

IchigoJam BASIC 1.4 コマンド一覧

コマンド	書式	説明
エル・イー・ディー LED	LED 数	数が1ならLEDが光り、0なら消える
ウェイト WAIT	WAIT 数1{,数2}	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まで待つ(-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。
コロソ :	:	コマンドを後ろに続けて書くときの区切り記号
ファン 1	行番号 {コマンド}	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)
ラン RUN	RUN	プログラムを実行する [F5]
リスト LIST	LIST {行番号1{,行番号2}}	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)
ゴートゥー GOTO	GOTO 行番号	指定した行番号へ飛び (式も指定可能)
エンド END	END	プログラムを終了する
イフ IF	IF 数 {THEN} コマンド1 {ELSE コマンド2}	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)
ゼン THEN	IF 数 {THEN} コマンド1 {ELSE コマンド2}	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)
エルズ ELSE	IF 数 {THEN} コマンド1 {ELSE コマンド2}	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)
ボタン BTN	BTN({数})	ボタンが押されれば1、そうでないとき0を返す (数: 0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE、省略で0)
ニュー NEW	NEW	プログラムを全部消す
プリント PRINT	PRINT {数や文字列}	数や文字列を表示する (文字列は"で囲む、"で連結できる)
クエスション ?	? {数や文字列}	数や文字列を表示する (文字列は"で囲む、"で連結できる)
ロケート LOCATE	LOCATE 数1,数2{,数3}	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3が0でなければ指定した場所にカーソルを表示する。
ロケート LC	LC 数1,数2{,数3}	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3が0でなければ指定した場所にカーソルを表示する。
クリア・スクリーン CLS	CLS	画面を全部消す
ランダム RND	RND(数)	0から数未満のでたらめな整数を返す
セーブ SAVE	SAVE {数}	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
ロード LOAD	LOAD {数}	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)
ファイルズ FILES	FILES {数1{,数2}}	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)
ビーブ BEEP	BEEP {数1{,数2}}	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
プレイ PLAY	PLAY {"MML"}	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)
テンポ TEMPO	TEMPO 数	再生中の音楽のテンポを変更する
プラス +	数 + 数	足し算する
マイナス -	数 - 数	1. 引き算する 2. 後に続く数をマイナスにする
アスタリク *	数 * 数	掛け算する
スラッシュ /	数 / 数	割り算する (小数点以下は切り捨て)

パーセント %	数 % 数	割り算した余りを返す
ブラケット ()	(式)	カッコ内の式を優先して計算する
レット LET	LET 変数,数	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)
イコール =	変数=数	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
インプット INPUT	INPUT {文字列},変数	キーボードやUARTからの入力で数値を変数にいれる (文字列とコマンドは省略可)
ティック TICK	TICK()	CLTからの時間を返す (1/60秒で1進む)
クリア・ティック CLT	CLT	TICK()のカウンタをリセットする
インキー INKEY	INKEY()	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)
レフト LEFT	LEFT	28を返す。INKEYのキーコードの判定や、SCROLLに使う
ライト RIGHT	RIGHT	29を返す。INKEYのキーコードの判定や、SCROLLに使う
アップ UP	UP	30を返す。INKEYのキーコードの判定や、SCROLLに使う
ダウン DOWN	DOWN	31を返す。INKEYのキーコードの判定や、SCROLLに使う
スペース SPACE	SPACE	32を返す。INKEYのキーコードの判定や、SCROLLに使う
キャラクター CHR	CHRS(数{,数2{,数3...})	PRINT内で、文字コードに対応する文字を返す (コマンド区切りで連続表記可)
アスキー ASC	ASC("文字")	文字に対する文字コードを返す
スクロール SCROLL	SCROLL 数	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)
スクリーン SCR	SCR({数,数})	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)
ヴィ・ピーク VPEEK	VPEEK({数,数})	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)
イコール・イコール ==	数 == 数	比較して等しい時に1、それ以外で0を返す。
ノット・イコール・トラウ !=	数 != 数	比較して等しくない時に1、それ以外で0を返す
レス・アンド・グレートー・ザン <>	数 <> 数	比較して等しくない時に1、それ以外で0を返す
レス・ザン・オア・イコール・トラウ <=	数 <= 数	比較して以下の時に1、それ以外で0を返す
レス・ザン <	数 < 数	比較して未満の時に1、それ以外で0を返す
グレートー・ザン・オア・イコール・トラウ >=	数 >= 数	比較して以上の時に1、それ以外で0を返す
グレートー・ザン >	数 > 数	比較してより大きい時に1、それ以外で0を返す
アンド AND	式 AND 式	どちらの式も1の時に1、それ以外で0を返す
アンド &&	式 && 式	どちらの式も1の時に1、それ以外で0を返す
オア OR	式 OR 式	どちらかの式が1の時に1、それ以外で0を返す
オア 	式 式	どちらかの式が1の時に1、それ以外で0を返す
ノット NOT	NOT 式	式が0の時に1、それ以外で0を返す
ノット !	! 式	式が0の時に1、それ以外で0を返す
リマーク REM	REM	これ以降のコマンドを実行しない (コメント機能)
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)
フォー FOR	FOR 変数=数1 TO 数2 {STEP 数3}:コマンド:NEXT	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)
トゥー TO	FOR 変数=数1 TO 数2 {STEP 数3}:コマンド:NEXT	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)
ステップ STEP	FOR 変数=数1 TO 数2 STEP 数3:コマンド:NEXT	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)

ネクスト NEXT	FOR 変数=数1 TO 数2 {STEP 数3}:コマンド:NEXT	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)
ポジション POS	POS({数})	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
ドロー DRAW	DRAW 数1, 数2,{数3, 数4},{数5}	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、省略時と1指定で線を引く *ver1.4以上
ポイント POINT	POINT(数1, 数2)	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
アウト OUT	OUT 数1,{数2}	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)
イン IN	IN({数})	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)
アナログ ANA	ANA({数})	外部入力電圧(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)
ピー・ダブリュー・エム PWM	PWM 数1, 数2,{数3}	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)
クリア・バリャブル CLV	CLV	変数、配列を全部0にする
クリア CLEAR	CLEAR	変数、配列を全部0にする
クリア・キー CLK	CLK	キーバツファとキーの状態をクリアする
クリア・アウトプット CLO	CLO	入出力ピンを初期状態に戻す
アブソリュート ABS	ABS(数)	絶対値を返す (マイナスはプラスになる)
アレイ []	[数]	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
ゴースブ GOSUB	GOSUB 行番号	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)
ゴースブ GSB	GSB 行番号	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)
リターン RETURN	RETURN	GOSUB/GSBの呼び出し元へ戻る
リターン RTN	RTN	GOSUB/GSBの呼び出し元へ戻る
デシ DEC	DEC\$(数1,{数2})	PRINT内で、数を文字列にする (2番目の数は桁数、省略可)
ハッシュ #	#16進数	16進数で数を表記する
ヘックス HEX	HEX\$(数1,{数2})	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)
バック・クォート '	'2進数	2進数で数を表記する
バイナリー BIN	BIN\$(数1,{数2})	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)
アンバサンド &	数 & 数	論理積 (ビット演算)
パイプ 	数 数	論理和 (ビット演算)
ハット ^	数 ^ 数	排他的論理和 (ビット演算)
シフト・ライト >>	数 >> 数	右シフトする (ビット演算)
シフト・レフト <<	数 << 数	左シフトする (ビット演算)
チルド ~	~数	ビット反転 (ビット演算)
コサイン COS	COS(数)	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
サイン SIN	SIN(数)	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
ストップ STOP	STOP	プログラムを中断する
コンティニュー CONT	CONT	実行中の行や、中断した行を再度実行する
サウンド SOUND	SOUND()	音が再生中なら1、そうで無いたとき0を返す

フリー FREE	FREE()	プログラムの残りメモリ数を返す
バージョン VER	VER()	IchigoJam BASICのバージョン番号を返す
リナンバー RENUM	RENUM {数1,{数2}}	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
ロードラン LRUN	LRUN {数}	プログラムを読み込み後、実行する
ファイル FILE	FILE()	最後にプログラムを読み込み、書き込みを行った数を返す
ライン LINE	LINE()	現在実行中の行番号を返す (非実行時は0)
エス・ランダム SRND	SRND 数	種を指定して乱数を初期化する
ヘルプ HELP	HELP	メモリマップを表示する
ピーク PEEK	PEEK(数)	メモリ読み出し (キャラクターパターン0-#7FFなど)
ポーク POKE	POKE 数,数	メモリへの書き込み (連続書き込み可能 POKE#700,1,2,3)
コピー COPY	COPY 数1, 数2, 数3	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)
クリア・パターン CLP	CLP	キャラクターパターン(#700-#7FF)を初期化する
ダブル・クォート "	"文字列"	文字列の先頭アドレスを返す
ストリング STR	STR\$(数1,{数2})	PRINT内で、文字列を返す (数2(省略可)で長さ指定)
レングス LEN	LEN("文字列")	文字列の長さを返す
アット・マーク @	@ラベル	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
ビデオ VIDEO	VIDEO 数1,{数2}	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
リセット RESET	RESET	IchigoJamをリセットする
スリープ SLEEP	SLEEP	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)
ユー・アート UART	UART 数1,{数2}	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、左記に+4で入力エコーバック、初期値2)、数2:シリアル受信設定 (0:オフ、1:オン 省略時)
ビー・ピー・エス BPS	BPS 数1,{数2}	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12Cの通信速度設定 (単位kHz、0:デフォルト400kHz)
オーケー OK	OK {数}	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)
アイ・ツー・シー・リード I2CR	I2CR(数1, 数2, 数3, 数4, 数5)	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)
アイ・ツー・シー・ライト I2CW	I2CW(数1, 数2, 数3, 数4, 数5)	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)
アイ・オー・ティー・イン IOT.IN	IoT.IN()	sakura.ioモジュールから受信した数を一つ読み込む
アイ・オー・ティー・アウト IOT.OUT	IoT.OUT 数	sakura.ioモジュールへ数をチャンネル0で即時送信する
ダブル・エス・エル・イー・ディー WS.LED	WS.LED 数1,{数2}	配列の先頭から緑赤青の順に設定された値でOUT1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
スイッチ SWITCH	SWITCH {数1,{数2}}	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
ユーザー USR	USR(数1,{数2})	数1で指定されたアドレスのマシン語を数2をパラメータとして呼び出す (数2省略で0)