

IchigoJam BASIC リファレンス ver 1.4

コマンド	解説	例
LED / エル・イー・ディー	数が1ならLEDが光り、0なら消える	LED 1
WAIT / ウェイト	数1の数だけ待つ（60で1秒）。マイナスの数を指定すると走査線分で待つ(-261でWAIT1と同じ時間)。省略できる数2に0を指定すると画面表示を止め低電力化して待つ。	WAIT 60
: / コロン	コマンドを後ろに続けて書くときの区切り記号	WAIT 60:LED 1
1 / ワン	1. 数は-32768から32767まで表記できる。 2. 行頭の1〜32767で指定された数は行番号として、コマンドと合わせて記録する。コマンドを省略するとその行を削除（16385以上は指定しないと表示しない）	10 LED1
RUN / ラン	プログラムを実行する [F5]	RUN
LIST / リスト	プログラムを表示する [F4]（行番号1で1行表示、行番号1がマイナスでその行まで表示、行番号2指定でその行まで表示、行番号2が0の時終わりまで表示、ESCで途中停止）	LIST 10,300
GOTO / ゴートゥー	指定した行番号へ飛ぶ（式も指定可能）	GOTO 10
END / エンド	プログラムを終了する	END
IF / イフ	数が0でなければコマンド1を実行し、0であればコマンド2を実行する（THEN,ELSE以降は省略可）	IF BTN() END
THEN / ゼン	数が0でなければコマンド1を実行し、0であればコマンド2を実行する（THEN,ELSE以降は省略可）	IF BTN() THEN END
ELSE / エルス	数が0でなければコマンド1を実行し、0であればコマンド2を実行する（THEN,ELSE以降は省略可）	IF BTN() END ELSE CONT
BTN / ボタン	ボタンが押されているば1、そうでないとき0を返す（数：0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE/X(88)、省略で0、-1でビットパターンで返す）	LED BTN()
NEW / ニュー	プログラムを全部消す	NEW
PRINT / プリント	数や文字列を表示する（文字列は"で囲む、";" で連結できる）	PRINT "HII";15
? / クエスション	数や文字列を表示する（文字列は"で囲む、";" で連結できる）	? "HII";15
LOCATE / ロケート	次に文字を書く位置を横、縦の順に指定する（左上が0,0、縦=-1で無表示）。数1のみでX+Y*幅。数3が0でなければ指定した場所にカーソルを表示する。	LOCATE 3,3
LC / ロケート	次に文字を書く位置を横、縦の順に指定する（左上が0,0、縦=-1で無表示）。数1のみでX+Y*幅。数3が0でなければ指定した場所にカーソルを表示する。	LC 3,3
CLS / クリア・スクリーン	画面を全部消す	CLS
RND / ランダム	0から数未満のでたらめな整数を返す	PRINT RND(6)
SAVE / セーブ	プログラムを保存する（0〜3の4つ、100-227 外付けEEPROM、省略で前回使用した数） ボタンを押した状態で起動すると0番を読み込み自動実行	SAVE 1
LOAD / ロード	プログラムを読み出す（0〜3の4つ、100-227 外付けEEPROM、省略で前回使用した数）	LOAD
FILES / ファイルズ	数1(省略可)〜数2のプログラム一覧を表示する（EEPROM内ファイル表示に対応、0指定ですべて表示、ESCで途中停止）	FILES
BEEP / ビープ	単音を鳴らす 周期(0-32767)と長さ(1/60秒単位)は省略可 ※SOUND(EX2)-GNDに圧電サウンダーなどの接続が必要	BEEP
PLAY / プレイ	MML、記述した音楽を鳴らす。省略で音停止 ※SOUND(EX2)-GNDに圧電サウンダーなどの接続が必要（MML、CDEFGAB：ドレミファソラシ、R：休符、.：音や休符を1.5倍伸ばす、T120：テンポ、O4：オクターブ(1-6)、N10：単音、<：オクターブ上げる、>：オクターブ下げる、\$：繰り返し、'：以後鳴らさない）	PLAY "\$CDE2CDE2"
TEMPO / テンポ	再生中の音楽のテンポを変更する	TEMPO 1200
+ / プラス	足し算する	PRINT 1+1
- / マイナス	1. 引き算する 2. 後続く数をマイナスにする	PRINT 2-1
* / アスタリスク	掛け算する	PRINT 7*8
// スラッシュ	割り算する（小数点以下は切り捨て）	PRINT 9/3
% / パーセント	割り算した余りを返す	PRINT 10%3
() / ブラケット	カッコ内の式を優先して計算する	PRINT 1+(1*2)
LET / レット	数をアルファベット 1 文字の変数や配列の値として記憶する（配列へ連続代入可能 LET[0],1,2)	LET A,1
= / イコール	1. 数をアルファベット 1 文字の変数や配列の値として記憶する。 2. 式として使うと==と同じ機能となる	A=1
INPUT / インプット	キーボードやUARTからの入力で数値を変数にいれる（文字列とコンマは省略可）	INPUT "ANS?",A
TICK / ティック	CLTからの時間を返す（1/60秒で1進む） *数に1指定で1/(60*261)秒で1進む時間	PRINT TICK()
CLT / クリア・ティック	TICK()のカウントをリセットする	CLT
INKEY / インキー	キーボードやUARTから 1 文字入力する（入力がない時は0、UARTから0が入力された時は#100）	PRINT INKEY()
LEFT / レフト	28を返す。INKEYのキーコードの判定や、SCROLLに使う	IF INKEY()=LEFT LED1
RIGHT / ライト	29を返す。INKEYのキーコードの判定や、SCROLLに使う	IF INKEY()=RIGHT LED1
UP / アップ	30を返す。INKEYのキーコードの判定や、SCROLLに使う	IF INKEY()=UP LED1
DOWN / ダウン	31を返す。INKEYのキーコードの判定や、SCROLLに使う	IF INKEY()=DOWN LED1
SPACE / スペース	32を返す。INKEYのキーコードの判定や、SCROLLに使う	IF INKEY()=SPACE LED1
CHR / キャラクター	PRINT内で、文字コードに対応する文字を返す（コンマ区切りで連続表記可）	PRINT CHR\$(65)
ASC / アスキー	文字に対する文字コードを返す	PRINT ASC("A")
SCROLL / スクロール	指定した方向に1キャラ分スクロールする（0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左）	SCROLL 2
SCR / スクリーン	画面上の指定した位置に書かれた文字コードを返す（指定なしで現在位置）	PRINT SCR(0,0)
VPEEK / ブイ・ピーク	画面上の指定した位置に書かれた文字コードを返す（指定なしで現在位置）	PRINT VPEEK(0,0)
== / イコール・イコール	比較して等しい時に1、それ以外で0を返す。	IF A==B LED 1
!= / ノット・イコール・トゥー	比較して等しくない時に1、それ以外で0を返す	IF A!=B LED 1
> / レス・アンド・グレーター・ザン	比較して等しくない時に1、それ以外で0を返す	IF A>B LED 1
<= / レス・ザン・オア・イコール・トゥー	比較して以下の時に1、それ以外で0を返す	IF A<=B LED 1
< / レス・ザン	比較して未満の時に1、それ以外で0を返す	IF A<B LED 1
>= / グレーター・ザン・オア・イコール・トゥー	比較して以上の時に1、それ以外で0を返す	IF A>=B LED 1
> / グレーター・ザン	比較してより大きい時に1、それ以外で0を返す	IF A>B LED 1
AND / アンド	どちらの式も1の時に1、それ以外で0を返す	IF A=1 AND B=1 LED 1
&& / アンド	どちらの式も1の時に1、それ以外で0を返す	IF A=1 && B=1 LED 1
OR / オア	どちらかの式が1の時に1、それ以外で0を返す	IF A=1 OR B=1 LED 1
/ オア	どちらかの式が1の時に1、それ以外で0を返す	IF A=1 B=1 LED 1
NOT / ノット	式が0の時に1、それ以外で0を返す	IF NOT A=1 LED 1
! / ノット	式が0の時に1、それ以外で0を返す	IF !(A=1) LED 1
REM / リマーク	これ以降のコマンドを実行しない（コメント機能）	REM START
' / シングル・クォート	これ以降のコマンドを実行しない（コメント機能）	START
FOR / フォー	変数に数1をいれ、数2になるまで数3ずつ増やしながらNEXTまでをくりかえす（STEPは省略可、6段まで）	FOR I=0 TO 10:?!:NEXT
TO / トゥー	変数に数1をいれ、数2になるまで数3ずつ増やしながらNEXTまでをくりかえす（STEP省略時は数3は1、6段まで）	FOR I=0 TO 10:?!:NEXT
STEP / ステップ	変数に数1をいれ、数2になるまで数3ずつ増やしながらNEXTまでをくりかえす（STEPは省略可、6段まで）	FOR I=0 TO 10:?!:NEXT
NEXT / ネクスト	変数に数1をいれ、数2になるまで数3ずつ増やしながらNEXTまでをくりかえす（STEPは省略可、6段まで）	FOR I=0 TO 10:?!:NEXT

POS / ポジション	カーソル位置を返す（数、省略時または0:X+Y*幅、1:X座標、2:Y座標） *ver1.4以上	?POS(0),POS(1)
DRAW / ドロー	数1,数2の座標から数3,数4の座標へ線を引く（座標は最大63x47）、数5に0指定で線を消し、2指定で反転する、省略時または1指定で線を引く *ver1.4以上	DRAW 1,5,10,15
POINT / ポイント	数1,数2の座標にDRAWで描かれた点または文字があるときに1、そうでないとき0を返す *ver1.4以上	?POINT(1,5)
OUT / アウト	外部出力OUT1-11に0または1を出力する 数2を省略でまとめて出力できる（数2に-1指定でINへ切り替え、-2指定でプルアップ付きINへ切り替え ※IN3は除く）	OUT 1,1
IN / イン	IN0-10から入力する（0または1） 数を省略してまとめて入力できる（IN0,1,4,9はプルアップ、IN5-8,10-11はOUTで切り替え時使用可能、IN0,9はボタン）	LET A,IN(1)
ANA / アナログ	外部入力 の電圧(0V-3.3V)を0-1023の数値で返す(2:IN2、5-8:IN5-8(OUT1-4)、0,9:BTN、省略で0)	?ANA()
PWM / ピー・ダブリュー・エム	外部出力OUT2-5に数2で0.01msec単位で指定するパルスを出力する（0-2000、周期20msec）、数3で周期を指定（省略時2000=20msec、マイナス値指定で周期1/480）	PWM 2,100
CLV / クリア・バリアブル	変数、配列を全部0にする	CLV
CLEAR / クリア	変数、配列を全部0にする	CLEAR
CLK / クリア・キー	キーバッファとキーの状態をクリアする	CLK
CLO / クリア・アウトプット	入出力ピンを初期状態に戻す	CLO
ABS / アブソリュート	絶対値を返す（マイナスはプラスになる）	?ABS(-2)
[] / アレイ	配列（[0]から[101]までの102コの連続した変数として使える） LET[0],1,2,3で連続代入可能	[3]=1
GOSUB / ゴーサブ	戻り先をスタックに積み、数または式で指定した行番号に飛ぶ（ネストは30段まで）	GOSUB 100
GSB / ゴーサブ	戻り先をスタックに積み、数または式で指定した行番号に飛ぶ（ネストは30段まで）	GSB 100
RETURN / リターン	GOSUB/GSBの呼び出し元へ戻る	RETURN
RTN / リターン	GOSUB/GSBの呼び出し元へ戻る	RTN
DEC / デシ	PRINT内で、数を文字列にする（2番目の数は桁数、省略可）	?DEC\$(99,3)
# / ハッシュ	16進数で数を表記する *この後THENを省略しA～Fで始まるコマンドを使う場合：で区切る	#FF
HEX / ヘックス	PRINT内で、数を16進数の文字列にする（2番目の数は桁数、省略可）	?HEX\$(255,2)
` / バック・クォート	2進数で数を表記する	`1010
BIN / バイナリー	PRINT内で、数を2進数の文字列にする（2番目の数は桁数、省略可）	?BIN\$(255,8)
& / アンパサンド	論理積（ビット演算）	?3&1
/ パイプ	論理和（ビット演算）	?3 1
^ / ハット	排他的論理和（ビット演算）	?A^1
>> / シフト・ライト	右シフトする（ビット演算）	?A>>1
<< / シフト・レフト	左シフトする（ビット演算）	?A<<1
~/チルダ	ビット反転（ビット演算）	?~A
COS / コサイン	指定された数を角度の度数としてコサインの値の256倍を返す *ver1.4以上	?COS(90)
SIN / サイン	指定された数を角度の度数としてサインの値の256倍を返す *ver1.4以上	?SIN(90)
STOP / ストップ	プログラムを中断する	STOP
CONT / コンティニュー	実行中の行や、中断した行を再度実行する	CONT
SOUND / サウンド	音が再生中なら1、そうで無いとき0をを返す	?SOUND()
FREE / フリー	プログラムの残りメモリ数を返す	?FREE()
VER / バージョン	IchigoJam BASICのバージョン番号を返す	?VER()
RENUM / リナンバー	プログラムの行番号を数1(省略時は10)から数2(省略時は10)刻みにする。GOTO/GOSUBの飛び先は手で変更必要な場合がある	RENUM
LRUN / ロードラン	プログラムを読み込み後、実行する	LRUN 1
FILE / ファイル	最後にプログラムを読み込み、書き込み行った数を返す	?FILE()
LINE / ライン	現在実行中の行番号を返す（非実行時は0）	?LINE()
SRND / エス・ランダム	種を指定して乱数を初期化する	SRND 0
HELP / ヘルプ	メモリマップを表示する	HELP
PEEK / ピーク	メモリ読み出し（キャラクターパターン0-#7FFなど）	?PEEK(#700)
POKE / ポーク	メモリへの書き込み（連続書き込み可能 POKE#700,1,2,3）	POKE #700,#FF
COPY / コピー	メモリコピー 数1のアドレスへ数2のアドレスから数3の長さ分コピー(数3マイナスでコピー方向が逆になる)	COPY #900,0,256
CLP / クリア・パターン	キャラクターパターン(#700-#7FF)を初期化する	CLP
" / ダブル・クォート	文字列の先頭アドレスを返す	A="ABC"
STR / スtring	PRINT内で、文字列を返す（数2(省略可)で長さ指定）	PRINT STR\$(A)
LEN / レングス	文字列の長さを返す	PRINT LEN("ABC")
@ / アット・マーク	行の先頭に書くとラベルとなり、行番号の代わりとして使える（GOTO @LOOPなど） *前方一致	@LOOP
VIDEO / ビデオ	画面表示非表示を切り替える 0で画面表示を停止し処理高速化（F8で表示）、省略可能な数2でVIDEO0時CPUクロックを1/数2に変更し省電力化、数1が2の倍数で白黒反転、数1が3以上で拡大モード	VIDEO 0
RESET / リセット	IchigoJamをリセットする	RESET
SLEEP / スリープ	プログラムを休止する (ボタンを押すと起動し、LRUN0を実行する)	SLEEP
UART / ユー・アート	数1：シリアル出力設定（0:オフ、1:PRINTのみ、2:PRINT/LC/CLS/SCROLL、3:PRINTのみ/改行コード\r\n、+4で入力エコーバック、+8で画面表示OFF、初期値:2）、数2：シリアル受信設定（0:オフ、1:オン、+2:ESC無効、+4:CR変換(13→10) 省略時1）	UART 0
BPS / ビー・ピー・エス	シリアル通信速度を変更する(0で初期値の115,200bps、-1:57600bps、-2:38400bps、-100以下指定で指定した数の-100倍bpsに指定、-2304:230400bps)。数2でI2Cの通信速度設定（単位kHz、0:デフォルト400kHz)	BPS 9600
OK / オーケー	OKやエラーメッセージの表示有無を切り替える（数、1:表示、2:非表示、省略で1）	OK 2
I2CR / アイ・ツー・シー・リード	I2Cで周辺機器から読み込む I2Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ（コマンド送信が1byteの時数3を省略可、コマンド送信が0byteの時数2/数3を省略可）	R=I2CR(#50,#114A,2,#114C,2)
I2CW / アイ・ツー・シー・ライト	I2Cで周辺機器に書き込む I2Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ（数4/数5は省略可、コマンド送信が1byteの時数3を省略可）	R=I2CW(#50,#114A,2,#114C,2)
IOT.IN / アイ・オー・ティー・イン	sakura.ioモジュールから受信した数を一つ読み込む	R=IoT.IN()
IOT.OUT / アイ・オー・ティー・アウト	sakura.ioモジュールへ数をチャンネル0で即時送信する	IoT.OUT 100
WS.LED / ダブリュー・エス・エル・イー・ディー	配列の先頭から緑赤青の順に設定された値でLEDに接続されたWS2812Bを数1の分光らせる。数2を指定するとその数だけ繰り返す。 *ver1.4以上	WS.LED 3
SWITCH / スイッチ	画面表示をテレビと液晶とを切り替える（数1 0:テレビ、1:液晶）、数2で液晶の濃さを指定	SWITCH
USR / ユーザー	数1で指定されたアドレスのマシン語(Arm Cortex-M0)を数2をパラメータとして呼び出す（数2省略で0）	A=USR(#700,0)