

Licensed by **IchigoJam**
 栃木工業高校が作ったIchigoJam®互換仕様
プログラミングパソコン
 BASIC言語,JavaScript,C言語などでプログラミング
 ロボットプログラミングにも
 ビジュアル・プログラミング(BlocklyJam)対応



■組立キット版 / ■完成品版

●仕様・特徴

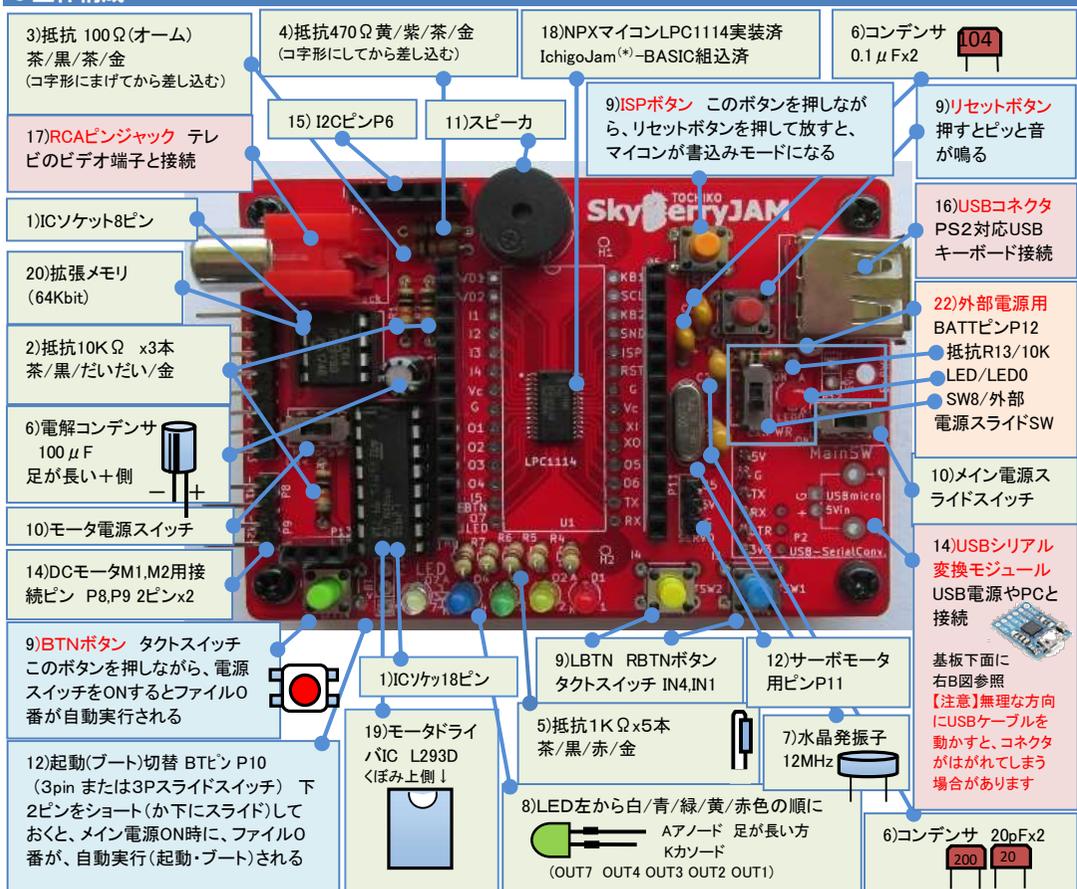


【IchigoJam* 仕様準拠】LPC1114-ARM Cortex-M0 32-bitマイコン etc.
SkyBerryJAM版IchigoJamBASIC搭載
 (正規ライセンス・初期テストプログラム内蔵版)
 ◀ロボット組込み応用例

【オリジナル仕様】
 USBシリアル変換モジュール付属(CP2102)
 拡張メモリエEPROM64Kbit付属(64K~512Kbit対応)
 DCモータドライバL293D付属/入力SWx2/LEDx4 etc.
 マイコンUSBケーブル・ビデオケーブル(試供版・特別付属)
ビジュアルプログラミング(BlocklyJam)対応

◀BASIC言語(搭載)ほかJavaScript言語,C言語mbedなど可能
●別途必要なもの
 T Vモニター・キーボード(PS/2対応のUSB型)・USB 5V電源

●全体構成



●製作に必要な工具 はんだごて・ラジオペンチ・ニッパ・ドライバなど *製作動画Youtubeにて配信中

●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。

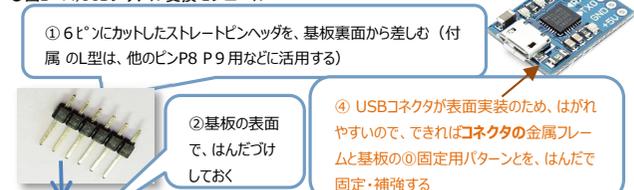
順番	部品名	数	記号	注意 (向きありに注意)	接続(*必須)
0	プリント基板	1	SkyBerryJAM基板		-
1	ICソケット(16P)	1	U2	*オプション (付属の場合もあり)	*
	ICソケット(8P)	1	U3		*
2	抵抗器10KΩ茶黒橙金	3	R1,R2,R3	コ字形に曲げておく	*
3	抵抗器100Ω茶黒茶金	1	R9		*
4	抵抗器470Ω黄紫茶金	1	R10		*
5	抵抗器1kΩ 茶黒赤金	5	R4,R5,R6,R7,R8	U字型に→	*
6	セラミックコンデンサ 20~22pF	2	C1,C2	200か 20 104	*
	セラミックコンデンサ 0.1μF	2	C3,C4		*
	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	任意(自由)
		1	P11(3pin)		
		1	P7(8pin)		
13	ピンヘッダL型(ストレート型)	1	P8,P9(計4pin)	14)の付属品利用	任意(自由)
		1	P12(2pin)		
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に接続(下図B) ニッパかカッターで切り分ける	*
		1	P4(14pin)		
15	ピンソケット	1	P5(14pin)	(下図A) (余りピンあり)	任意(自由)
		1	P6(5pin)	それぞれ単独pinの場合もあり	
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 (直接はんだ付け可)	任意(自由)
20	拡張メモリ EEPROM 64Kbit	1	U3IC	向きあり, 下図C 試供品	任意(自由)
21	スパーサ・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BATT (P12からの5V入力)を使用する場合は接続する *SW8: 外部電源バッテリーBATT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

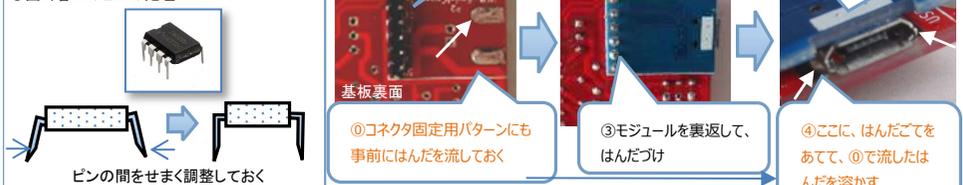
●図A 15)ピンソケットの分割例



●図B 14)USBシリアル変換モジュール



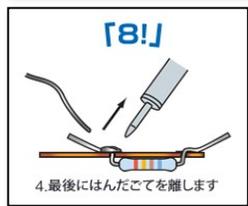
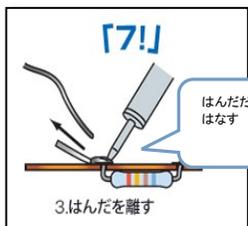
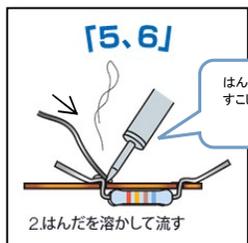
●図C 各ICのピンの処理



注意 *先がとがっているものがあるので、手にささないように気をつけてください。*やけどしないよう注意してください。

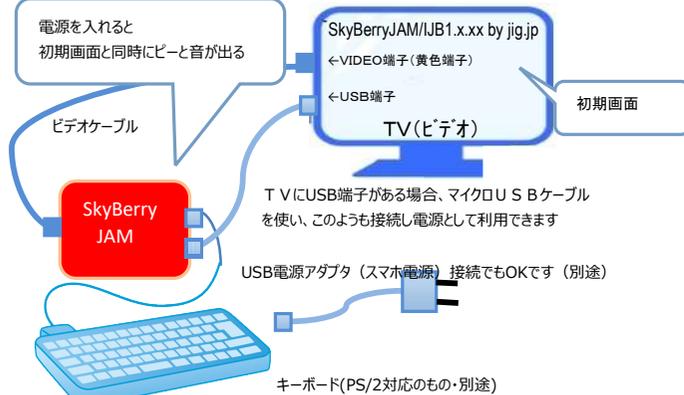
はんだづけの方法

* 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**、ピーという音とともに、IchigoJam BASIC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**、ピ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

*初期テストプログラム内蔵ファームが組み込まれている場合 以下が可能

- ・<BTN>ボタンを押しながら、電源スイッチを入れ直すと、ファイル0が自動で起動
- ・LEDが右から順に点灯（音階も出る）永久くつかし → **Reset**ボタンで停止

- 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
 - 6) **LED 1<ENTER>** と入力し、一番左のLED7が点灯するか確認する
 - 6) **LED 0<ENTER>** と入力し、一番左のLED7が消えるか確認する
- 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 ボクにもわかる 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 ロボットやセンサを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。
（切った線が飛び場場合があります。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/>

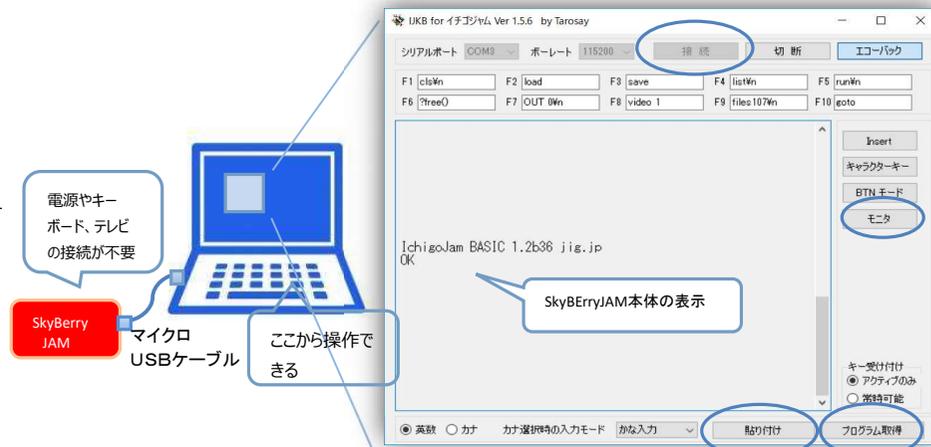
- 1) 「CP210x ドライバ」等として検索、同社のサイトよりダウンロードし、インストールする
- 2) マイクロUSBケーブルで本体のモジュールとPCを接続する。（接続音がる）
- 3) **デバイスマネージャを起動して**、以下となるよう確認する
（COMポートの番号は、PCによって変わる）



電源やキーボード、テレビの接続不要
SkyBerry JAM
マイクロUSBケーブル
（充電専用では不可・データ転送も可能なもの）

●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) I J K B (IchigoJam用ターミナルソフトウェア・フリーソフト) 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。
IJUtilitiesは、次のページ（システムソフトウェアのバージョンアップ方法の●2）を参照してください

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

IchigoJam ダウンロード

利用規約も必ず読んでください。目的のバージョンをダウンロード（仕様が変更される場合があります **バージョンアップは自己責任でおこなってください**）

いつでもバージョンアップ・変更が可能

SkyBerry JAM

利用規約（日本語、英語）に同意の上、ダウンロードしてください。第三者への配布にはライセンスが必要となります。（詳しくはこちら）

IchigoJam ver 1.4シリーズ

三角関数や描画への対応、要望の多かったJP/USキーボード配列の変更に対応

ver 1.4.3 / 栃木工業高校による互換機SkyBerryJAM（スカイベリージャム）1,000ライセンス記念ファームウェア

ver 1.4.2

ver 1.4.1

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

名前	更新日時	種類	サイズ
skyberryjam-lcd-jpkbd-rom-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB
skyberryjam-lcd-uskbd.bin	2020/09/03 17:20	BIN ファイル	24 KB
skyberryjam-lcd-uskbd-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB
skyberryjam-lcd-uskbd-rom.bin	2020/09/03 17:20	BIN ファイル	24 KB
skyberryjam-lcd-uskbd-rom-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB
skyberryjam-ntsc-jpkbd.bin	2020/09/03 17:20	BIN ファイル	24 KB
skyberryjam-ntsc-jpkbd-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB
skyberryjam-ntsc-jpkbd-rom.bin	2020/09/03 17:20	BIN ファイル	24 KB
skyberryjam-ntsc-jpkbd-rom-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB
skyberryjam-ntsc-uskbd.bin	2020/09/03 17:20	BIN ファイル	24 KB
skyberryjam-ntsc-uskbd-preset.bin	2020/09/03 17:20	BIN ファイル	25 KB

(****: preset は初期テストプログラム付き版 rom:ファイル0-3書き込み保護機能版)

ビジュアル・プログラミング

●BlocklyJamというブロックプログラミングツール

このソフトは、SkyBerryJAM本体とPCをUSBケーブルで接続、PC画面にマウスで「ブロック」を並べていくだけで、プログラミング学習ができるというシステムです。(鴻池泰元氏開発・公開)そのシステム用のSkyBerryJAM専用「コマンド・ブロック」やそのインストール方法が、栃木工業高校の公式サイト内に公開されています。

これを使うと、LEDの点滅など基本プログラムから、センサを使ったロボットプログラミングまで、マウス一つでプログラミング可能となります。



●2 書き込み方法「IJUtilities」ソフトウェアの利用

- 1)「IJUtilities」と検索して、ダウンロード、解凍する。
- 2)解凍したフォルダ中より、IJUtilities.exe を実行する。(▼以下)

ターミナル

Files / Edit:

Utilities:

References:

Web:

ターミナルセンター

ファイル 編集 オプション ウィンドウ ヘルプ

スキャン

接続

Port: COM3

Firm: C:\DOC~20\SkyBerryJAM\firm\skyberryjam-1.4.3\skyberryjam-ntsc-jpkbd-preset.bin

インストール: Start

書き込み状況の表示

●3 マイコンLPC1114へ書き込み（アップデート）

- 1)SkyBerryJAM本体とPCをUSBケーブルで接続する
- 2)接続音が鳴る（ならない場合、デバイスマネージャでUSBシリアルモジュールCP210xのドライバーソフト、COMなど確認 前ページ●1参照）
- 3)＜電源スイッチ＞ONして、＜ISPボタン＞を押したまま、＜RESETボタン＞を押し・はなす
- 4)SkyBerryJAM本体のLED 7が薄く点灯する（マイコン自体が、ISP書き込みモードとなる）
- 5)上記ソフトの「スキャン」ボタンを押し、COM x が出るか確認する
- 6)左の＜Firm書換え＞ボタンを押す。右側のWindowsの＜選択＞ボタンから書き込むファイルを選択する
- 7)＜スタート＞ボタンで書き込む。USBシリアルモジュールのLEDが点滅しながら、上記ソフトに書き込み状況が表示される
- 8) 正常に書き込みが終わると、SkyBerryJAM本体から「ピッ」と起動音がでる

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://ichigolatte.shizentai.jp/>



子どもパソコンSkyBerryJAM 組立キット版/完成品版

設計・企画：栃木県立栃木工業高等学校 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は関jig.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が管理する登録商標または商標です
 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い幸いです

- 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>

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)で合体

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>になる

- 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>

F4キーでもOK

- 8)実行(じっこう)させてみよう

RUN<enter>

RUNでまとめて実行 F5キーでもOK

- 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

びーと音になる

(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> ファイル 1 にほぞん

番号→

メモリファイルのばんごうは、0から3、100から107

- 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、かくにんしてみよう。

LOAD 1<enter>

LIST<enter>

プログラムは、いったんきえるが、メモリに保存しておけば、またよびだせる

- 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しておく

NEW<enter>

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

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>

OUT 1, 1 でもOK

- 2) そのとなりLED2をひからせよう

OUT 2<enter>

OUT 2, 1 でもOK だが...

- 3) またそのとなりLED3をひからせよう

OUT 4<enter>

OUT 3, 1 でもOK だが...

- 4) さらにとなりLED4をひからせよう

OUT 8 <enter>

OUT 4, 1 でもOK だが...

- 5) LEDをいっぺんにけそう

OUT 0 <enter>

OUT 1, 0などで一つずつ消す
OUT 0でいっぺんにもけせる

- 6) LED1~4をすべてひからせよう

OUT 8+4+2+1 <enter>

OUT 1, 1などで一つずつ点ける

OUT 15 でもOK

8 4 2 1

- 7) OUT 10 <enter> をうつとどうなる?

10 → LED OOOO

10 = ()

- 8) LEDをじどうで右からじゅんばんんにひからせよう

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>

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

- 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>

F4でもOK ミスがあればしゅうせい

(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>

0~3はCPU本体のメモリ 100以降は拡張メモリEEPROM

IchigoJam BASIC リファレンス ver 1.4

コマンド	解説	例
LED 数 / 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 数1{,数2} / 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
行番号 {コマンド} / 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 {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の時終わりで表示、ESCで途中停止)	LIST 10,300
GOTO 行番号 / ゴートウ	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 数 {THEN} コマンド1 {ELSE コマンド2} / 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、ただし16進数後は:で区切る必要あり)	IF BTN() END
IF 数 {THEN} コマンド1 {ELSE コマンド2} / 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、ただし16進数後は:で区切る必要あり)	IF BTN() THEN END
IF 数 {THEN} コマンド1 {ELSE コマンド2} / 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、ただし16進数後は:で区切る必要あり)	IF BTN() END ELSE CONT
BTN(数) / ボタン	ボタンが押されていれば1、そうでないとき0を返す (数: 0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE/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 {数や文字列} / 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 "HI!";15
? {数や文字列} / 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)	? "HI!";15
LOCATE 数1{,数2{,数3}} / 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 数1{,数2{,数3}} / 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(数) / 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 {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	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 {"MML"} / 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 数 / テンポ	再生中の音楽のテンポを変更する	TEMPO 1200
数 + 数 / プラス	足し算する	PRINT 1+1
数 - 数 / マイナス	1. 引き算する 2. 後に続く数をマイナスにする	PRINT 2-1
数 * 数 / アスタリスク	掛け算する	PRINT 7*8
数 / 数 / スラッシュ	割り算する (小数点以下は切り捨て)	PRINT 9/3
数 % 数 / パーセント	割り算した余りを返す	PRINT 10%3
{式} / 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 変数,数 / 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
変数=数 / 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 {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に入れる (文字列とコンマは省略可)	INPUT "ANS?",A
TICK({数}) / 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\$(数{,数2{,数3...}}) / 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("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)	PRINT SCR(0,0)
VPEEK({数,数}) / 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)
数 == 数 / 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
数 != 数 / 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
数 <> 数 / 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
数 <= 数 / 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
数 < 数 / 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
数 >= 数 / 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
数 > 数 / 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
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	IF A=1 AND B=1 LED 1
式 && 式 / 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
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	IF A=1 OR B=1 LED 1
式 式 / 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 式 / 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
! 式 / 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 変数=数1 TO 数2 (STEP 数3):コマンド:NEXT / 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
FOR 変数=数1 TO 数2 (STEP 数3):コマンド:NEXT / 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

FOR 変数=数1 TO 数2 STEP 数3:コマンド:NEXT / 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
FOR 変数=数1 TO 数2 {STEP 数3}:コマンド: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({数}) / 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 数1,数2,{数3,数4},{数5} / 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(数1,数2) / 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 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	外部入力の数値(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
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(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] / アレイ	配列 ([0]から[10]までの1023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 / 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 行番号 / 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\$(数1,{数2}) / 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)
#16進数 / 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\$(数1,{数2}) / 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)
`2進数 / バック・クォート	2進数で数を表記する	`1010
BIN\$(数1,{数2}) / 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)
数 & 数 / アンパサンド	論理積 (ビット演算)	?3&1
数 数 / パイプ	論理和 (ビット演算)	?3 1
数 ^ 数 / ハット	排他的論理和 (ビット演算)	?A^1
数 >> 数 / シフト・ライト	右シフトする (ビット演算)	?A>>1
数 << 数 / シフト・レフト	左シフトする (ビット演算)	?A<<1
~数 / チルダ	ビット反転 (ビット演算)	?~A
COS(数) / 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(数) / 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を返す	?SOUND()
FREE() / フリー	プログラムの残りメモリ数を返す	?FREE()
VER({数}) / バージョン	IchigoJam BASICのバージョン番号を返す	?VER()
RENUM {数1,{数2}} / リナンパー	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が必要な場合がある	RENUM
LRUN {数1,{数2}} / ロードラン	ファイル数1のプログラムを読み込み後、実行する(数2が指定されていたらその行へGOTO)	LRUN 1
FILE() / ファイル	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	?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(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" / ダブル・クォート	文字列の先頭アドレスを返す	A="ABC"
STR\$(数1,{数2}) / 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("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、+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 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 {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、1:表示、2:非表示、省略で1)	OK 2
I2CR(数1,数2,数3,数4,数5) / 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(数1,数2,数3,数4,数5) / 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・ディー・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 数 / アイ・オー・ディー・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 数1,{数2} / 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 {数1,{数2}} / 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(数1,{数2}) / 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)