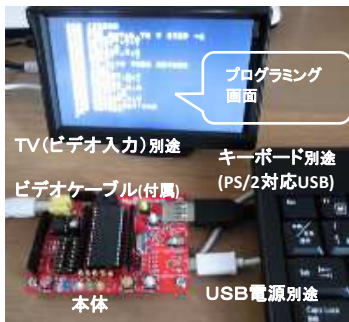


栃木工業高校が作ったIchigoJam[®]互換仕様
プログラミング専用パソコン(マイコンボード)
キーボード(PS/2)とテレビ(Video),USB電源で
BASIC言語などでプログラミング
プログラミングロボットカーにも発展できる

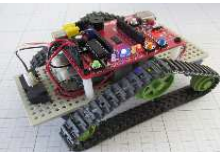
●仕様・特徴



【IchigoJam[®]仕様準拠】LPC1114 - ARM Cortex-M0搭載 32-bitマイコン
12MHzクロック/PS2キーボード入力/VIDEO-NTSC出力/入力SWx1/出力LEDx1
IchigoJam=BASICファームウェア正規ライセンス版搭載

【オリジナル仕様】

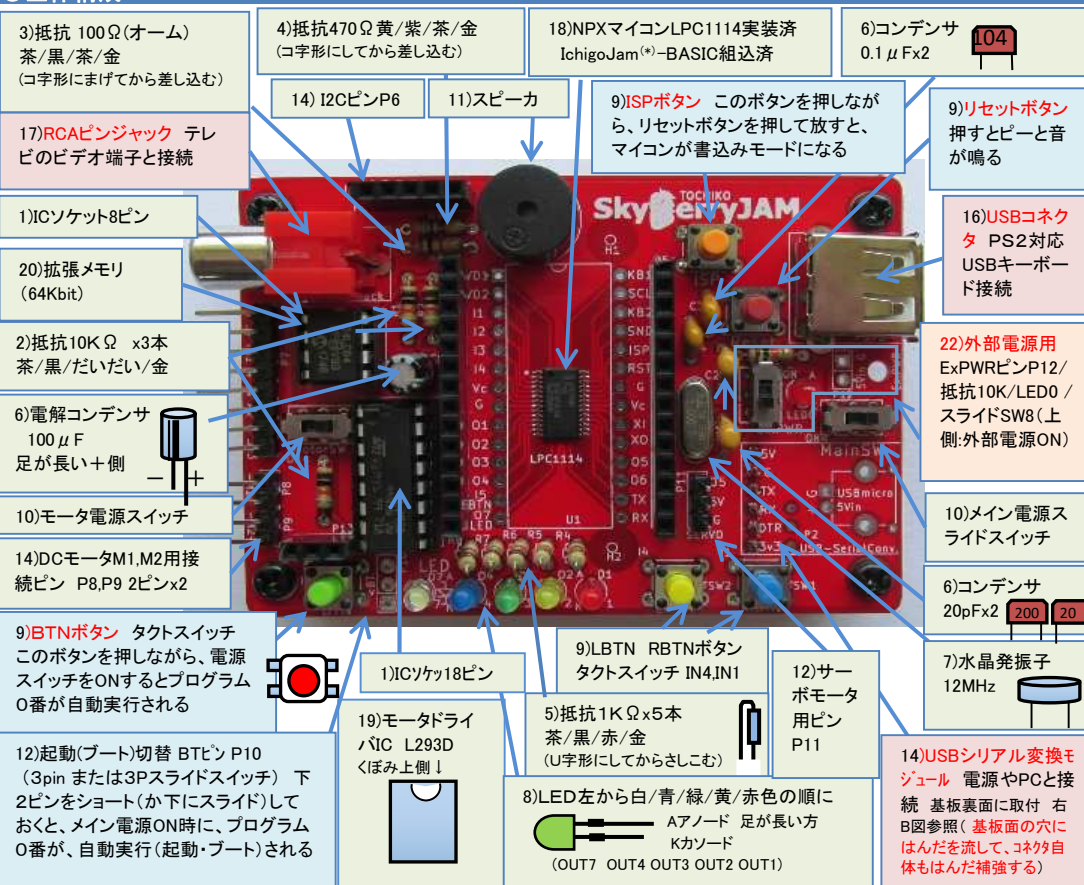
USBシリアル変換モジュール付属(CP2102)
拡張メモリEEPROM64Kbit付属(64K~512Kbit対応)
DCモータドライバIC L293D付属/入力SWx2
ISP-SW/RESET-SW/スピーカ/LEDx4
各センサー用端子/サーボモータ用端子
マイクロUSBケーブル・ビデオケーブル(特別付属)
←TVモニター・キーボード(PS/2)・USB電源(別途必要)で
BASIC言語(搭載)やJavaScript言語,C言語mbedなども可能



●製作に必要な工具

はんだごて・はんだ
ラジオペンチ・ニッパ
ドライバ などの工具

●全体構成



●部品表

【組立キット版】の場合は、以下の順番に従って背の低いものから、基板の記号のところへはんだづけしてください。

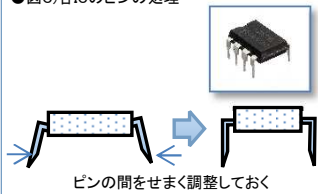
順番	部品名	数	記号	注意 (向きありに注意)	接続(*必須)
0	プリント基板	1	SkyBerryJAM基板		-
1	ICソケット(16P)	1	U2	向きあり くぼみに	*
	ICソケット(8P)	1	U3		*
2	抵抗器10KΩ 茶黒橙金	3	R1,R2,R3	コ字形に曲げておく	*
3	抵抗器100Ω 茶黒茶金	1	R9		*
4	抵抗器470Ω 黄紫茶金	1	R10		*
5	抵抗器1kΩ 茶黒赤金	5	R4,R5,R6,R7,R8	U字型に→	*
6	セラミックコンデンサ 20~22pF(200)	2	C1,C2	200 か 20	*
	セラミックコンデンサ 0.1μF(104)	2	C3,C4		104
	電解コンデンサ100μF	1	C5	向き+あり	*
7	水晶発振子 12MHz	1	Y1		*
8	LED 赤,黄,緑,青,白	各1	LED1,2,3,4,7	向きAKあり	*
9	タクトスイッチ	5	SW1,2,3,4,5	向きあり 色は自由に	*
10	スライドスイッチ	2	SW6,SW7	ピン先を斜めにカットしておくとうりやすい	*
11	スピーカ	1	SP	向き+不問	*
12	ピンヘッダ ストレート型	1	P2(6pin)	ラジオペンチなどで折る (余りピンあり)	*
			P10(3pin) P11(3pin) P12(3pin)		任意(自由)
13	ピンヘッダ L型 (またはストレート型)	1	P7(8pin)		任意(自由)
			P8,P9(計4pin)		
14	USBシリアル変換モジュール	1	P2	ストレートピンヘッダ接続(下図B)	*
15	ピンソケット 40pin	1	P4(14pin)	ニッパかカッターで切り分ける (下図A)(余りピンあり)	任意(自由)
			P5(14pin)		
			P6(5pin)		
16	USB Aジャック	1	P3		*
17	RCAピンジャック	1	P1	色は任意色	*
18	NPXマイコン LPC1114FDH28	1	U1IC	実装済	*
19	DCモータドライバL293D	1	U2IC	向きあり, 下図C (代用品SN754410NE)	任意(自由)
20	拡張メモリ EEPROM 64Kbit	1	U3IC	向きあり, 下図C	任意(自由)
21	スペーサ・ねじ	各4		基板の四すみ	*
22	スライドスイッチ	1	SW8	外部電源ExPWR(P12からの5V入力)を使用する場合は接続する *SW8:外部電源スイッチ (上側ON・LED0点灯/下側OFF)	任意(自由)
	抵抗器10KΩ 茶黒橙金	1	R13		
	LED	1	LED0		

備考 *各部品は代用品の場合があります/外部接続のためのピンヘッダやピンソケットは目的に応じて自由に接続してください。

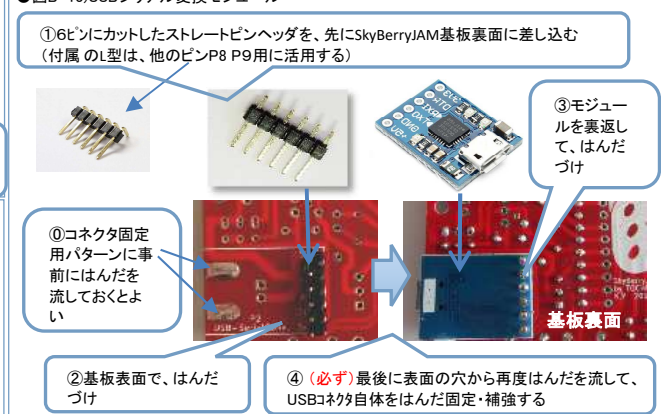
●図A 14)ピンソケットの分割例



●図C 各ICのピンの処理

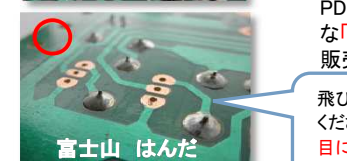
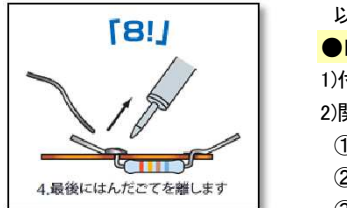
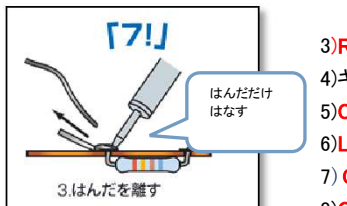
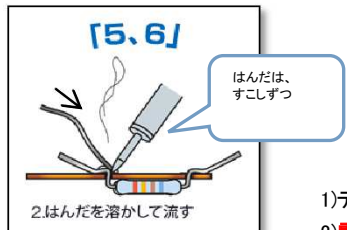
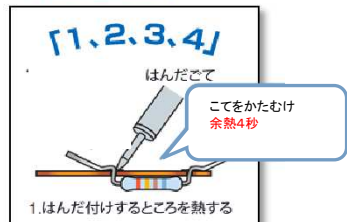


●図B 13)USBシリアル変換モジュール



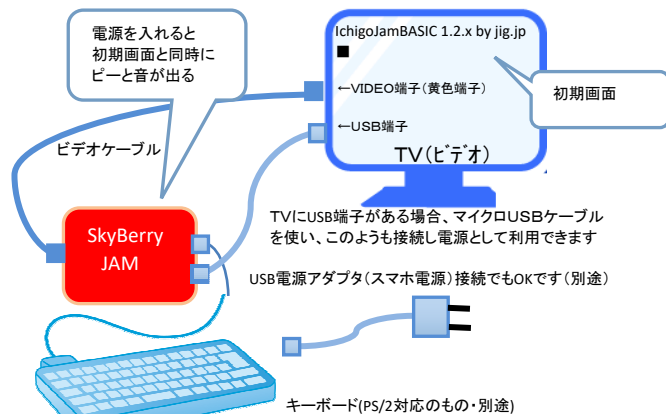
はんだづけの方法

* やけどに注意しましょう



●基本動作確認をしましょう

組立が終わりましたら、下図のように接続し、動作の確認を次のように行ってください



- 1) テレビ、キーボード、USB電源アダプタを上のように本体と接続する
 - 2) **電源スイッチをON**、**ビー**という音とともに、IchigoJamBASICの起動画面を確認する(上図)
(お使いのテレビによっては、本体との相性により画面が乱れる場合があります)
 - 3) **Resetボタン**を押して、ビーという音とともに初期画面が再び表示されるか確認する
 - 4) キーボードを打ち、文字が正しく画面に表示されるか確認する
 - 5) **GLS<ENTER>**と入力し、画面がクリアされるか確認する(クリアスクリーン)
 - 6) **LED 1<ENTER>**と入力し、一番左のLED7が点灯するか確認する
 - 7) **OUT 255<ENTER>**と入力し、LEDがすべて点灯するか確認する
 - 8) **OUT 0<ENTER>**と入力し、LEDがすべて消灯するか確認する
- 以上 いずれかが不動作の場合は、再度組立を確認してください

●BASICプログラミングをしてみましょう

- 1) 付属の資料ページ4「初級編」などを利用して、基本をマスターしてみましょう
- 2) 関連サイトから情報を得てみましょう
 - ① ichigojam公式サイト <http://ichigojam.net> プログラミング演習書や情報がDLできます
 - ② イチゴジャムレシピ <http://15jamrecipe.jimdo.com> 各種情報が得られます
 - ③ ポクにもわかる IchigoJam用 マイコンボード 専門的な情報が得られます
 - ④ 「栃木工業高校」公式サイト内 SkyBerryJAMページ (SKYBERRYJAMで検索可能)
- 3) IchigoJam 関連一般書籍から情報を得てみましょう
関連書籍も販売されています。

●応用編 ロボットやセンサとつないでみましょう

SkyBerryJAMは、モータ駆動ICを標準搭載していますので、小型の直流モータを2個までコントロールできます。また、各種センサやサーボモータも接続することができます。上記④サイト内に、ロボットレシピ(ロボットの作り方PDF)情報がUPされています。また右図のような「**プログラミングロボットカーベースキット**」も販売されていますのでご利用ください。

飛び出した部分は、根元からニッパで切ってください。(切った線が飛ぶ場合があります。目に入らないよう注意してください)



プログラミングロボットカーベースキット組立例

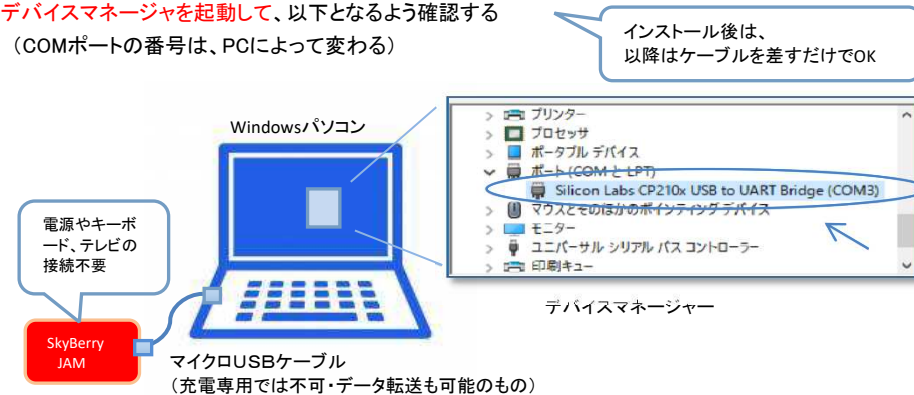
パソコンとSkyBerryJAMとの連携方法 (WindowsPCの場合)

SkyBerryJAMは、USBシリアル変換モジュールを標準搭載しているため、パソコンと連携することができます。

●1 USBシリアル変換モジュールドライバのインストール

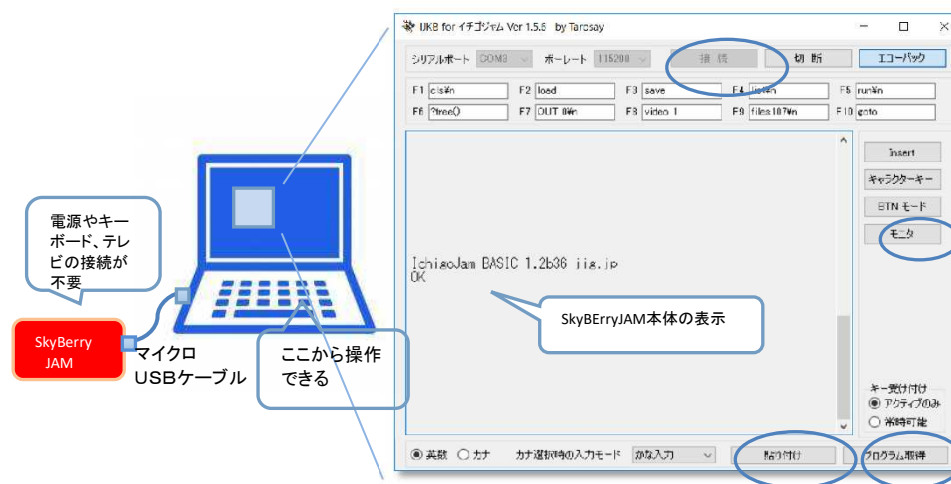
Silcon Labs 社 <http://jp.silabs.com/>

- 1) 「CP210x ドライバ」等として検索、同社のサイトよりダウンロードし、インストールする
- 2) マイクロUSBケーブルで本体のモジュールとPCを接続する。(接続音がある)
- 3) **デバイスマネージャを起動して**、以下となるよう確認する
(COMポートの番号は、PCによって変わる)



●2 ターミナル(遠隔)操作

パソコンと連携すると、パソコンのディスプレイやキーボードを使い、SkyBerryJAMを操作することができます (SkyBerryJAM本体に、電源やキーボード、テレビの接続が不要となります)



- 1) IJKB(IchigoJam用ターミナルソフトウェア・フリー)をダウンロードし、解凍
- 2) その中のSetUpを実行しインストールする
- 3) SkyBerryJAM本体の電源を入れ、上記ソフトを起動、接続ボタンをクリック
- 4) 本体のRESETボタンを押すと、ソフト画面に初期文字(上図参照)が出ればOK
- 5) <モニター>ボタンで、リアルな操作画面での操作も可能
- 6) <プログラム取得>や<貼り付け>を使うとパソコンとのファイルのやり取りが可能
- 7) その他のソフトウェアとして、IJUtilities (Windows・OS X・Linux版)やteratermなどがあります。

システムのバージョンアップや他のシステムへの切り替えは、次のように行います

●1 システムソフト(ファーム)ウェアのダウンロード(IchigoJam[®]-BASIC の場合)

- 1)「ichigojam」で検索、以下サイトichigojam.net からダウンロードする
- 2)PCの適当な場所へ保存して、解凍しておく

- 3)解凍したファイルのうち必要なファイルは以下の「ichigojam-ntsc-jpkbd.hex」
(*英語版キーボードを利用の場合は、「ichigojam-ntsc-uskbd.hex」)

名前	更新日時	種類	サイズ
_MACOSX	2017/08/05 15:44	ファイル フォルダ	
IchigoJam.html	2017/04/29 16:37	Firefox HTML Doc...	22 KB
IchigoJam-en.html	2017/04/29 16:37	Firefox HTML Doc...	18 KB
ichigojam-license.pdf	2015/06/16 12:02	Adobe Acrobat D...	138 KB
ichigojam-ntsc-jpkbd.bin	2017/07/27 13:47	BIN ファイル	24 KB
ichigojam-ntsc-jpkbd.hex	2017/07/27 13:47	HEX ファイル	68 KB
ichigojam-ntsc-uskbd.bin	2017/07/27 13:46	BIN ファイル	24 KB
ichigojam-ntsc-uskbd.hex	2017/07/27 13:46	HEX ファイル	68 KB
readme.txt	2017/07/27 13:47	テキスト ドキュメント	2 KB
readme-en.txt	2017/07/27 13:47	テキスト ドキュメント	2 KB

中・上級者向けとして
●IchigoLatte(JavaScript言語システム)

http://ichigo latte.shizentai.jp/
BASICシステムだけでなく、IchigoLatteという、JavaScript言語システムパソコンにすることもできます。ダウンロードして、解凍してください。次項のソフトウェアで、このHexファイルを指定して、SkyBerryJAMに書き込んでください。

●mbed (オンラインC言語)
ARM社が無料で公開しているmbedというオンラインでプログラミング開発を行うことができるシステムも活用できます。「mbed」で検索してみてください

●2 書き込みソフトウェア「フラッシュマジック」のダウンロード

- 1)「flashmagic」と検索してサイトhttp://www.flashmagictool.com/よりダウンロードして解凍する
- 2)実行して、下記のように LPC1114/102, COM番号、HEXファイルを先ほどのファイルに設定する

FlashMagic 各種設定

●3 マイコンLPC1114へ書き込み(アップデート)

- 1)SkyBerryJAM本体の電源ONして、ISPボタンを押したまま、RESETボタンを押しはなす
- 2)そうすると、本体のLED7が点灯する(マイコンが書き込みモードとなる)
- 3)上記ソフトの「Start」ボタンを押す 終わると一番下の行にFinishedと表示される
- 4)SkyBerryJAMのRESETボタンを押し、「ピッ」と音が鳴れば、正常書き込み、BASIC起動
- 5)テレビモニタと接続し、再度RESETボタンを押し、IchigoJamBASICの起動画面を確認する

関連情報サイト

「栃木工業高校」公式サイト内 SkyBerryJAM
ichigojam公式サイト http://ichigojam.net
イチゴジャムレシピhttp://15jamrecipe.jimdo.com
ボクにもわかる IchigoJam用 マイコンボード
IchigoLatte http://ichigo latte.shizentai.jp/



こどもぱそこんSkyBerryJAM 組立キット

設計・企画: 栃木県立栃木工業高等学校情報技術科 skyberryjam@tochigi-edu.ed.jp
生産・販売: VKSテック(栃木県宇都宮市) vks-tec@trade.gmob.jp
*IchigoJamは機ig.jpの登録商標です
*SkyBerryJAMおよびロゴマークは、栃木県立栃木工業高校の商標です
商品に関しまして、不備などがございましたら、上記メール宛ご連絡くだされば幸いです

●1) 一番左のLED7をひからせよう

LED 1<enter>

●2) LEDをけそう

LED 0<enter>

●3) LEDをひからせ、じどうでけそう

LED 1:LED 0<enter>

●4) LEDをひからせ、1秒待って、けす

LED 1:WAIT 60:LED 0<enter>

●5)うへの数字を2秒にしゅうせいしてみよう

LED 1:WAIT 120:LED 0<enter>

●6) LEDをずっとチカチカ点滅(てんめつ)させてみよう

10 LED 1:WAIT 60<enter>

20 LED 0:WAIT 60<enter>

30 GOTO 10<enter>

●7) プログラムぜんたいをがめんに出してかくにんしてみよう

LIST<enter>

●8)実行(じっこう)させてみよう

RUN<enter>

●9) プログラムの実行をとめてみよう

ESC(エスケープ)キーをおす

(かだい1) もっとはやくチカチカさせるにはどうする？

●10) つくったプログラムを保存(ほぞん)しておきましょう

SAVE 1<enter>

●11) リセットボタンをおすか、でんげんをきって、また入れてみる

●12) ファイル1をよみだしてみよう。

LOAD 1<enter>

●13)また実行(じっこう)してみよう

RUN<enter>

●14)また、ていしさせよう

ESC(エスケープ)キーをおす

●15)あたらしいプログラムをつくるときはけておく

NEW<enter>

コンピュータのけたのいみ

64 8 4 2 1

LED 7 4 3 2 1

<enter>でコマンド(命令)をコンピュータにおくります。

2つの命令を : (コロン)でつなぐ

でもうごきがはやくて
光りが見えない どうする？

WAIT 60 (待て 60は1秒)
を間に入れる

こんどは、うってもすぐにはうご
きません。あとでまとめてうご
かします。

10行目へ行け(もどれ)という命令=**くりかえし**

F4キーでもOK

RUNでまとめて実行
F5キーでもOK

びーと音になる

ファイルのばんごう
は、0から3,
100から107もOK。

プログラムは、きえるが、メモリーに保存しておけ
ば、またよびだせる

LIST(F4)でかくにん、RUN(F5)で実行、ESCで止める SAVE(番号) F3でほぞん LOAD(番号) F2でよびだし CLS F1で画面クリア

LED編 OUTコマンド 出力命令(しゅつりょくめいれい)

●1) 一番右はしのLED1をひからせよう

OUT 1<enter>

●2) そのとなりLED2をひからせよう

OUT 2<enter>

●3) またそのとなりLED3をひからせよう

OUT 4<enter>

●4) さらにとなりLED4をひからせよう

OUT 8 <enter>

●5) LEDをいっぺんにけそう

OUT 0 <enter>

●6) LED1~4をすべてひからせよう

OUT 8+4+2+1 <enter>

●7) OUT 10 <enter> をうつとどうなる？

●8) LEDをじどうで右からじゅんばんにひからせよう

NEW <enter>

10 OUT 1:WAIT 60 <enter>

20 OUT 2:WAIT 60 <enter>

30 OUT 4:WAIT 60 <enter>

40 OUT 8:WAIT 60 <enter>

50 OUT 0 <enter>

RUN <enter>

●9) プログラムの実行をとめて、プログラムをかくにんしておこう

ESC(エスケープ)キーをおして、LIST <enter>

(かだい1) ずっとくりかえしさせるにはどうする？ (60 GOTO 10<enter>をうつ)

(かだい2) LED3緑(3秒光る)LED2黄(1秒) LED1赤(3秒)として、
信号機プログラムにかいぞうしよう(光る時間もかえてみよう)



LED3 LED2 LED1

●10) ファイル2に(ほぞん)しておこう

SAVE 2 <enter>

●11) ファイル(107番まで)をかくにんしておこう

FILES 107<enter>

OUT 1, 1 でもOK

OUT 2, 1 でもOK だが...

OUT 3, 1 でもOK だが...

OUT 4, 1 でもOK だが...

OUT 1, 0などで一つずつ消す
いっぺんにはできない

OUT 1, 1などで一つずつ点ける
いっぺんにはできない

OUT 15 でもOK

8 4 2 1

10 → LED ○○○○

10 = ()

人間の世界

つく●を1で
つかない○を0で書こう
コンピュータの世界

F4でもOK ミスがあればしゅうせい

IchigoJam BASIC リファレンス ver 1.2.4

キーボード操作

操作	解説
キー	文字を入力する
Shift / シフト	キーと共に押し記号や小文字などを入力する
カタカナ	アルファベットとカタカナ（ローマ字入力）を切り替える（CTRL+SHIFT / コントロール+シフトでも可）
Enter / エンター	コマンドを実行する（プログラム変更時その行でEnterキー）
Shift+Enter / シフト+エンター	行を分割する
ESC / エスケープ	プログラムの実行、リスト表示、ファイル一覧表示を止める
カーソルキー	カーソルキーを移動する
Backspace / バックスペース	カーソルの前の文字を消す
Delete / デリート	カーソルにある文字を消す
ALT / オルト	0-9/A-Zと合わせて押すことで拡張文字入力（SHIFT押しながらで切り替え）、' 'と合わせて押して'_'、']'と合わせて押して'¥'や'¥'の入力（無変換でも可）
Home End / ホーム エンド	カーソルを行頭へ移動、カーソルを行末へ移動
Page Up Page Down / ページアップ ページダウン	カーソルを画面上へ移動、カーソルを画面下へ移動
Caps / キャップス	大文字と小文字を切り替える
Insert / インサート	キーボードの上書きモード/挿入モードを切り替える（CTRL+ALTでも可能）
ファンクションキー	F1:画面クリア、F2:LOAD、F3:SAVE、F4:LIST、F5:RUN、F6:?FREE()、F7:OUT0、F8:VIDEO1、F9:FILES、F10:SWITCH
ボタン	押しながら起動でFILE0を自動実行する

初級コマンド

コマンド	解説	例
LED 数 / エルイーディー	数が1なら光り、0なら消える	LED 1
WAIT 数1{,数2} / ウェイト	数1の数値フレーム分待つ 60で約 1 秒、省略可の数2指定で低電力化、数1のマイナス指定で走査線分待つ(-261でWAIT1と同等)	WAIT 60
: / コロン	コマンドを連結する	WAIT 60:LED 1
行番号 コマンド	プログラムとしてコマンドを記録する（1~32767まで、標準表示は16384まで）	10 LED1
行番号	指定した行番号のプログラムを消す	10
RUN / ラン	プログラムを実行する [F5]	RUN
LIST {行番号1{,行番号2}} / 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 行番号 / ゴートゥー	指定した行番号へ飛び（式も指定可能）	GOTO 10
END / エンド	プログラムを終了する	END
IF 数 {THEN} 次1 {ELSE 次2} / イフ・ゼン・エルス	数が0でなければ次1を実行し、0であれば次2を実行する（THEN,ELSE以降は省略可）	IF BTN() END
BTN({数}) / ボタン	ボタンが押されていれば1、そうで無いとき0を返す（数：0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE、省略で0）	LED BTN()
NEW / ニュー	プログラムを全部消す	NEW
PRINT {数や文字列} / プリント	文字を表示する（文字列は"で囲む、;"で連結できる）省略形：?	PRINT "HI!"
LOCATE 数,数 / ロケート	次に文字を書く位置を横、縦の順に指定する（縦=-1で無表示）省略形：LC	LOCATE 3,3
CLS / クリア スクリーン	画面を全部消す	CLS
RND(数) / ランダム	0から数未満の正数をランダムに返す	PRINT RND(6)
SAVE {数} / 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 {数} / 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 {数1{,数2}} / 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 {数1{,数2}} / ビープ	BEEPを鳴らす 周期(1-255)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必要	BEEP
PLAY {MML} / プレイ	MMLで記述した音楽を再生する MML省略で停止 ※SOUND(EX2)-GNDに圧電サウンダーなどの接続必要（次項のMML参照）	PLAY "\$CDE2CDE2"
TEMPO 数 / テンポ	再生中の音楽のテンポを変更する	TEMPO 1200
数 + 数	足し算する	PRINT 1+1
数 - 数	引き算する	PRINT 2-1
数 * 数	掛け算する	PRINT 7*8
数 / 数	割り算する（小数点以下は切り捨て）	PRINT 9/3
数 % 数	割り算した余りを返す	PRINT 10%3
(数)	カッコ内を優先して計算する	PRINT 1+(1*2)
LET 変数,数 / レット	アルファベット 1文字を変数として数の値を入れる（配列に連続代入可能 LET[0],1,2）省略形：変数=数	LET A,1
INPUT {文字列},変数 / 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（約1/60秒で1進む）	PRINT TICK()
CLT / クリア ティック	時間をリセットする	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()
定数	LEFT=28, RIGHT=29, UP=30, DOWN=31, SPACE=32	IF INKEY()=SPACE LED1
CHR\$(数) / キャラ	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("文字") / アスキー	文字に対する文字コードを返す	PRINT ASC("A")
SCROLL 数 / スクロール	指定した方向に1キャラクター分スクロールする（0/UP:上、1/RIGHT:右、2/DOWN:下、3/LEFT:左）	SCROLL 2
SCR({数},数) / 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） 別名：VPEEK	PRINT SCR(0,0)
数 = 数	比較して等しい時に1、それ以外で0を返す（==でも可）	IF A=B LED 1
数 <> 数	比較して等しくない時に1、それ以外で0を返す（!=でも可）	IF A<>B LED 1
数 <= 数	比較して以下の時に1、それ以外で0を返す	IF A<=B LED 1
数 < 数	比較して未満の時に1、それ以外で0を返す	IF A<B LED 1
数 >= 数	比較して以上の時に1、それ以外で0を返す	IF A>=B LED 1
数 > 数	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
式 AND 式 / アンド	どちらの式も1の時に1、それ以外で0を返す（&&でも可）	IF A=1 AND B=1 LED 1
式 OR 式 / オア	どちらかの式が1の時に1、それ以外で0を返す（ でも可）	IF A=1 OR B=1 LED 1
NOT 式 / ノット	式が0の時に1、それ以外で0を返す（!でも可）	IF NOT A=1 LED 1
REM / リマーク	これ以降の命令を実行しない（コメント機能）省略形：'	REM START
FOR 変数=数1 TO 数2 {STEP 数3} NEXT / フォー・トゥー・ステップ・ネクスト	変数に数1をいれ、数2になるまで数3ずつ増やしながらかえす（STEPは省略可、6段まで）	FOR I=0 TO 10:?:I:NEXT
OUT 数1{,数2} / 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({数}) / 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({数}) / 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 数1,数2,{数3} / 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

MML (PLAYコマンド内)

コマンド	解説	例
音	音(C D E F G A B /ドレミファソラシ)を鳴らす (Rは休符、スペースはスキップされる)	CDER FG
音n	長さを指定して音を鳴らす (.を付けると半分の長さ分伸びる)	C4 E2. D1 F32
音+	半音上げる	C+ D+
音-	半音下げる	D- E-
Tn	テンポ (TEMPO命令で後から変更可能) 初期値:120	T96CDE
Ln	長さ指定しないと時の長さ(1,2,3,4,8,16,32) 初期値:4	CL8DC
On	オクターブ指定 O1C(低音)からO5B(高音)まで 初期値:3	O3CO2C
<	オクターブ上げる (ver1.1と逆なので注意)	C<C<C
>	オクターブ下げる (ver1.1と逆なので注意)	C>C>C
\$	これ以降のMMLを繰り返す (BGMに便利)	C\$DE
Nn	1-255 音の高さ指定してLで指定した長さで鳴らす (BEEP命令と同じ)	N10N5
'	以降のMMLを鳴らさない	C'DE

上級コマンド

コマンド	解説	例
CLV / クリア バリアブル	変数、配列を全部0にする 別名: CLEAR	CLV
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(数) / 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)
[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	[3]=1
GOSUB 行番号 RETURN / ゴーサブ・リターン	数または式で指定した行番号に飛び、RETURNで戻ってくる 省略形:GSB (30段まで)/RTN	GOSUB 100
DEC\$(数,{数}) / デシ	数を文字列にする (2番目の数は桁数、省略可)	?DEC\$(99,3)
#16進数	16進数で数を表記する	#FF
HEX\$(数,{数}) / ヘックス	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)
`2進数	2進数で数を表記する	`1010
BIN\$(数,{数}) / バイナリー	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)
数 & 数	論理積 (ビット演算)	?3&1
数 数	論理和 (ビット演算)	?3 1
数 ^ 数	排他的論理和 (ビット演算)	?A^1
数 >> 数	右シフトする (ビット演算)	?A>>1
数 << 数	左シフトする (ビット演算)	?A<<1
~数	ビット反転 (ビット演算)	?~A
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()
LANG() / ランゲージ	IchigoJam BASICの言語番号を返す (1:日本語, 2:モンゴル語, 3:ベトナム語)	?LANG()
RENUM {数1,{数2}} / リナンバー	プログラムの行番号を数1から数2刻みにする (数省略で10、GOTO/GOSUBの飛び先は手で変更必要)	RENUM
LRUN {数} / 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 数 / エスランド	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(数) / 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 数,数 / 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 数1,数2,数3 / 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
"文字列"	文字列の先頭アドレスを返す	A="ABC"
STR\$(数1,{数2}) / ストリング	文字列を返す (数2(省略可)で長さ指定)	PRINT STR\$(A)
LEN("文字列") / レングス	文字列の長さを返す	PRINT LEN("ABC")
@ラベル / 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 数1,{数2} / 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 数1,{数2} / 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、初期値:2)、数2:シリアル受信設定 (0:オフ、1:オン 省略時)	UART 0
BPS 数1,{数2} / 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以下で-100倍bps)、I2C通信速度を変更する(kHz) (0で初期値の400kHz、1kHz-6000kHz)	BPS 9600
OK {数} / 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、2でメッセージOFF、省略はON)	OK 2
I2CR(数1,数2,数3,数4,数5) / アイツーシー	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,#700,2,#702,2)
I2CW(数1,数2,数3,数4,数5) / アイツーシー	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,#700,2,#702,2)
SWITCH {数} / スイッチ	画面表示をテレビと液晶とを切り替える(0:テレビ、1:液晶)	SWITCH
USR(数1,数2) / ユーザー	アドレス数1のマシン語を数2をパラメータとして呼び出し (注意! 高確率でIchigoJamが停止する)	A=USR(#700,0)

演算子優先順位

優先順	演算子	備考
1	()	カッコ
2	- ~ ! NOT	単項演算子 (マイナス、ビット反転、論理否定)
3	* / % MOD << >> & ^	掛け算・割り算・除算・ビットシフト・ビットアンド・ビットXOR
4	+ -	足し算・引き算・ビットオア
5	= != < > <= >=	論理比較