CAT	コマンド	キャット	CAT "ファイル名"	ファイルの内容表示 SDカード内の指定したテキストファイルの内容を画面に表示する。 表示テキストには先頭にコメントコマンド(')が付加される。	CAT "/SRC/SAMPLE.BAS"
CH	定数	シー・エッチ	CH	コンソールテキスト画面 縦文字数 実際の値は利用するデバイス(シリアル、NTSC、 OLED、TFT)、設定に依存	Y=Y+1:IF Y>=CH Y=CH-1
CHR\$	文字列関数	シー・エッチ・アール・ダラー	CHR\$(文字コード[,文字コード,...,文字コード]) ※[]は省略可能を示す。 ※..は以下同様に繰り返し可を示す。	文字コード・文字列変換 PRINT文で利用し、文字コードを文字列に変換して出力する。 文字コードは複数指定が可能、全角文字にも対応	?CHR\$(65,66,67,68) ABCD OK
CIRCLE	コマンド	サークル	CIRCLE 横中心座標,縦中心座標,半径,色,モード	円の描画 指定した中心座標(横中心座標,縦中心座標)を起点に指定した半径の円を色で描画する。モードに0を指定した場合は、線描画、1を指定した場合は塗りつぶし描画を行う。	CIRCLE 50,50,40,1,0
CLS	コマンド	シー・エル・エス クリア・スクリーン	CLS [モード] ※[]は省略可能を示す。	画面クリア コンソール画面の表示内容を消去し、カーソル位置をホームポジションに移動する(モードに0を指定した場合も同機能)。モードに1を指定した場合は、デバイス画面(OLED、TFT、NTSC)の表示を消去する。	CLS
CLT	コマンド	シー・エル・ティ クリア・ティック	CLT	システム起動経過時間のリセット TICK関数で取得できるシステム起動経過時間をリセット(=0)する。	CLT:?TICK() 0
CLV	コマンド	シー・エル・バイ、 クリア・バリュー	CLV	変数領域の初期化 変数領域(変数、配列)の初期化する。LRUN コマンドにてプログラムをロードして実行する場合、変数領域は初期化しないため、明示的に初期化する場合は本コマンドを用いる。	CLV
COLOR	コマンド	カラー	COLOR 文字色[,背景色]	文字の前景色、背景色の設定 コンソール画面の文字色、背景色を設定する(シリアルコンソール、TFT画面でのみ有効)。	COLOR 7,4:CLS
CONFIG	コマンド	コンフィーグ	CONFIG 項目番号, 設定値 [,設定値2, 設定値3] 1)CONFIG 0,垂直同期補正[,横補正, 縦補正] 2)CONFIG 1,レイアウト指定値 3)CONFIG 2,プログラム番号	システム環境設定 項目番号で指定した機器の設定を行う。設定はSAVECONFIGコマンドで保存可能。 1)NTSC信号補正 垂直同期補正 -2~2、横補正・縦補正は位置補正 -15 ~ 15 2)キーボード設定 キーボードレイアウト指定値 0:日本語(デフォルト) 1:US 3)起動時プログラム自動起動設定 プログラム番号 0~5、-1で自動起動解除	CONFIG 0,2,10,10 CONFIG 1,0 CONFIG 2,0
CONSOLE	コマンド	コンソール	CONSOLE モード	コンソール画面切替 シリアルターミナル、デバイス画面間で利用するコンソール画面の切替を行う。 モードの指定が0の場合、デバイス画面をコンソール画面として利用する。モードが1の場合、シリアルポート(USBポートまたは、ポート PA9、PA10)をコンソール画面として利用する。	コンソール画面をシリアルターミナルに切り替える CONSOLE 1 コンソール画面をデバイス画面(OLED、TFT、NTSC)に切り替える CONSOLE 0
CSCROLL	コマンド	シー・スクロール	CSCROLL x1,y1,x2,y2,方向	テキスト画面のスクロール 左上座標x1、y1、右下座標x2、y2で指定した矩形領域の内容を指定した方向にスクロールする。NTSC、OLED、TFT画面のコンソール画面で利用可能。シリアルコンソールは未対応。	10 FOR I=1 TO CW 20 CSCROLL 0,0,CW-1,CH-1,LEFT:WAIT 100 30 NEXT I
CW	定数	シー・ダブル	CW	コンソールテキスト画面 横文字数 実際の値は利用するデバイス(シリアル、NTSC、 OLED、TFT)、設定に依存	X=X+1:IF X>=CW X=CW-1

キーワード	種別	読み	書式	概要	利用例
DATE	コマンド	デイト	DATE	現在時刻の表示 内蔵RTCから現在の時刻を読み、その情報を画面に表示する。	DATE 2018/11/24 [Sat] 12:00:03
DELETE	コマンド	デリート	DELETE 行番号 DELETE 先頭行番号, 末尾行番号	プログラムの指定行削除 プログラム内の指定した行、指定した範囲(先頭行番号、末尾行番号)の行を削除する。	DELETE 200,250
DMP\$	文字列関数	ディー・エム・ピー・ダラー	DMP\$(数値[,小数点桁数[,整数部桁数]]) ※[]は省略可能を示す。	数値から小数点付き数値文字列への変換 PRINT文中で利用し、指定した数値を小数点付きの文字列数値に変換して出力する。引数の小数点桁数(省略時は2)をnとした場合、数値÷10 ⁿ の計算を行い、小数点以下の数値を含めて出力する。整数部桁数を指定した場合、整数部の桁を固定表示する、指定桁数に満たない場合は空白で補完する。	?DMP\$(3141,3) 3.141 ?"[";DMP\$(3141,3,3);"]" [3.141]
DOWN	定数	ダウン	DOWN	スクロール方向指定 DOWN(=1) CSCROLL、GSCROLLコマンドのスクロール方向を指定する引数に利用する	CSCROLL 0,0,CW-1,CH-1,DOWN
DWBMP	コマンド	ディ・ダブル・ピー・エム・ピー、 ドゥロウ・ビットマップ	DWBMP "ファイル名", 横座標, 縦座標 [,横座標2,縦座標2,幅,高さ[,色指定]]	ビットマップファイルの表示 SDカード内のビットマップファイルの指定領域(横座標2,縦座標2,幅,高さで指定、省略時は画像サイズ)の画像を切り出して、指定した座標(横座標, 縦座標)に表示する。 NTSC、OLEDディスプレイの場合、ビットマップファイルはモノクロ(白黒2値)画像のみ利用可能。TFTデスプレイの場合、24ビット色ビットマップファイルも利用出来る。	10 CLS 20 DWBMP "TT.BMP",0,0 30 DWBMP "TT.BMP",16,66,0,32,96,32 40 GOTO 40
EEPFORMAT	コマンド	イー・イー・ピー・フォーマット	EEPFORMAT	仮想EEPROMのフォーマット 仮想EEPROMのフォーマットして初期化を行う。SAVECONFIGで保存した内容及、EEPWRITEにて書き込んだデータは初期化される。	EEPFORMAT
EEPREAD	関数	イー・イー・ピー・リード	EEPREAD(アドレス)	仮想EEPROMデータ読み込み 仮想EEPROMの指定したアドレスから2バイトデータを取得する。読み出し単位は2バイトである。仮想EEPROMに初めて利用する場合は、EPFORMAT コマンドでの初期化を行う必要がある。	EEPWRITE 1234,123 ?EEPREAD(1234) 123
EEPWRITE	コマンド	イー・イー・ピー・ライト	EEPWRITE アドレス, データ	仮想EEPROMデータ書き込み 仮想EEPROMの指定したアドレスにデータを書き込む。書き込み単位は2バイトである。仮想EEPROMに初めてデータ書き込みを行う場合は事前に EEPFORMAT コマンドで初期化をしておく必要である。	EEPWRITE 1234, 123 ?EEPREAD(1234) 123
ELSE	補助制御コマンド	エルス	IF 条件 処理1 ELSE 処理2	IF文の条件が偽場合の実行文の記述 条件が偽の場合、ELSE以降の処理2を実行する。	IF X&1 ?"キスウ" ELSE ?"グウスウ"
END	制御コマンド	エンド	END	プログラムの終了 プログラムを終了し、編集可能状態に復帰する。	END
ERASE	コマンド	イレイス	ERASE プログラム番号 ERASE 開始プログラム番号, 終了プログラム番号	内部フラッシュメモリのプログラム削除 内部フラッシュメモリに保存されているプログラム(プログラム番号0～5)を削除する。削除対象のプログラム番号またはプログラム番号の範囲(開始プログラム番号、終了プログラム番号)を指定する。	ERASE 0,5
EXPORT	コマンド	エクスポート	EXPORT EXPORT 対象番号 EXPORT 開始番号,終了番号	フラッシュメモリの内容のエクスポート 内のフラッシュメモリに保存されているプログラム(プログラム番号0～5)をコンソール画面に出力する。引数を指定しない場合は、すべてのプログラムを出力する。引数を指定した場合、該当する番号のプログラムを出	EXPORT 1,2
FILES	コマンド	ファイルズ	FILES FILES 開始プログラム番号 FILES 開始プログラム番号, 終了プログラム番号 FILES "ファイルパス"	プログラム・ファイル一覧表示 マイコン内のフラッシュメモリに保存されているプログラムまたは、SDカード内に保存されているファイルを一覧を表示する。引数を指定しない場合は、内部フラッシュメモリ内のプログラム一覧を表示する。ファイルパス(ワイルドカード指定可能)を指定した場合は、そのファイルにマッチするファイルを一覧表示する。	FILES FILES 0,3 FILES "*.bas"
FNT	定数	フォント、エフ・エヌ・ティー	FNT	スクリーンフォント格納アドレス(=\$2FA0) NTSC、OLED、TFT画面で利用する半角フォント256文字分格納アドレスの先頭アドレス	10 C=ASC("A") 20 FOR I=0 TO 7 30 ?BIN\$(PEEK(FNT+C*8+I),8) 40 NEXT I
FOR	制御コマンド	フォー	FOR 変数=初期値 TO 最終値 [STEP 変化量] 処理 NEXT 変数 ※[]は省略可能を示す。	繰り返し文 FOR～NEXT間で、変数を初期値から最終値まで変化させ、処理を繰り返す。STEP文により、変数の変化量の指定が可能(省略時は1)。	10 V=0 20 FOR I=1 TO 10 STEP 1 30 V=V+I 40 NEXT I 50 PRINT "V=";V
FREE	関数	フリー	FREE()	プログラム領域の残量取得 プログラム領域の残りバイト数を返す。	プログラムサイズを調べる ?4095-FREE()
FWRITE	コマンド	エフ・ライト	FWITE 仮想アドレス, データ	プログラム保存領域への直接書き込み 指定したPRG2～PRG2+4095の範囲の仮想アドレスに2バイト単位でデータを書き込む。仮想アドレスは必ず偶数アドレスを指定する。書き込みを行うには、事前にERASEコマンドで対象のバンクを消去する必要がある。	10 BANK 5 20 ERASE 5 30 FWRITE PRG2,\$1234 40 ?HEX\$(PEEK(PRG2+1),2);HEX\$(PEEK(PRG2),2)
GCLS	コマンド	ジー・シー・エル・エス	GCLS	グラフィックデバイス画面のクリア NTSC版、TFT版、OLED版のグラフィックデバイス画面のクリアを行う(CLS 1と同じ)。 グラフィックデバイス画面をコンソールとして利用している場合は、CLSと同じ動作となる。 NTSC版のシリアルコンソールモードでは、何も処理を行わない。	GCLS
GCOLOR	コマンド	ジー・カラー	GCOLOR 前景色[,背景色]	文字描画の前景色、背景色の設定 グラフィック画面の文字色、背景色を設定する。設定は、GCLS、GPRINT、BITMAP(単色表示時)、KANJIコマンドによる描画に有効である。NTSC画面では何もしない。	GCOLOR 7,4
GETDATE	コマンド	ゲット・デイト	GETDATE 年格納変数, 月格納変数, 日格納変数, 曜日格納変数	日付の取得 引数に指定した変数に、現在の年、月、日、曜日(0:月～6:日)を返す。	GETDATE A,B,C,D ?#-2,A;"/";B;"/";C 2018/11/24
GETS	関数	ゲット・エス	1)GETS(仮想アドレス[,最大長さ]) 2)GETS(仮想アドレス,最大長さ,モード[,参照元仮想アドレス]) ※[]は省略可能を示す。	1)文字列対話入力 モード指定の省略または、モードに0を指定した場合、文字列の対話入力を行い、そのデータを仮想アドレスに格納し、そのアドレスを関数値として返す。最大長さをしていない場合は32文字まで入力可能。 2)仮想アドレスからのコピー モードに1を指定した場合、参照元仮想アドレスからバイナリデータとして読み出しを行い、文字列として仮想アドレスに格納する。参照元仮想アドレスの指定がない場合は、領域のみ確保する。	10 ?"Your name="; 20 S=GETS(MEM,20) 30 ?"Hello,";STR\$(S)
GETTIME	コマンド	ゲット・タイム	GETTIME 時格納変数 , 分格納変数 , 秒格納変数	時刻の取得 引数に指定した変数に、現在の時、分、日、秒を返す。	GETTIME A,B,C ?#-2,A;":";B;":";C 12:00:03
GH	定数	ジー・エッチ	GH	グラフィック画面 縦ドット数 実際の値は利用するデバイス(シリアル、NTSC、 OLED、TFT)、設定に依存	RECT 0,0,CW-1,CH-1,1,1
GINP	関数	ギー・イン・ピー	GINP(横座標,縦座標,高さ,幅,色)	指定矩形内のピクセル有無のチェック 指定した領域内の指定した色のピクセルが存在するをチェックし、存在する場合は1、そうでない場合は0を返	?GINP(10,10,30,30,1)

キーワード	種別	読み	書式	概要	利用例
GOSUB	制御コマンド	ゴー・サブ	GOSUB 行番号 GOSUB "ラベル"	サブルーチン呼び出し 指定した行番号、またはラベル付きの行にジャンプし、RETURNで復帰する。	GOSUB 5000 GOSUB "Draw"
GOTO	制御コマンド	ゴー・ツー	GOTO 行番号 GOTO "ラベル"	ジャンプ 指定した行またはラベル付きの行にジャンプする。	GOTO 5000 GOTO "LOOP"
GPEEK	関数	ジー・ピーク	GPEEK(横座標,縦座標)	指定座標のドット色コード参照 デバイスグラフィック画面(NTSC、OLED、TFT)上の指定座標ドットの色コードをする。指定座標が範囲外の場合は0を返す。	?GPEEK(50,50)
GPIO	コマンド	ジー・ピー・アイ・オー	GPIO ピン番号,モード	GPIO機能設定 ボード上の指定したピン番号の入出力機能の設定を行う。 ピン番号の指定には、0～34の数値または、定数PA0～PC15が可能である。 モードには、OUTPUT_OD、OUTPUT、INPUT_FL、INPUT_PU、INPUT_PD、ANALOG、PWMの指定が可能である。 定数を利用した場合、GPIOの省略が可能である(下記は等価) GPIO PA0,OUTPUT PA0,OUTPUT	10 GPIO PC13,OUTPUT 20 S=HIGH 30 OUT PC13,S 40 WAIT 500 50 S=!S 60 GOTO 30
GPRINT	コマンド	ジー・プリント	GPRINT 横座標,縦座標,[#桁,]文字列(または数値、文字列関数)[;文字列;...][;] ※[]は省略可能を示す。	文字列の描画 指定したグラフィック座標にグラフィックとして文字列を描画する。 横座標、縦座標以降の記述及び出力結果は PRINT 文と同等である。ただし';'による改行抑制は出来ない	GPRINT 10,10,"Hello,World"
GRAM	定数	ジーラム	GRAM	グラフィック画面用VRAM先頭アドレス(=\$37A0) NTSC、OLED画面のグラフィック表示用領域の先頭アドレス、TFT版では利用出来ない。 ※領域のバイト数ははGW/8×GHとなる。	画面のグラフィック表示内容をファイルに保存する BSAVE "DISP.IMG",GRAM,GW/8*GH 保存した上記ファイルをロードして画面に表示する BLOAD "DISP.IMG",GRAM,GW/8*GH
GSCROLL	コマンド	ジー・スクロール	GSCROLL 横座標1,縦座標1,横座標2,縦座標2,方向	グラフィックスクロール 画面上の指定範囲表示内容を1ドット単位でスクロールする。方向にはUP(0):上、DOWN(1):下、RIGHT(2):右、LEFT(3):左の定数またはカッコ内の数値の指定が可能。 TFT版では非対応。NTSC版では、横座標1は8の倍数、横座標2は8の倍数-1でのみ指定可能(バイト境界指定のみ有効)。	GSCROLL 0, 0,GW-1,GH-1 ,LEFT
GW	定数	ジー・ダブル	GW	グラフィック画面 横ドット数 実際の値は利用するデバイス(シリアル、NTSC、 OLED、TFT)、設定に依存	RECT 0,0,CW-1,CH-1,1,1
HEX\$	文字列関数	ヘックス・ダラー	HEX\$(数値[,桁数]) ※[]は省略可能を示す。	数値から16進数文字列への変換 プリント文中で利用し、数値を16進数に変換して出力する。桁を指定した場合、桁数に満たない場合0で埋める。	?HEX\$(123,4) 007B OK
HIGH	定数	ハイ	HIGH	デジタル信号HIGH(=1) デジタル入出力のロジックレベルを表す定数	OUT PC13,HIGH
I2CCLK	コマンド	アイ・ツー・シー・クロック	I2CCLK バス速度	I2Cバス速度設定 I2Cバスのクロック速度を設定する。バス速度には100(=100kHz)または、400(=400kHz)の指定が可能。 デフォルト値は、400(400kHz)。	I2CCLK 100
I2CR	関数	アイ・ツー・シー・アール(リード)	I2CR(デバイスアドレス,コマンドアドレス,コマンド長,受信データアドレス,データ長)	I2Cバス データ受信 I2Cスレーブデバイスからデータを受信する。	100 POKE MEM+0,\$00,\$00 110 POKE MEM+2,64,65,66,67 120 R=I2CR(\$50,MEM,2,MEM+6,4)
I2CW	関数	アイ・ツー・シー・ダブルユ(ライト)	I2CW(デバイスアドレス,コマンドアドレス,コマンド長,データアドレス,データ長)	I2Cバス データ送信 I2Cスレーブデバイスにデータを送信する。	100 POKE MEM+0,\$00,\$00 110 POKE MEM+2,64,65,66,67 120 R=I2CW(\$50,MEM,2,MEM+2,4)
IF	制御コマンド	イフ	IF 条件 処理1 IF 条件 処理1 ELSE 処理2	条件付き実行 条件が真の場合、処理1を実行する。ELSE文がある場合、偽なら処理2を実行する。	IF X>0 ?"Xハ0イショウ" IF X&1 ?"キスウ" ELSE ?"ゲウスウ"
IN	関数	イン	IN(ピン番号)	GPIO デジタル入力 指定ピンの入力値を読み取り、その値を返す。 事前にGPIOコマンドによる機能設定(入力設定)が必要である。	GPIO PA0,INPUT_FL ?IN(PA0)
INKEY	関数	イン・キー	INKEY()	入力文字コード取得 キーボードの入力状態をチェックし、押したキーの文字コード1 ～ 255を返す。キーが押されていない場合は0を返す。	入力したキーの文字コードを表示する 10 I= INKEY() 20 IF I>0 LOCATE 0,0: ?#3,I 30 GOTO 30
INPUT	コマンド	インプット	INPUT 変数 INPUT 変数,オーバーフロー時の既定値 INPUT "プロンプト", 変数 INPUT "プロンプト", 変数, オーバーフロー時の既定値	数値の対話入力 数値(符号+-、0～9)の入力を行い、変数に格納する。入力は[ENTER]キーで確定する。プロンプトの指定が可能、省略時は"変数名="を表示する。オーバーフロー時の既定値を指定した場合、オーバーフローエラーを発生せず、変数に既定値を格納する。	10 INPUT "AGE(0-150)=",A,-1 20 IF (A<0) OR (A>150) GOTO 10 30 IF A<15 ? "You are so young."
INPUT_FL	定数	インプット・フロート	INPUT_FL	GPIOモード設定定数 インプット・フロート(=2) GPIOコマンドのモード指定に利用、ArduinoのINPUTと同じ	GPIO PB0,INPUT_FL
INPUT_PD	定数	インプット・プルダウン、 インプット・ピー・ディー	INPUT_PD	GPIOモード設定定数 インプット・プルダウン(=5) GPIOコマンドのモード指定に利用	GPIO PB0,INPUT_PD
INPUT_PU	定数	インプット・プルアップ、 インプット・ピー・ユー	INPUT_PU	GPIOモード設定定数 インプット・プルアップ(=4) GPIOコマンドのモード指定に利用	GPIO PB0,INPUT_PU
KANJI	コマンド	カンジ	KANJI 横座標,縦座標,"文字列"	漢字文字列の描画 指定した座標にSDカード漢字フォントを使った文字列描画を行う。NTSC、OLED、TFTのグラフィック描画にて利用可能。フォントの前景色、背景色はGCOLORコマンドで指定する。 利用する漢字フォントサイズ等は事前にSETKANJIFONTコマンドで設定する。	SETKANJI 16,1,2,2,GW-1 KANJI 0,0,"さいたま"
KFONT	関数	ケー・フォント	KFONT(仮想アドレス,文字コード,フォントサイズ)	漢字フォントデータの取得 文字コード、フォントサイズ(8、10、12、14、16、20、24)に対応するフォントデータを指定した仮想アドレスに格納しする。 戻り値: 0:該当フォントなし、1:全角文字、 2:半角文字 半角カタカナコードを指定した場合、全角カタカナのデータ(全角)を格納する。	10 R=KFONT(MEM,ASC("あ"),8) 20 FOR I=0 TO 7 30 FOR J=0 TO 7 40 IF PEEK(MEM+I)&(\$80>>J) ?"#"; ELSE ?" "; 50 NEXT J 60 ? 70 NEXT I
LDBMP	コマンド	ロード・ビットマップ エル・ディー・ピー・エム・ピー	LDBMP "ファイル名",仮想アドレス,横座標,縦座標,幅,高さ[,色指定]	ビットマップ画像ファイルの読み込み SDカード内のビットマップ画像ファイル(モノクロ2値)の指定領域を指定した仮想メモリ上に格納する。引数の横座標、縦座標、幅、高さは画像の読み込み領域を指定する。色指定に1を指定した場合は反転して読み込む。省略または0を指定した場合はそのままの形式で読み込む。	10 CLS 20 LDBMP "TT.BMP",MEM,0,0,96,64 30 BITMAP 10,30,MEM,0,96,64,2 40 GOTO 40

キーワード	種別	読み	書式	概要	利用例
LEFT	定数	レフト	LEFT	スクロール方向指定 LEFT(=3) CSCROLL、GSCROLLコマンドのスクロール方向を指定する引数に利用する	CSCROLL 0,0,CW-1,CH-1,LEFT
LEN	関数	レン	LEN(変数) LEN(文字列)	文字列の文字数の取得 文字列、変数で参照してる文字列の文字数をカウントし、その値を返す。全角文字、半角文字とも1文字としてカウントする。	10 ?LEN("1あ2い34う5678") 20 A="ABさいたまCDEF" 30 ?LEN(A)
LET	コマンド	レット	LET 変数=数値 LET 配列変数=数値 [,数値, . . .] ※LETは省略可能	変数への数値代入 変数に数値を代入する。LETは省略可能。数値には式、数値、定数(文字列定数を含む)、変数、配列変数等 を評価した整数値である。配列変数への代入は、複数の値を指定した連続代入が可能。指定した添え字を起 点に順番に代入する。	LET X=5 LET @(3)=1,2,3,4,5 X=5:@(3)=1,2,3,4,5
LINE	コマンド	ライン	LINE 横座標1,縦座標1,横座標2,縦座標2,色	直線の描画 指定したグラフィク座標 (横座標1,縦座標1)と(横座標2,縦座標2)間を結ぶ直線を指定した色で描画す	LINE 0,0,GW-1,GH-1,1
LIST	コマンド	リスト	LIST LIST 表示行番号 LIST 表示開始行番号,表示終了行番号	プログラムリストの表示 プログラムリストの表示する。引数を指定しない場合は、全てのプログラムを表示する。 表示番号を指定した場合は、その行のプログラムリストを表示する。 表示開始番号、表示終了番号を指定した場合は、そこ範囲のプログラムリストを表示する。	LIST
LOAD	コマンド	ロード	LOAD LOAD プログラム番号 LOAD "ファイル名" LOAD "ファイル名", モード	プログラム読込 内部フラッシュメモリ、SDカードからプログラムを読み込む。 引数なしでは、内部フラッシュメモリのプログラム番号0を読み込む。 プログラム番号指定では、そのプログラム番号を読み込む。 ファイル名を指定した場合は、SDカードから指定ファイル読み込む。 モードに1を指定した場合は、既存のプログラムを消さずに追記して読み込む(ただし、同じ行番号は上書 き)。モード省略または0では、既存プログラムを消去後、プログラムをロードする。	LOAD LOAD 1 LOAD "SAMPLE.BAS"
LOCATE	コマンド	ロケート	LOCATE 横座標,縦座標	文字の表示位置の指定 コンソール画面のテキスト表示位置を指定する。	LOCATE 20,20:? "Hello"
LOW	定数	ロウ	LOW	デジタル信号LOW(=0) デジタル入出力のロジックレベルを表す定数	OUT PC13,LOW
LRUN	コマンド	エル・ラン	LRUN プログラム番号 LRUN プログラム番号,行番号 LRUN プログラム番号,"ラベル" LRUN "ファイル名" LRUN "ファイル名",行番号 LRUN "ファイル名","ラベル"	指定プログラムの実行 プログラム番号または、ファイル名で指定したプログラムをロードし、実行する。 行番号、ラベルの指定がある場合は、その行からプログラムを実行する。 LRUN にてプログラムを実行した場合、変数領域の初期化しない。直前のプログラムが設定した変数の値を引 き継ぐ。明示的に初期化したい場合は、CLVコマンドを実行する。	10 ?"***** メニュー *****" 20 ?"1:GEME1.BAS" 30 ?"2:GEME2.BAS" 40 ?"3:GEME3.BAS" 50 INPUT "No.=";n:IF (N>3) OR (N<1) GOTO 50 60 LRUN "GEME";n;".BAS"
LSB	定数	エル・エス・ビー	LSB	ビット入出力方向 下位 (=0) 下位ビットから送受信、SHIFTOUT、SHIFTIN コマンドで利用する。	SHIFTOUT PB12,PB14,LSB,D
MAP	関数	マップ	MAP(数値,範囲下限,範囲上限,新範囲下限,新範囲上限)	数値のスケール変換 指定した値のスケール変換を行い、その値を返す。	アナログ入力0~4095を電圧0~3.3Vに変換して表示 V=MAP(ANA(PA7),0,4095,0,3300) ?DMP\$(V,3);"V"
MEM	定数	メモ、 エム・イー・エム	MEM	ユーザーワーク領域先頭アドレス(=\$2BA0) ユーザーが自由に利用出来るワーク領域(1024バイト)の先頭アドレスを返す。	POKE MEM,1,2,3,4,5,6,7,8,9,10
MKDIR	コマンド	メイク・ディア、 エム・ケー・ディー・アイ・アール	MKDIR "ディレクトリ名"	ディレクトリの作成 SDカードにディレクトリを作成する。指定するディレクトリ名は、上位ディレクトリ名を含めて最大63文字まで指 定できる。作成するディレクトリの名称は8文字可能、大文字小文字の区別なし。	MKDIR "SRC"
MSB	定数	エム・エス・ビー	MSB	ビット入出力方向 上位 (=1) 上位ビットから送受信、SHIFTOUT、SHIFTIN コマンドで利用する。	SHIFTOUT PB12,PB14,MSB,D
NEW	コマンド	ニュー	NEW	プログラムの消去 プログラム領域のプログラム、変数、配列変数を消去・初期化する。	NEW
NEXT	制御コマンド	ネクスト	FOR 変数=初期値 TO 最終値 [STEP 変化量] 処理 NEXT 変数 ※[]は省略可能を示す。	FOR文の繰り返し範囲を指定 FOR ~NEXT間で、変数を初期値から最終値まで変化させ、処理を繰り返す。STEP文により、変数の変化量 の指定が可能(省略時は1)。	10 V=0 20 FOR I=1 TO 10 STEP 1 30 V=V+I 40 NEXT I 50 PRINT "V=";V
NOTONE	コマンド	ノー・トーン	NOTONE	単音出力停止 PB9ピンからのパルス出力を停止する。TONE 0と同じ。	TONE 800:WAIT 300 NOTONE
OFF	定数	オフ	OFF	デジタル信号OFF(=0) デジタル2値の状態を表す定数	IF IN(PA0)=OFF ?"SW is OFF"
OK	補助コマンド	オー・ケイ	OK	入力待機 スクリーンエディタでの入力待機状態を示す。コマンドラインで直接実行した場合は何も処理を行わない。行 番号付きのプログラム内で記述した場合は、何も処理をしない。	OK
ON	定数	オン	ON	デジタル信号ON(=1) デジタル2値の状態を表す定数	IF IN(PA0)=ON ?"SW is ON"
OR	二項演算子	オア	数値1 OR 数値2	論理OR演算子 数値1または数値2が真(0以外)であれば、真(1)を返す。そうでなければ、偽(0)を返す。	10 INPUT "AGE(0-150)=",A,-1 20 IF (A<0) OR (A>150) GOTO 10 30 IF A<15 ? "You are so young."
OUT	コマンド	アウト	OUT ピン番号,出力値	デジタル出力 指定ピンから、指定した出力を行う。 出力を行う場合は事前にGPIOコマンドによる機能設定(出力設定)が必要である。 ピン番号の指定には、0~34の数値または、定数PA0~PC15が可能である。 出力値の指定は、LOWまたは0で0Vを出力、HIGHまたは0以外の値で3.3Vを出力する。	10 GPIO PC13,OUTPUT 20 S=HIGH 30 OUT PC13,S 40 WAIT 500 50 S=!S 60 GOTO 30
OUTPUT	定数	アウトプット	OUTPUT	GPIOモード設定定数 アウトプット(=0) GPIOコマンドのモード指定に利用	GPIO PB0,OUTPUT
OUTPUT_OD	定数	アウトプット・オープン・ドレイ ン、	OUTPUT_OD	GPIOモード設定定数 アウトプット・オープン・ドレイン(=1) GPIOコマンドのモード指定に利用	GPIO PB0,OUTPUT_OD
PA0	定数	ピー・エー・ゼロ	PA0	PA0ピンのピン番号定数(=0) アナログ入力、デジタル入出力で利用可能	GPIO PA0,OUTPUT OUT PA0,HIGH
PA1	定数	ピー・エー・1	PA1	PA1ピンのピン番号定数(=1) アナログ入力、デジタル入出力で利用可能、NTSC版では利用禁止(同期信号で利用)	GPIO PA1,OUTPUT OUT PA1,HIGH

キーワード	種別	読み	書式	概要	利用例
PA2	定数	ピー・エー・2	PA2	PA2ピンのピン番号定数(=2) アナログ入力、デジタル入出力で利用可能	GPIO PA2,OUTPUT OUT PA2,HIGH
PA3	定数	ピー・エー・3	PA3	PA3ピンのピン番号定数(=3) アナログ入力、デジタル入出力で利用可能	GPIO PA3,OUTPUT OUT PA3,HIGH
PA4	定数	ピー・エー・4	PA4	PA4ピンのピン番号定数(=4) アナログ入力、デジタル入出力で利用可能、SDカードとの併用(SPI:NSSで利用)	GPIO PA4,OUTPUT OUT PA4,HIGH
PA5	定数	ピー・エー・5	PA5	PA5ピンのピン番号定数(=5) アナログ入力、デジタル入出力で利用可能、SDカードとの併用(SPI:SCKで利用)	GPIO PA5,OUTPUT OUT PA5,HIGH
PA6	定数	ピー・エー・6	PA6	PA6ピンのピン番号定数(=6) アナログ入力、デジタル入出力で利用可能、SDカードとの併用(SPI:MISOで利用)	GPIO PA6,OUTPUT OUT PA6,HIGH
PA7	定数	ピー・エー・7	PA7	PA7ピンのピン番号定数(=7) アナログ入力、デジタル入出力で利用可能、SDカードとの併用(SPI:MOSIで利用)	GPIO PA7,OUTPUT OUT PA7,HIGH
PA8	定数	ピー・エー・8	PA8	PA8ピンのピン番号定数(=8) デジタル入出力、PWMで利用可能	GPIO PA8,OUTPUT OUT PA8,HIGH
PA9	定数	ピー・エー・9	PA9	PA9ピンのピン番号定数(=9) デジタル入出力、PWMで利用可能、シリアル通信(TxD)との併用不可	GPIO PA9,OUTPUT OUT PA9,HIGH
PA10	定数	ピー・エー・10	PA10	PA10ピンのピン番号定数(=10) デジタル入出力、PWMで利用可能、シリアル通信(RxD)との併用不可	GPIO PA10,OUTPUT OUT PA10,HIGH
PA11	定数	ピー・エー・11	PA11	PA11ピンのピン番号定数(=11) USBで占有・利用禁止(D-で利用)	IF P=PA11 ? "Cannot use"
PA12	定数	ピー・エー・12	PA12	PA12ピンのピン番号定数(=12) USBで占有・利用禁止(D+で利用)	IF P=PA12 ? "Cannot use"
PA13	定数	ピー・エー・13	PA13	PA13ピンのピン番号定数(=13) デジタル入出力で利用可能、ST-LINKとの併用に注意	GPIO PA13,OUTPUT OUT PA13,HIGH
PA14	定数	ピー・エー・14	PA14	PA14ピンのピン番号定数(=14) デジタル入出力で利用可能、ST-LINKとの併用に注意	GPIO PA14,OUTPUT OUT PA14,HIGH
PA15	定数	ピー・エー・15	PA15	PA15ピンのピン番号定数(=15) デジタル入出力で利用可能	GPIO PA15,OUTPUT OUT PA15,HIGH
PB0	定数	ピー・ビー・0	PB0	PB0ピンのピン番号定数(=16) アナログ入力、デジタル入出力、PWMで利用可能	GPIO PB0,OUTPUT OUT PB0,HIGH
PB1	定数	ピー・ビー・1	PB1	PB1ピンのピン番号定数(=17) アナログ入力、デジタル入出力、PWMで利用可能	GPIO PB1,OUTPUT OUT PB1,HIGH
PB2	定数	ピー・ビー・2	PB2	PB2ピンのピン番号定数(=18) デジタル入出力(プルアップ抵抗あり)、BOOT1、起動時ターミナルモード選択	GPIO PB2,OUTPUT OUT PB2,HIGH
PB3	定数	ピー・ビー・3	PB3	PB3ピンのピン番号定数(=19) デジタル入出力で利用可能	GPIO PB3,OUTPUT OUT PB3,HIGH
PB4	定数	ピー・ビー・4	PB4	PB4ピンのピン番号定数(=20) デジタル入出力で利用可能、PS/2キーボード利用時占有(CLK)	GPIO PB4,OUTPUT OUT PB4,HIGH
PB5	定数	ピー・ビー・5	PB5	PB5ピンのピン番号定数(=21) デジタル入出力で利用可能、PS/2キーボード利用時占有(DATA)	GPIO PB5,OUTPUT OUT PB5,HIGH
PB6	定数	ピー・ビー・6	PB6	PB6ピンのピン番号定数(=22) I2C専用 SCL	IF P=PB6 ? "Cannot use"
PB7	定数	ピー・ビー・7	PB7	PB7ピンのピン番号定数(=23) I2C専用 SDA	IF P=PB7 ? "Cannot use"
PB8	定数	ピー・ビー・8	PB8	PB8ピンのピン番号定数(=24) デジタル入出力で利用可能	GPIO PB8,OUTPUT OUT PB8,HIGH
PB9	定数	ピー・ビー・9	PB9	PB9ピンのピン番号定数(=25) SOUND(PWM)専用	IF P=PB9 ? "Cannot use"
PB10	定数	ピー・ビー・10	PB10	PB10ピンのピン番号定数(=26) デジタル入出力で利用可能	GPIO PB10,OUTPUT OUT PB10,HIGH
PB11	定数	ピー・ビー・11	PB11	PB11ピンのピン番号定数(=27) デジタル入出力で利用可能、OLED、TFT利用時は使用禁止(CSで利用)	GPIO PB11,OUTPUT OUT PB11,HIGH
PB12	定数	ピー・ビー・12	PB12	PB12ピンのピン番号定数(=28) デジタル入出力で利用可能、OLED、TFT利用時は使用禁止(SC・DCで利用)	GPIO PB12,OUTPUT OUT PB12,HIGH
PB13	定数	ピー・ビー・13	PB13	PB13ピンのピン番号定数(=29) デジタル入出力で利用可能、OLED、TFT利用時は使用禁止(SPI SFAで利用)	GPIO PB13,OUTPUT OUT PB13,HIGH
PB14	定数	ピー・ビー・14	PB14	PB14ピンのピン番号定数(=30) デジタル入出力で利用可能、TFT利用時は使用禁止(TFT_SDOで利用)	GPIO PB14,OUTPUT OUT PB14,HIGH
PB15	定数	ピー・ビー・15	PB15	PB15ピンのピン番号定数(=31) デジタル入出力で利用可能、NTSC、TFT利用時は使用禁止(SPI SDAで利用)	GPIO PB15,OUTPUT OUT PB15,HIGH
PC13	定数	ピー・シー・13	PC13	PC13ピンのピン番号定数(=32) LED、デジタル入出力(プルアップ抵抗あり)	GPIO PC13,OUTPUT OUT PC13,LOW
PC14	定数	ピー・シー・14	PC14	PC14ピンのピン番号定数(=33) RTCで利用のため使用禁止	IF P=PC14 ? "Cannot use"
PC15	定数	ピー・シー・15	PC15	PC15ピンのピン番号定数(=34) RTCで利用のため使用禁止	IF P=PC15 ? "Cannot use"
PEEK	関数	ピーク	PEEK(仮想アドレス)	指定アドレスの値参照 指定した仮想アドレスに格納されている値(1バイト)を返す。	?PEEK(MEM+10)
PLAY	コマンド	プレイ	PLAY "MML文"	音楽演奏(MML文演奏) MML(Music Macro Language)にて定義した文法に従って、接続している圧電スピーカーを使って単音による、音楽演奏を行う。 詳細については、リファレンスマニュアルを参照。 "MML文"は、PRINT文と同様の記述が可能(文字列関数、連結';'等の利用が可能) RUNコマンドでプログラム実行の際、テンポや音の高さ等はデフォルト値に初期化される。	「ドラミファソラシド」と演奏する PLAY "CDEFGABxC"
POKE	コマンド	ポーク	POKE 仮想アドレス,数値 [,数値,...] ※..は以下同様に繰り返し可を示す。	メモリへの1バイトデータの書き込み 指定した仮想アドレスに数値を書き込む。数値は複数指定可能。	POKE MEM+1,123 (MEMは、ユーザー開放領域の先頭アドレス)

キーワード	種別	読み	書式	概要	利用例
POUT	コマンド	ピー・アウト	POUT ピン番号,デューティー値[,周波数]	PWM出力 指定ピンから、PWM出力を行う。 デューティー値： 0～4095 :0がデューティー比 0%～100%に対応 0で出力停止 周波数： 0 ～ 32767 (単位Hz)、省略時490Hz 事前に GPIOコマンドによるPWM設定(出力設定)が必要。 ピン番号は下記のみ有効、同一グループでは周波数が共通 PA6,PA7,PB0,PB1 (グループ1) PA8,PA9,PA10 (グループ2)	PB0にLEDを接続して1秒周期(1Hz)で5秒間点滅させる 10 GPIO PB0,PWM 20 POUT PB0,2047,1 30 WAIT 5000 40 POUT PB0,0,1
PRG	定数	プログ、ピー・アール・ジー	PRG	プログラム領域先頭アドレス(=\$1BA0) BASICプログラム(中間コード形式)が格納されている領域(4096バイト)の先頭アドレス	中間コード(バイナリ)形式で、プログラムをファイルに保存する BSAVE "PRG.BIN",PRG,4096
PRG2	定数	プログ2、ピー・アール・ジー2	PRG2	フラッシュメモリ内プログラム保存領域バンク先頭アドレス(=4F40) プログラム保存領域0～5のうちのBANK指定した領域(4096バイト)の先頭アドレス	プログラム番号5に保存されているプログラムをファイルに保存 BANK 5:BSAVE "PRG5.BIN",PRG2,4096 ファイルから直接プログラム番号5にプログラムを書き込む BANK 5:ERASE 5:BLOAD "PRG5.BIN",PRG2,4096
PRINT	コマンド	プリント	PRINT [#桁,]文字列(または数値、文字列関数)[;文字列;...][;] ※[]は省略可能を示す。 ※..は以下同様に繰り返し可を示す。	文字列・数値の表示 文字列または数値(または式の評価結果)、文字列関数を表示する。 ';'(セミコロン)により、文字列を連結して表示可能。文末に';'をつけた場合、改行しない。';'の代わりに','(カンマ)も利用可能。 PRINT文のみの場合改行のみを行う。 PRINT文の省略形、?も利用可能。	10 H=20:M=30:S=0 20 PRINT "TIME="; 30 PRINT #-2,H;" ":"M ":"S RUN TIME=20:30:00 OK
PSET	コマンド	ピー・セット、プセット	PSET 横座標,縦座標,色	点の描画 指定したグラフィク座標に指定した色の点を描画する。	PSET 50,20,1
PULSEIN	関数	パルス・イン	PULSEIN(パルス入力ピン番号,検出信号,タイムアウト[,スケール値])	入力パルス幅の計測 パルス入力ピン番号で指定したピンから入力パルス幅を計測し、その値を返す。 測定完了は、入力ピンの状態が検出信号の状態になった時点で計測を開始し、検出信号の状態でなくなった時点で計測を終了する。 検出信号には、LOW、HIGH または 0、1の測定対象のパルスを指定する。 タイムアウトにはパルス検出待ちタイムアウト時間 0 ～ 32767 (ミリ秒)を指定する。 スケール値には、計測時間のスケール変換 1 ～ 1327671 (デフォルト値 1)を指定する。 戻り値は、 正常時: 測定した パルス幅 0 ～ 32767(単位はスケール値 × マイクロ秒)、 タイムアウト時:0、 オーバーフロー時:-1 である。	パルス幅を計測する 10 CLS 20 GPIO PA8,INPUT_PU 30 A=PULSEIN(PA8,LOW,200,1) 40 LOCATE 0,0:?"#8,A 50 GOTO 30
PWM	定数	ピー・ダブル・エム	PWM	GPIOモード設定定数 PWM出力(=7) GPIOコマンドのモード指定に利用	GPIO PB0,PWM
RECT	コマンド	レクト	RECT 横座標1,縦座標1,横座標2,縦座標2,色,モード	矩形の描画 指定した座標に指定した色で矩形を描画する。モードに0を指定した場合は、線描画、1を指定した場合は塗りつぶし描画を行う。	RECT 20,20,40,40,1,1
REDRAW	コマンド	リ・ドゥロウ	REDRAW	画面の再表示 VPOKEまたは、VPEEKコマンドでVRAMに直接書き込んだ内容を表示画面に反映する。または、何ならの理由により表示内容が壊れた場合に再表示を行う。	10 POKE VRAM,ASC("A"),ASC("B") 20 REDRAW
REM	コマンド	レム、リマークス	REM コメント	コメントの記述 プログラムに説明等を記載する。‘ (シングルクォート)は REM の省略形.REM 以降の文字以降はすべてコメントとみなし、プログラムとして実行しない。プログラムの先頭行にコメントを付けた場合は、FILES コマンドで保存プログラム一覧を表示時に、各プログラムの見出しとなる。	1 REM Print starts 10 I=0:'initialize value 20 PRINT "**"; 30 I=I+1:IF I<10 GOTO 20 40 PRINT
REMOVE	コマンド	リムーブ	REMOVE "ファイル名"	ファイルの削除 SDカード内の指定したファイルを削除する。ファイル名にはディレクトリ名+ファイル名の形式での記述も可能である。	REMOVE "/SRC/SAMPLE.BAS"
RENUM	コマンド	リ・ナム、リ・ナンパー	RENUM RENUM 開始行番号 RENUM 開始行番号,増分	行番号の振り直し プログラム全体の行番号を指定した条件にて振り直す。引数を省略した場合は、行番号を10行から10間隔で振り直す。開始番号のみ指定した場合、指定した開始番号から10間隔で振り直す。開始番号と増分を指定した場合、指定した開始番号から指定した増分で振り直す。	RENUM 100,10
RETURN	制御コマンド	リターン	RETURN	サブルーチン呼び出しからの復帰 直前のGORUB文の呼び出しから復帰し、GOTUB文の次の位置に戻る。	10 GOSUB "SUB" 20 END 100 "SUB" 110 ?"Hello" 120 RETURN
RGB	関数	アール・ジー・ビー	RGB(赤,緑,青)	3原色からRGBコード変換 赤(0～31)、緑(0～63)、青(0～31)を合成した16ビットのRGBコードを返す。	COLOR RGB(15,15,15)
RIGHT	定数	ライト	RIGHT	スクロール方向指定 RIGHT(=2) CSCROLL、GSCROLLコマンドのスクロール方向を指定する引数に利用する	CSCROLL 0,0,CW-1,CH-1,RIGHT
RMDIR	コマンド	リムーブ・ディア、アール・エム・ディー・アイ・アール	RMDIR "ディレクトリ名"	ディレクトリの削除 SDカード内の指定したディレクトリを削除する。指定したディレクトリにサブディレクトリやファイルが存在する場合、そのファイルも削除する。	RMDIR "SRC"
RND	関数	ランド	RND(数値)	乱数の発生 0から指定した値－1の範囲の乱数を発生させ、その値を返す。	10 @(0)="グー","チョキ","パー" 20 R=RND(3) 30 ?STR\$(@((R)) 40 END
RUN	コマンド	ラン	RUN	プログラムの実行 プログラムの実行を行う。実行中のプログラムは ESCキーまたは、CTRL+Cキーにて強制終了することが出来る。シリアルコンソール上で ESCキーにて中断する場合は、ESCキーを2回押す必要がある。	RUN
SAVE	コマンド	セーブ	SAVE SAVE プログラム番号 SAVE "ファイル名" SAVE "ファイル名", 形式	プログラムの保存 プログラムをマイコン内のフラッシュメモリ、またはSDカードに保存する。 引数省略時は、内部フラッシュメモリのプログラム番号0に保存する。プログラム番号指定時は、そのプログラム番号領域に保存する。。引数にファイル名を指定した場合、SDカードに保存する。保存形式に0を指定した場合、中間コード形式(バイナリ)で保存する。1を指定または、形式を指定しない場合は、テキスト形式で保	SAVE SAVE 3 SAVE "SAMPLE.BAS" SAVE "SAMPLE.BAS",0
SAVECONF	コマンド	セーブ・コンフィーク	SAVECONFIG	システム環境設定の保存 CONFIG コマンドの設定変更をフラッシュメモリに保存する。設定内容は再起動後も有効である。	SAVECONFIG

キーワード	種別	読み	書式	概要	利用例
SCLOSE	コマンド	エス・クローズ	SCLOSE	シリアル通信クローズ データ通信設定したシリアルポートのシリアル通信を終了する。	10 SMODE 0 20 SOPEN "9600" 30 SPRINT "Hello,TinyBASIC" 40 SCLOSE
SCREEN	コマンド	スクリーン	SCREEN モード [,画面向き]	スクリーンモードの設定 デバイスコンソールNTSC、OLED、TFT画面のスクリーンモードの設定を行う。モード1でテキスト文字サイズ(NTSC版、OLED版では1〜3、TFT版では1〜6)、画面の向きで画面のテキストの回転(0〜3で90度ずつ回転、NTSC版では指定できない)を行う。	SCREEN 1
SETDATE	コマンド	セット・デイト	SETDATE 年,月,日,時,分,秒	時刻の設定 指定した時刻を内蔵RTCに設定する。	SETDATE 2018,11,24,12,0,3
SETKANJI	コマンド	セット・カンジ	SETKANJI サイズ,[倍角 [,横間隔, 行間[, 折り返し]]]	漢字設定 KANJIコマンドで利用する漢字フォントに対しての設定を行う。 サイズにはフォントサイズ(8、10、12、14、16、20、24)、倍角には1〜16、横間隔、行間は文字と文字の間に空けるドット数、折り返しには画面右側の折り返しする横座標を指定する。 デフォルトの設定は、SETKANJI 8,1,0,0,GW-1 である。	SETKANJI 16,1,2,2,GW-1 KANJI 0,0,"さいたま"
SHIFTIN	関数	シフト・イン	SHIFTIN(データピン番号,クロックピン番号,入力形式[,条件]) ※[]は省略可能を示す。	GPIO シリアルシフト入力 クロックピンにて同期を行い、データピンから1バイト分のデータを1ビットずつ入力する。入力形式には、LSBまたはMSBを指定する。条件はクロックの読み取りタイミングHIGHまたはLOWを指定する(省略時はHIGH)。	20 GPIO PB12,INPUT_FL 30 GPIO PB13,OUTPUT 40 GPIO PB14,OUTPUT 50 OUT PB13,HIGH 60 OUT PB13,LOW 70 ?SHIFTIN(PB12,PB14,LSB,LOW)
SHIFTOUT	コマンド	シフト・アウト	SHIFTOUT データピン番号,クロックピン番号,出力形式, 出力データ	GPIO シリアルシフト出力 クロックにて同期を行い、データピンから1バイト分のデータを1ビットずつ出力する。 データピン、クロックピンは事前に GPIO コマンドによる機能設定(デジタル出力)が必要である。出力形式には、LSBまたはMSBを指定する。	10 GPIO PB12,OUTPUT 20 GPIO PB14,OUTPUT 30 D=\$A0 40 OUT PB13,LOW 50 SHIFTOUT PB12,PB14,MSB,D 60 OUT PB13,HIGH
SMODE	コマンド	エス・モード	1)SMODE 0 2)SMODE 1,"通信速度" 3)SMODE 2 4)SMODE 3,制御コード処理指定	シリアルポート機能設定 2つのシリアルポート(USBポート、GPIOシリアルポート)に関する機能設定を行う。 1)USBポートをコンソール、GPIOシリアルポートをデータ通信で利用(デフォルト) 2)USBポートをデータ通信(SREAD、SWRITE等)、GPIOシリアルポートをコンソールとして利用する、利用の際、引数にて通信速度の指定を行う。 3)シリアル通信禁止、USBポート、GPIOシリアルポートとも入出力を行わない。 4)ターミナル利用の制御コードの処理の有無を設定する。0で無効、1(デフォルト)で有効。	GPIOシリアルポート(PA9、PA10)を115,200bpsでオープンし、コンソール画面として利用する。 SMODE 1,"115200" OK
SOPEN	コマンド	エス・オープン	SOPEN "通信速度"	シリアル通信オープン データ通信設定したGPIOシリアルポート(PA9、PA10)のシリアル通信の利用を開始する。通信速度は文字列形式の数値で指定する。 ※USBポートでのデータ通信時は、SOPENは不要である(常にオープンしているため)。	10 SMODE 0 20 SOPEN "9600" 30 SPRINT "Hello,TinyBASIC" 40 SCLOSE
SPRINT	コマンド	エス・プリント	SPRINT SPRINT [#桁,]文字列(または数値、文字列関数)[;文字列;...][;] ※[]は省略可能を示す。 ※.は以下同様に繰り返し可を示す。	シリアル通信文字列出力 指定した文字列、数値をデータ通信設定したシリアルポートに出力する。 書式及び使い方は PRINT文と同じである。文字列関数の利用も可能。	10 SMODE 0 20 SOPEN "9600" 30 SPRINT "Hello,TinyBASIC" 40 SCLOSE
SREAD	関数	エス・リード	SREAD()	シリアル通信1バイト受信 データ通信設定したシリアルポートから1バイトデータを受信する。 データが無い場合は-1を返す。データが無い場合の受信待ちは行わない。	10 SOPEN "115200" 20 D=SREAD() 30 IF D>=32 ?CHR\$(D); 40 GOTO 20
SREADY	関数	エス・レディ	SREADY()	シリアル通信データ受信確認 シリアルポートに受信可能なデータが着信しているかを確認する。データがある場合は1、ない場合は0を返す。	10 SOPEN "115200" 20 IF SREADY()==0 GOTO 20 30 D=SREAD() 40 ?CHR\$(D); 50 GOTO 20
STEP	補助コマンド	ステップ	FOR 変数=初期値 TO 最終値 [STEP 変化量] 処理 NEXT 変数 ※[]は省略可能を示す。	FOR文の変数の変化量の指定 FOR 〜NEXT間で、変数を初期値から最終値まで変化させ、処理を繰り返す。STEP文により、変数の変化量の指定が可能(省略時は1)。	10 V=0 20 FOR I=1 TO 10 STEP 1 30 V=V+I 40 NEXT I 50 PRINT "V=";V
STR\$	文字列関数	エス・ティー・アール・ダラー	STR\$(変数[,先頭位置,長さ]) STR\$("文字列"[,先頭位置,長さ]) ※[]は省略可能を示す。	文字列参照・文字列の切り出し PRINT文中で利用し、指定した変数が参照している文字列を出力する。 先頭位置、長さの指定がある場合は、文字を切り出して出力する。 変数の代わりに直接文字列の指定も可。	10 S="Hello,World." 20 ?STR\$(S) 30 ?STR\$(S,7,5) RUN Hello,World World OK
STRCMP	関数	エス・ティー・アール・コンブ	STRCMP(文字列1,文字列2[,長さ]) ※文字列1、文字列2には、文字列定数、文字列を参照している変数の指定が可能	文字列の比較 文字列1と文字列2を比較し、等しい場合は0、等しくない場合は1を返す。	10 S=GETS(MEM) 20 IF STRCMP(S,"end") END 30 GOTO 10
SWRITE	コマンド	エス・ライト	SWRITE 送信データ	シリアル通信1バイト出力 データ通信設定したシリアルポートに1バイト分のデータを出力する。	10 SOPEN "115200" 20 FOR C=ASC("A") TO ASC("Z") 30 SWRITE C 40 NEXT C 50 SCLOSE
SYSINFO	コマンド	シス・インフォ	SYSINFO	システム情報表示 スタックアドレス、ヒープアドレス、SRAM未使用サイズ、キーワード数を表示する。	SYSINFO Stack Top:20004F8F Heap Top :20004B50 SRAM Free:1087 symbol Num:217

キーワード	種別	読み	書式	概要	利用例
TEMPO	コマンド	テンポ	TEMPO テンポ	音楽演奏のテンポの設定 PLAY コマンドによる音楽演奏のテンポの設定を行う(デフォルト値は120)。 RUNコマンドでプログラム実行の際、テンポはデフォルト値に初期化される。	10 TEMPO 160 20 PLAY "CDEFGAB"
TICK	関数	ティック	TICK() TICK(モード)	システム起動経過時間の取得 起動からの経過時間を返す。経過時間はCLTコマンドにて初期化が可能。 モードの指定無し、または0を指定の場合、ミリ秒単位の経過時間を返す。 1を指定した場合は、秒単位の経過時間を返す。	1万回ループの実行時間を測定する 20 CLT:A=0 30 FOR I=1 TO 10000 40 A=A+1 50 NEXT I 60 ?A 70 ?TICK();"msec"
TO	補助コマンド	ツー	FOR 変数=初期値 TO 最終値 [STEP 変化量] 処理 NEXT 変数 ※[]は省略可能を示す。	FOR文の変数最終値指定 FOR ~NEXT間で、変数を初期値から最終値まで変化させ、処理を繰り返す。STEP文により、変数の変化量の指定が可能(省略時は1)。	10 V=0 20 FOR I=1 TO 10 STEP 1 30 V=V+I 40 NEXT I 50 PRINT "V=";V
TONE	コマンド	トーン	TONE 周波数[,出力期間]	単音出力 PB9ピンより、指定した周波数のパルス出力(デューティ比50%)を行う。 PB9 ピンに圧電スピーカ(圧電サウンダ)を接続することで単音出力を行う。 引数の出力期間を指定しない場合または0を指定した場合、パルスを出し続ける。指定した場合、その期間だけパルスを出力する。周波数に0を指定した場合、パルス出力を停止する。	スペースキーを押したら音を鳴らす 10 IF INKEY() = 32 TONE 800,50 20 GOTO 10
UP	定数	アップ	UP	スクロール方向指定 UP(=0) CSCROLL、GSCROLLコマンドのスクロール方向を指定する引数に利用する	CSCROLL 0,0,CW-1,CH-1,UP
VAR	定数	バー ブイ・エー・アール	VAR	変数領域先頭アドレス(\$1900) 変数領域(A~Z, A0~Z6)の440バイトの先頭アドレスを返す。	10 POKE VAR,123,0 20 ?A
VPEEK	関数	ブイ・ピーク	VPEEK(横座標,縦座標)	指定座標のテキスト文字コード参照 画面上の指定座標の文字コードを取得する。指定座標が範囲外の場合は0を返す。	?VPEEK(10,10)
VRAM	定数	バイラム	VRAM	コンソールテキスト画面VRAM 先頭アドレス(=\$0000) テキストスクリーン 横CW(文字)×縦C(H文字)バイト分の領域の先頭アドレスを返す。	10 POKE VRAM,ASC("A"),ASC("B"),ASC("C") 20 REDRAW
WAIT	コマンド	ウェイト	WAIT 時間(ミリ秒)	時間待ち 指定した時間経過するまで、時間待ち(ウェイト)を行う。	10 FOR I=1 TO 5 20 WAIT 500:? "Hello" 30 NEXT I
WIDTH	コマンド	ワイス	WIDTH 横文字数,縦文字数	シリアルコンソール画面サイズ設定 スクリーンエディタの表示文字数(縦・横)、スクロール範囲の設定を行う。 最小サイズ16桁×10行 ~ 最大128桁×45行 まで指定可能(デフォルトは80x24)。	WIDTH 120,24
ZEN	関数	ゼン	ZEN(文字コード)	半角文字コードを全角に変換 指定した文字コードが半角文字の場合、その全角文字コードを返す。	?HEX\$(ZEN(ASC("A")))