IchigoJam（いちごじゃむ）でプログラミング

Mission CARD（ミッションカード）POSSIBLE!

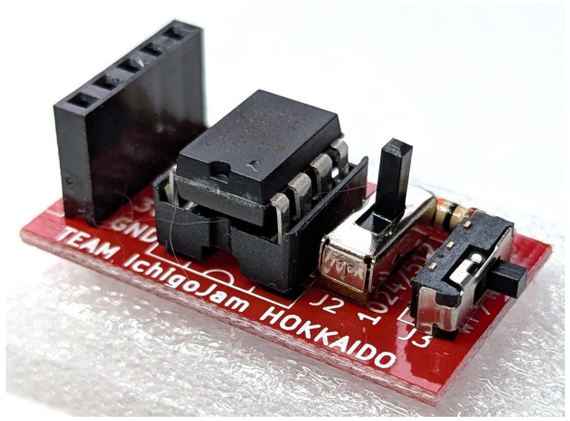
君だけの小っちゃなコンピュータ。

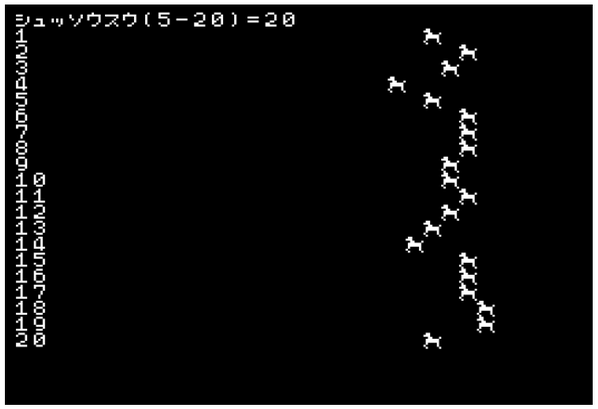
わくわくドキドキ

外部メモリを作ってみよう

IchigoJam

PCN





2025.06.08 Ver.5.2

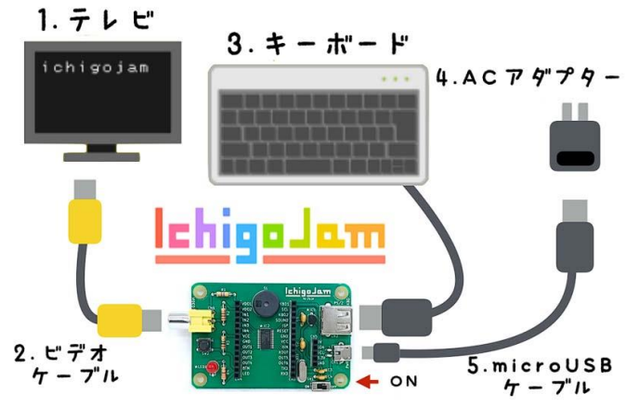
・本資料はNPO法人NEXTDAYの協力のもとNPO法人小樽青少年の科学の芽を育てる会が作成しました。
・資料の作成にあたり以下の資料を参照しました。
＞親子でベーシック入門 IchigoJamではじめてのプログラミング （出版社: ジャムハウス）
＞IchigoJamでプログラミング （発売：プログラミングクラブネットワーク）
・原稿についてはICHIGOJAM開発者**福野 泰介様**のブログを参照させて頂いてます。
・上田市マルチメディア情報センター 斎藤　史郎 様の講習会テキストを参照させて頂いています。
・IchigoJam Advent Calendar 参加プログラムを参照させて頂いています。

小樽別院　寺子屋教室
主催　小樽青少年の科学の芽を育てる会
協力　浄土真宗本願寺派 本願寺小樽別院
協力　NPO法人NEXTDAY

IchigoJamを動かそう！

・IchigoJamパソコンの部品配置と名前を覚えよう

液晶モニター



PS/2キーボード

ビデオ端子

キーボード端子

タクトスイッチ

電源端子

マイクロUSB
5V

LED

I/O
端子

I/O
端子

入　　切

電源スイッチ

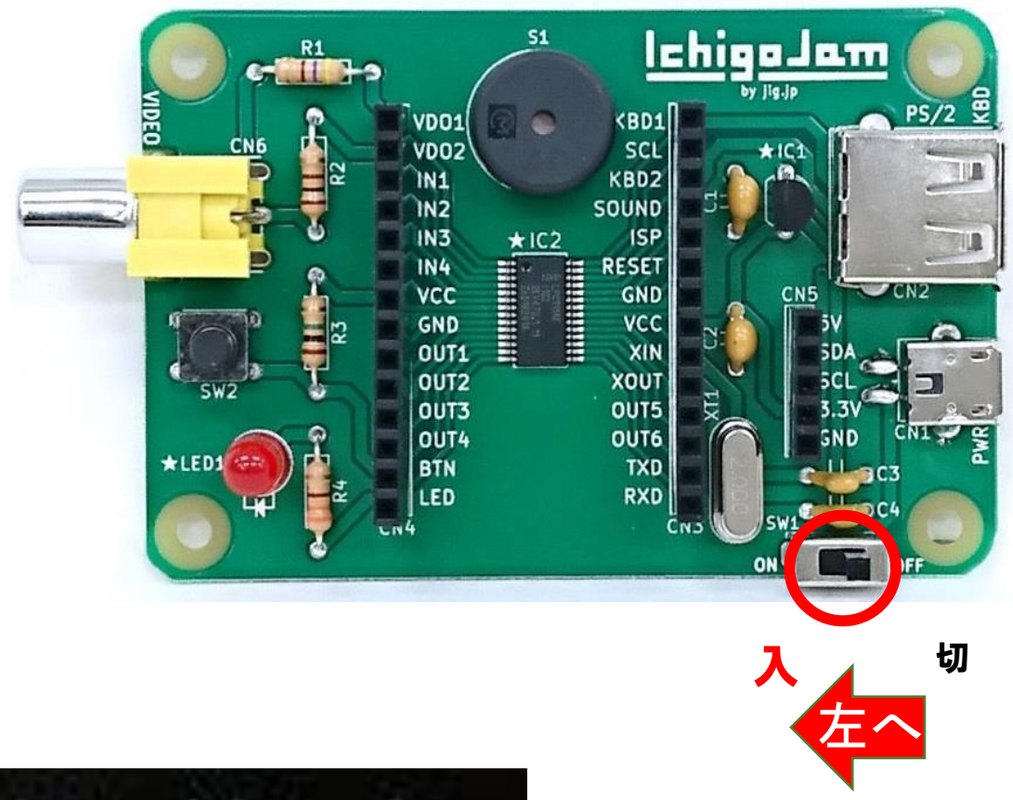
◆電源スイッチを入れよう

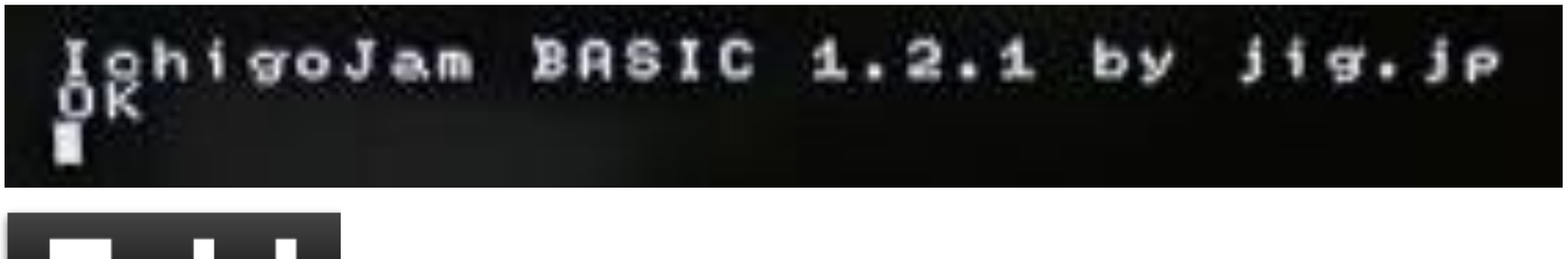
a. 電源スイッチを　右から左　へ動かす

左へ

b. 起動すると画面に文字が表示されます

ピッ！　と起動音が鳴ります。





命令（コマンド）が正しく動作したときに表示されます。

IchigiJamから 「わかったよ！」「動いたよ！」 と話して（表示して）います。

シンタックス　エラー

「分からないよ！！」 時

IchigoJam　って　どんなパソコン　！

IchigoJamは手のひらに乗せられる大きさの、プログラミング専用こどもパソコンです。

初心者向けプログラミング言語BASICを使っているので、誰でも気軽にプログラミングを始めることができます。
部品をはんだ付けして自分で組み立てることで、電子工作も体験できます。

直接命令を実行させたり、プログラムを記憶させて実行する事ができます。

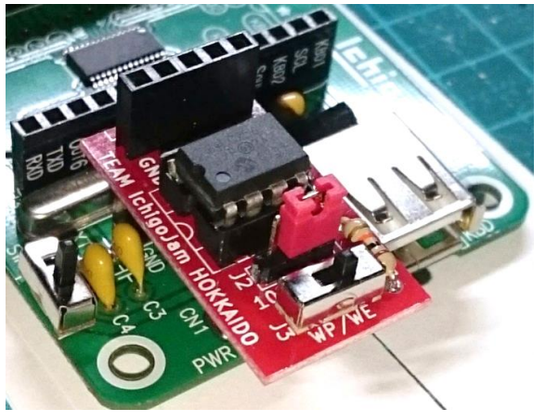
プログラムが短くて気軽に遊べるゲームがつくれます。
どんどんゲームを作って遊んでいると、プログラミングが身に付きます。

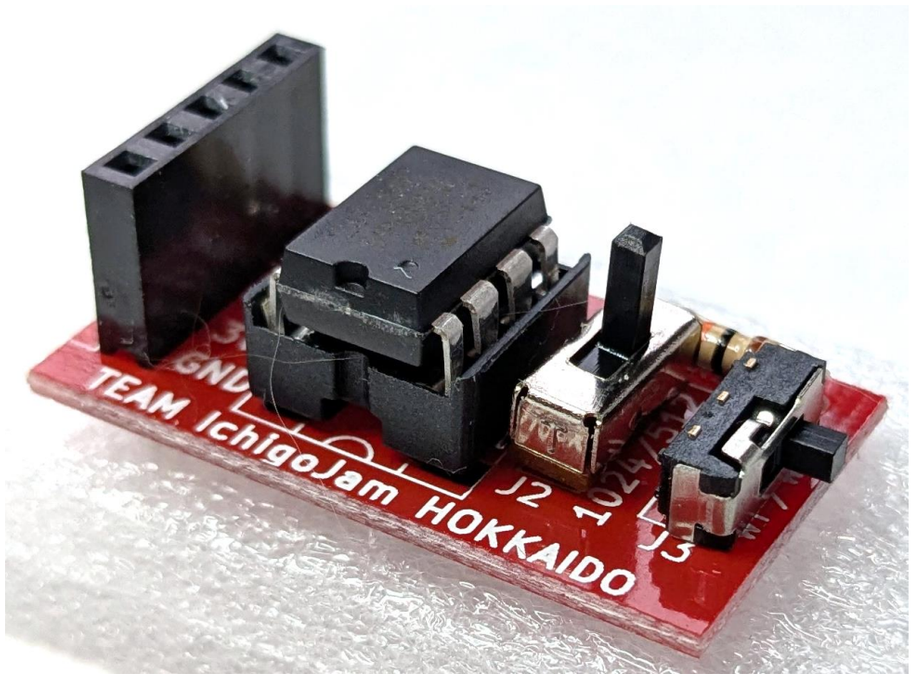
外部メモリーを組立よう！

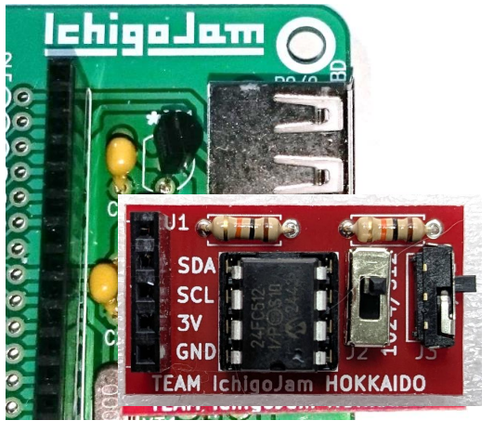
外部メモリキット

EEPROM_IJH-003

EEPROMはIchigoJam用の外部記憶装置モジュールです。EEPROMを接続することで、IchigoJam BASICのプログラムを本体の保存領域とは別に32個まで保存することができます。プログラムを書き込んだEEPROMを差し替えて使用したり、作成したプログラムをEEPROMを介して他の人と交換する、といった使い方も可能です。





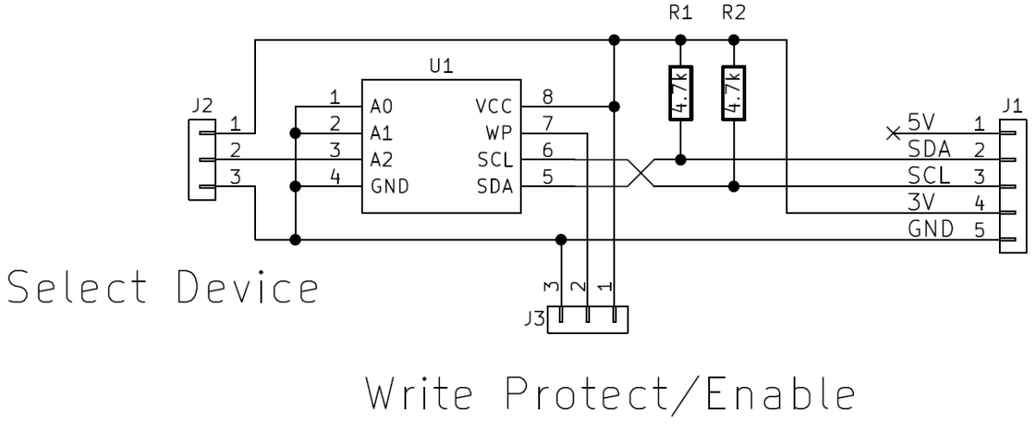


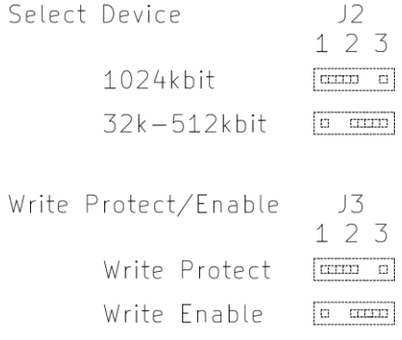
動作確認EEPROM （秋月電子通商）

- 24LC256　：付属品
- 24LC512-I/P
- 24FC1025-I/P

【参考】・外部記憶装置
EEPROM - イチゴジャム レシピ
https://goo.gl/x5B421
・福野泰介の一日一創
http://fukuno.jig.jp/1719

CN5端子に取り付けるだけで、拡張性もそのままで使えます。

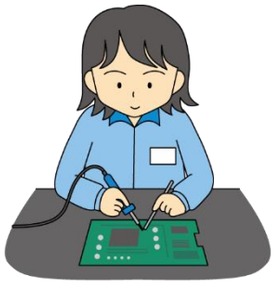




◆**J2　スライドスイッチ**
512　：24LC256、24LC512-I/P
1024　：24FC1025-I/P

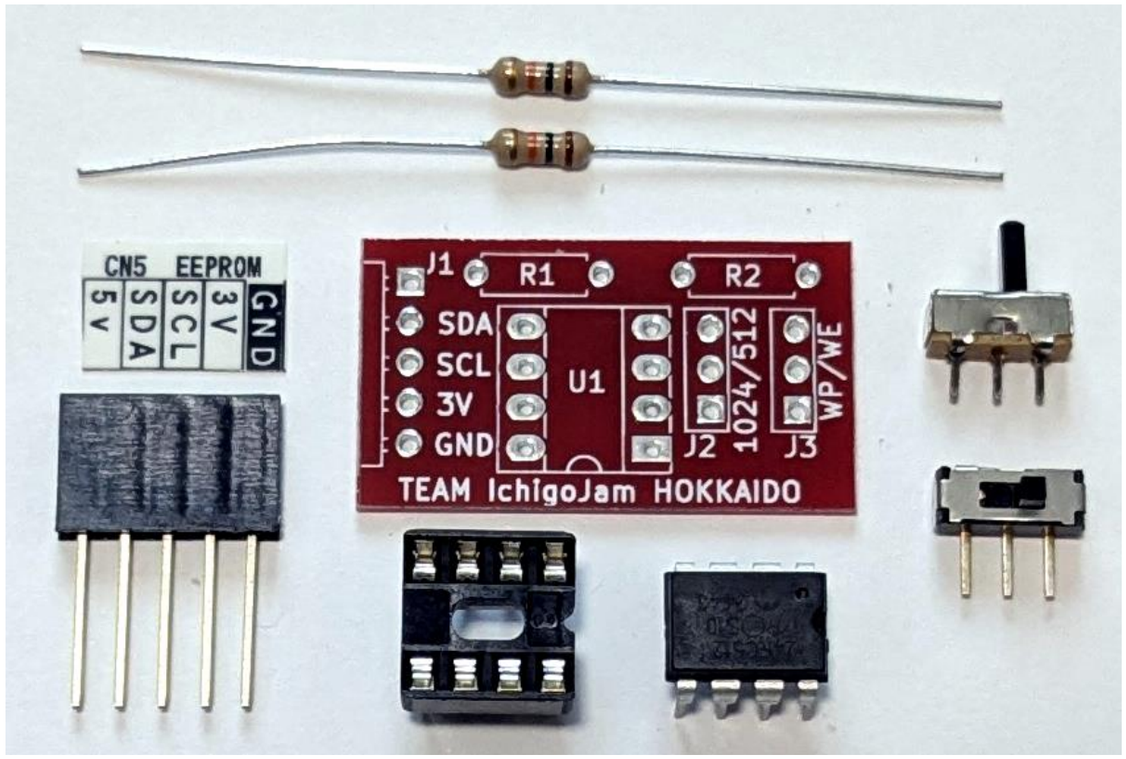
◆**J3　スライドスイッチ**
WE　：読み書きできる
WP　：書き込みを禁止

番号	名称	機能
1	NC	使用しません
2	SDA	I²C データ線
3	SCL	I²C クロック線
4	VCC	3.3V 出力(IchigoJam より)
5	GND	GND

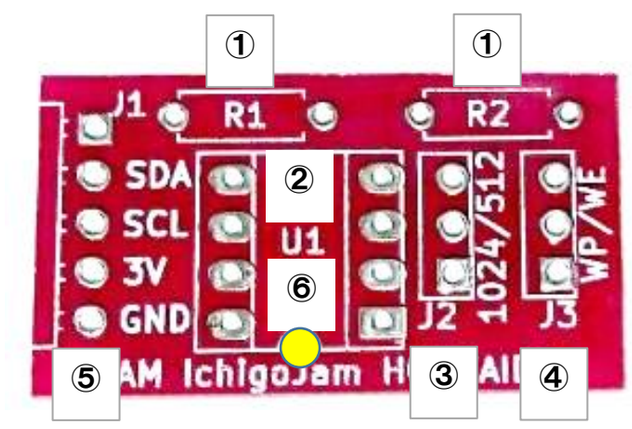


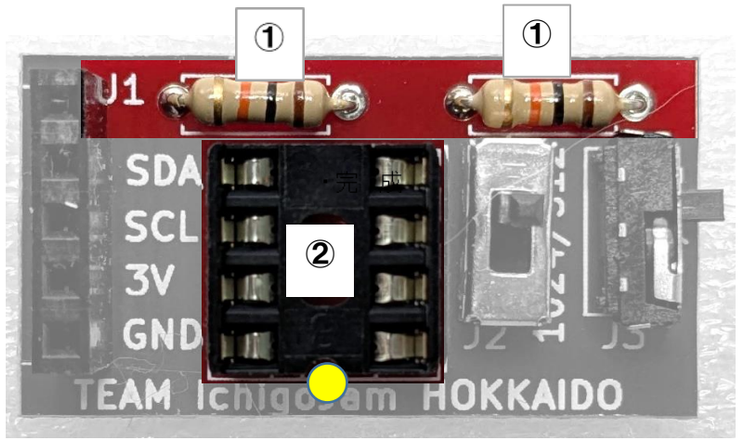
●組み立て手順

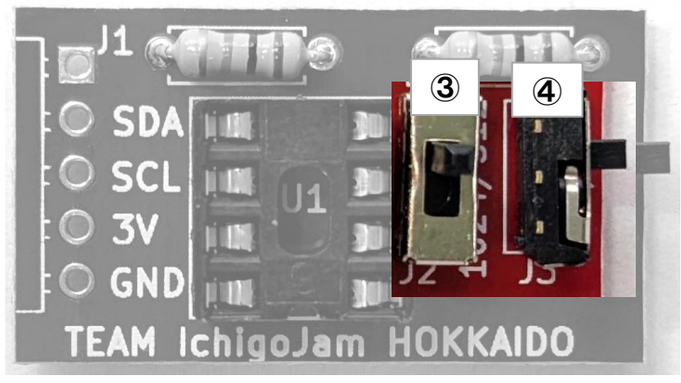
① R1　抵抗（10KΩ）
　R2　抵抗（10KΩ）
② U1　EEPROMソケット（8P）※　　向きに注意
③ J2 スライド縦スイッチ（3P）　向きに注意
④ J3 スライド横スイッチ（3P）　向きに注意
⑤ J1 ピンソケット（5P）
　※取り付ける前にシールを貼ります
⑥ EEPROMをソケットに取り付ける　※　　向きに注意

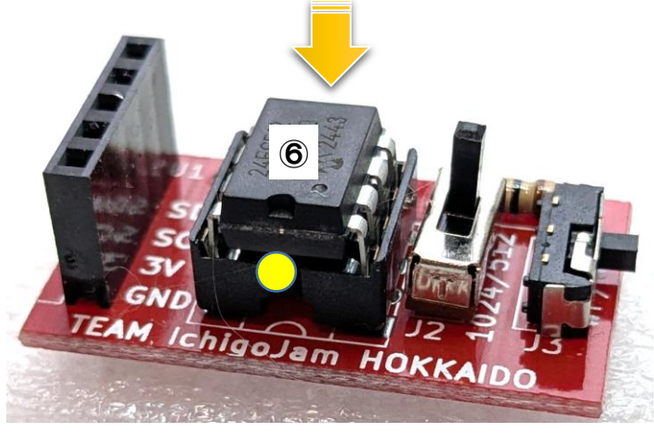


●組み立て手順

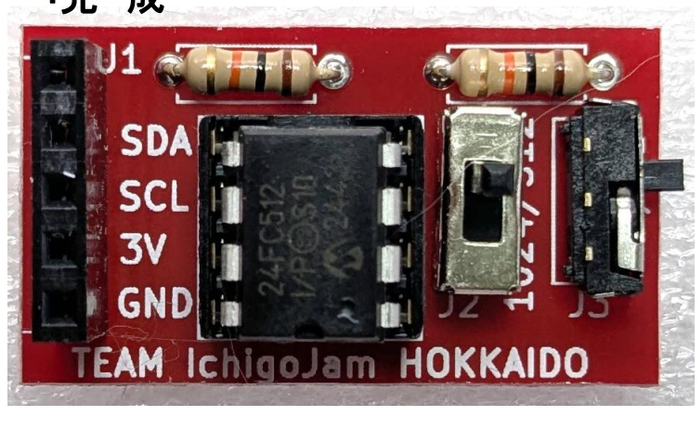








·完　成



WE

WP

※ROMソケットやEEPROMの向きに注意

外部メモリーを使ってみよう！

●使い方

・保存しているプログラムを**確認するとき**

FILES　110　　:110番までのプログラムが表示される

FILES　140　　:140番までのプログラムが表示される

・**読み込むとき**

LOAD　100　　:100番のプログラムが読み込まれる

LOAD　115　　:115番のプログラムが読み込まれる

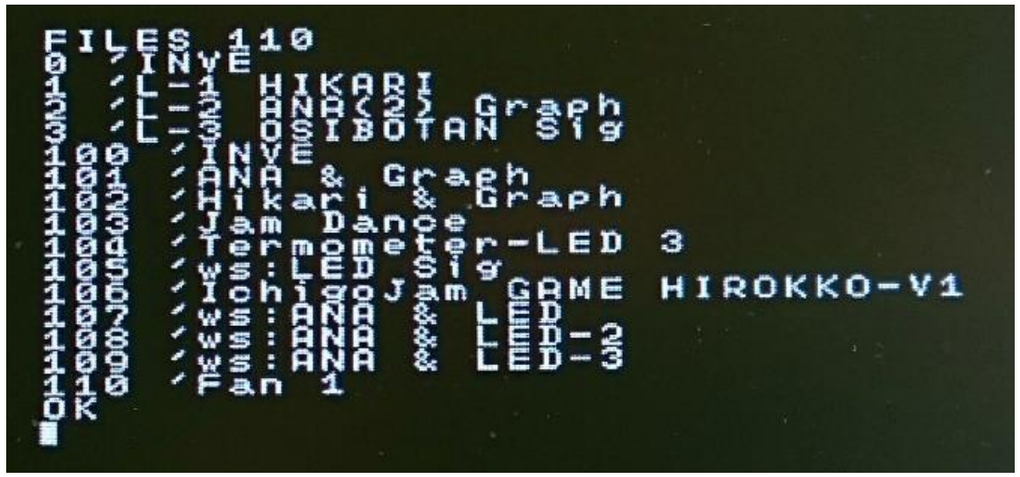
・**保存するとき**

SAVE　100　　: 100番にプログラムが保存される

SAVE　125　　: 125番にプログラムが保存される

※保存するプログラムの1行目に、プログラムのタイトルを記載すると区別がしやすくなります。

FILES 130　　:130番まで表示



・保存しているプログラムを**確認するとき**

FILES 110　　:110番まで表示

・プログラム番号は0番～3番がICHIGOJAM本体で100番以降が外部モリーの番号です。

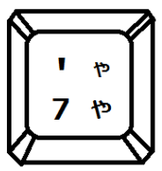
※保存したプログラムの1行が表示されます

プログラムの1行目にタイトルを記載すると分かりやすくなります。

書き方例

1 ‘LED Signal

‘（アポストロフィー文字)は
Shift(シフト)キーを押したまま数字の 7 を押す



外部メモリーのお約束です　！

1　外部メモリーを付けたり、取り外すときは必ずICHIGOJAMの電源を切ってから行いましょう。

2　プログラムを外部メモリーに書き込むときは、J3スライドスイッチをWE(書き込み可)にしてから行いましょう。

3　プログラムを外部メモリーに書き込まないときは、J3スライドスイッチをWP(書き込み不可)にしておきましょう。

4　プログラムを外部メモリーに書き込こんだら、ちゃんと書き込まれているかFILES命令を使って確認しましょう

```
FILES 120 ↵
```

・保存しているプログラムを120番まで表示する

今日のお約束です　！

1　プログラムを入力して、完成したら外部メモリーに保存して、周りのお友達とプログラムの交換をしましょう。

```
SAVE 101 ↵
```

プログラムを保存するときは、**保存する番号**に注意しましょう。

2　プログラムを入力しているときは、5行ぐらい入力したらプログラムを必ず退避しましょう。

```
SAVE0 ↵
```

正しく入力されているか、時々リストしてみましょう。

```
LIST ↵
```

3　正しく入力されているか、リストして確認できたら、実行してみましょう。

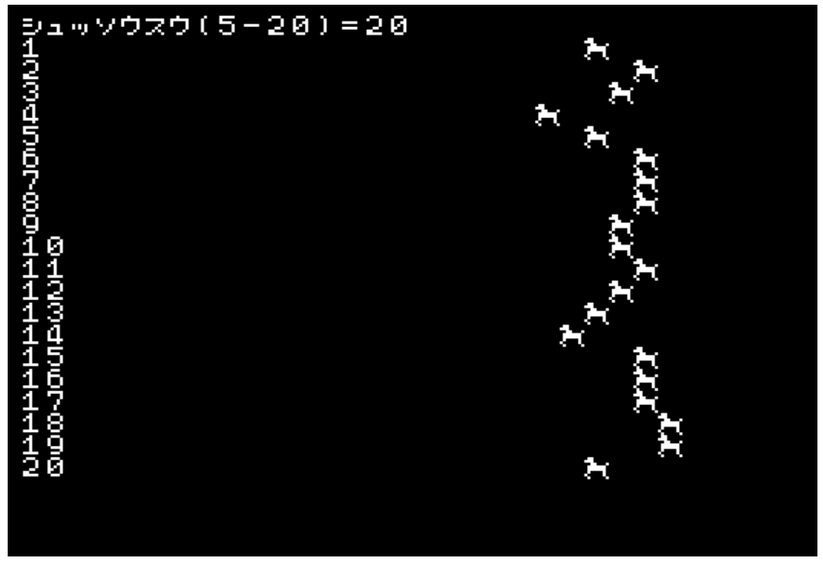
```
RUN ↵
```

4　ゲームができたら、無敵や高速にしたり改造してみましょう。

※　わからない事があったら先生に聞いてね

競馬ゲーム　を作ろう　！！

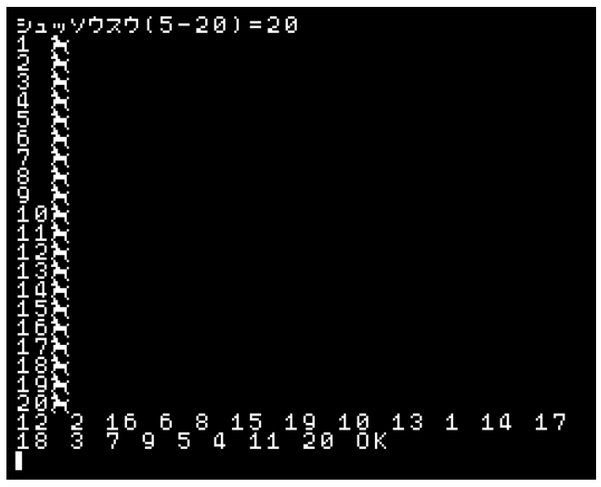
馬がスタートして、ランダムに走ります。全頭がゴールすると、ゴール順の番号が表示されて終了します



起動すると、出走数を聞かれます。
出走する馬の数を5～20頭の範囲で
入力してください。

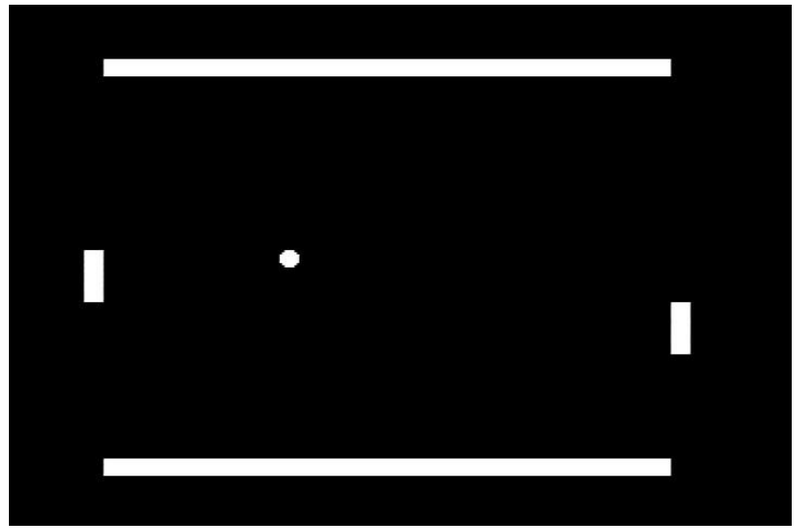
```
1‘　KEIBA GAME
10 CLV:CLS
20 POKE #700,#60,#E0,#31,#7E,#7E,#42,#82,#41
30 @M
40 INPUT "ｼｭｯｿｳｽｳ(5-20)=",M
50 IF M<5 OR M>20 CLS:GOTO @M
60 FOR H=1 TO M
70 LC 0,H:?H
80 [H]=29:LC 29,H:?CHR$(224)
90 NEXT:H=1
100 @LOOP
110 IF RND(2) && [H]>2 [H]=[H]-1:LC [H],H:
?CHR$(224,0):IF [H]=2 N=N+1:[20+N]=H
120 H=H+1:IF H>M H=1
125 WAIT 0
130 IF N<M GOTO @LOOP
```

```
140 LC 0,21
150 FOR H=1 TO M
160 ?[20+H];" ";
170 NEXT
```



テニス　ゲームを作ろう　！！

昔のビデオゲーム「Pong」風に2人で対戦するテニスゲームです。実行すると左右のパドル、上下の壁、ボールが表示されます。動くボールをパドルで打ち返してください。打ち返せずにミスするとゲームオーバーです。



```
1 ‘TENISU GAME
10 CLV:CLS:P=10:Q=10:X=16:Y=1:D=1:E=1
20 LC 1,0:FOR Z=1 TO 29:?CHR$(143);:NEXT
30 LC 1,23:FOR Z=1 TO 29:?CHR$(143);:NEXT
40 P=P-BTN(88)*(P>0)+BTN(32)*(P<19)
50 LC 0,P:?CHR$(0,28,31,1,28,31,1,28,31,1,28,31,0);
60 Q=Q-BTN(30)*(Q>0)+BTN(31)*(Q<19)
70 LC 30,Q:?CHR$(0,28,31,1,28,31,1,28,31,1,28,31,0);
80 U=X+D:V=Y+E:C=SCR(U,V)
90 IF C=1 D=-D:BEEP 30,2:GOTO 80
100 IF C=143 E=-E:BEEP 20,2:GOTO 80
110 LC X,Y:?CHR$(0);
120 LC U,V:?CHR$(233);
130 X=U:Y=V
140 WAIT 3
150 IF X>0 AND X<30 GOTO 40
160 BEEP 30,30:LC 0:CLK
```

パドルのキー操作は以下のとおり
左プレイヤー：Xで上移動、スペースで下移動します
右プレイヤー：↑で上移動、↓で下移動します

ＰＫ戦　3ゴール　ゲームを作ろう　！！

□key　F5（又はrunエンター）でプレイ　←↑→でキック

```
1  ‘PK GAME
10 CLV:VIDEO 3
20 CLS:?"[=====";CHR$(250);"=====]"
30 X=6:LC X,6:?CHR$(233)
40 LC 0,9:?"GOAL=";G
50 IF BTN(LEFT)=1 THEN D=-1:GOTO 70
51 IF BTN(UP)=1 THEN D=0:GOTO 70
52 IF BTN(RIGHT)=1 THEN D=1:GOTO 70
60 GOTO 50
70 FOR Y=6 TO 2 STEP -1
80 LC X,Y:?" ":X=X+D
90 LC X,Y-1:?CHR$(233):WAIT 5
100 NEXT
110 LC 6,0:?"="
120 R=RND(3)-1
130 LC 6+R*5,0:?CHR$(250+R):WAIT 30
140 IF D<>R G=G+1:PLAY"T620O5CRCRCRC.RO4A-.B-.O5CO4B-
O5C.":WAIT 180:GOTO 20
150 BEEP 30,30:WAIT 60:LC 4,5:?"GAME OVER"
```

Web版Ichigojamを使ってみよう！！

Web ブラウザ上で IchigoJam BASIC が動作します。インターネットに接続できる環境であれば、お使いのブラウザですぐに使用できます。
OS を問わずに使用できます。

KEY キーボードを表示します

ESC エスケープを入力します

EXPORT テキストデータを出力します

IMPORT テキストデータを入力します

FULL 全画面表示します

I/O 入出力の設定をおこないます

AUDIO ON 音のオン・オフを設定します

Ichigojamパソコンを持っていなくてもProgramを作ったり実行する事ができるよ

インターネットに接続しているパソコンのブラウザから下のアドレスをアクセスしてね

```
https://fukuno.jig.jp/app/IchigoJam/
```

コマンドをおぼえよう　！！

プログラムをリストするには

LIST ↵

範囲を指定してリスト
LIST　10,100

プログラムをすべて削除するには

NEW ↵

プログラムを実行するには

RUN ↵

画面表示をすべて消します（クリア スクリーン）

CLS ↵

プログラムをメモリーに記憶させるには

SAVE ↵

(0～3)の4個のメモリーがあります

プログラムの行削除は行番号だけを打ちます

10 ↵

プログラムをメモリーから読みだすには

LOAD ↵

メモリーの内容は電源を切っても記憶しています

実行しているプログラムを停止するには

Esc キーを押します

エスケープと読みます
(Escape：逃れる、抜け出す、脱出する)の省略型です

メモリーの内容をみるには

FILES ↵

コマンドはキー入力だけでなくファンクションキーでも代用できるよ

F1	F2	F3	F4	F5	F6	F7	F8	F9
CLS	LOAD	SAVE	LIST	RUN	?FREE()	OUT0	VIDEO1	FILES

「IchigoJam BASIC 1.4 コマンド一覧」　https://fukuno.jig.jp/app/csv/ichigojam-cmd.html

IchigoJam BASIC 1.1 cheatsheet

＊赤文字は省略可能を示す　＊青文字は説明文

ファンクションキー

F1	F2	F3	F4	F5	F6	F7	F8	F9
CLS	LOAD	SAVE	LIST	RUN	?FREE()	OUT0	VIDEO1	FILES

Dec	Hex	Bin
0	#00	0000
1	#01	0001
2	#02	0010
3	#03	0011
4	#04	0100
5	#05	0101
6	#06	0110
7	#07	0111
8	#08	1000
9	#09	1001
10	#0A	1010
11	#0B	1011
12	#0C	1100
13	#0D	1101
14	#0E	1110
15	#0F	1111

4bit：0～15
8bit：0～255
　　(-128～127)
16bit：0～65535
　　(-32768～32767)

＊扱える数値は 16 ビット範囲
小数値は使えません。

VIDEO1	KBD1
VIDEO2	EX1
IN1	KBD2
IN2	SOUND
IN3	ISP
IN4	RESET
VCC	GND
GND	VCC
OUT1	-
OUT2	-
OUT3	OUT5
OUT4	OUT6
BTN	TXD
LED	RXD

演算の優先順位

高い ｜ ( ) ｜ - ~ ! NOT ｜ * / % MOD << >> & ^ ｜ + - | ｜ = != < > <= >= ｜ AND ｜ OR ｜ 低い

●式／演算

【　加　　算　】数 + 数
【　減　　算　】数 - 数
【　乗　　算　】数 * 数
【　除　　算　】数 / 数
【　剰　　余　】数 % 数
【　否　　定　】NOT 式
【　論理積　　】数 & 数
【　論理和　　】数 | 数
【排他的論理和】数 ^ 数
【　右シフト　】数 >> 数
【　左シフト　】数 << 数
【　ビット反転】~ 数
【優先順位変更】(~)
式 AND 式　省略形：&&
式 OR 式　　省略形：||

●代入／変数／配列変数

[数]　0～101 まで
LET 変数 , 数　省略形：変数 = 数
LET [数], 数 ,.. 数 n

●リセット／初期化

CLK　キーバッファ消去
CLP　パターン初期化
CLS　画面消去
CLT　時間をリセット
CLV　変数を消去し全て 0 に
SRND 数　乱数の種を設定

●キー入力／ボタン

BTN(0/UP/DOWN/RIGHT/LEFT/SPACE)
INKEY()　リアルタイムキー入力
INPUT 文字列 , 変数

●画面関係

LOCATE X 座標 , Y 座標 省略形：LC
PRINT 数や文字列　省略形：?
SCR(X 座標 , Y 座標)
SCROLL 数　0: 上、1: 右、2: 下、3: 左
VIDEO 数 1, 数 2

●関数

ABS(数)
ASC(" 文字 ")
BIN$(数 , 桁数)
CHR$(数 ,.. 数 n)
DEC$(数 , 桁数)
HEX$(数 , 桁数)
RND(数)

●数値表記

123　10 進数
　　(-32768 ～ 32767)
#E9　16 進数 (0 ～ #FFFF)
`1001　2 進数

●定数

LEFT　左　: 28
RIGHT　右　: 29
UP　上　: 30
DOWN　下　: 31
SPACE　空白: 32

●条件判断／条件式

IF 数 THEN 次 ELSE 次 2
【　等　し　い　】数 1= 数 2
【　等しくない　】数 1<> 数 2
【　小　さ　い　】数 1< 数 2
【小さいか等しい】数 1<= 数 2
【　大　き　い　】数 1> 数 2
【大きいか等しい】数 1>= 数 2

●移動／繰り返し／サブルーチン

FOR 変数 = 数 1 TO 数 2 STEP 数 3 ～ NEXT
GOSUB 行番号　省略形：GSB
GOTO 行番号
RETURN　省略形：RTN

●ハードウェア

ANA(数)　0～1023
BPS 通信速度　省略時：115,200bps
I2CR(数 1, 数 2, 数 3, 数 4, 数 5)
I2CW(数 1, 数 2, 数 3, 数 4, 数 5)
IN(数)　IN1-9 から入力
LED 数　0: 消灯、1: 点灯
OUT 数 1, 数 2　OUT1-7 に出力
PWM 数 1, 数 2, 数 3
RESET　リセット
SLEEP　スリープ（ボタンを押すと復帰）
UART シリアル出力設定 , シリアル受信設定
WAIT 数　60 で約 1 秒

●ファイル

FILE()
FILES 数 1, 数 2
LOAD 数
LRUN 数 , 行番号
RUN
SAVE 数

●プログラム

CONT　再度実行する
END　プログラムを終了
FREE()　プログラムの残りメモリ数
LINE()　現在実行中の行番号
LIST 行番号 1, 行番号 2
NEW　プログラムを消す
RENUM 数 1, 数 2
STOP　処理を中断する

●メモリ操作／マシン語

PEEK(アドレス)
POKE アドレス , 数 ,... 数 n
USR(アドレス , 数)

●その他

HELP　メモリマップを表示
REM 注釈　省略形：'
TICK()　tick 時間 (1/60) を返す
VER()　バージョン番号を返す

メモリマップ

#0000	文字パターン (#00～ #DF)
#0700	PCG パターン (#E0～ #FF)
#0800	配列変数・変数
#0900	画面 (32 文字 ×24 行)
#0C00	プログラムリスト
#1001	
#1002	キーが押されたビット
#1003	キーバッファ格納数 (最大 14)
#1004～F	キーバッファ

●音楽／サウンド

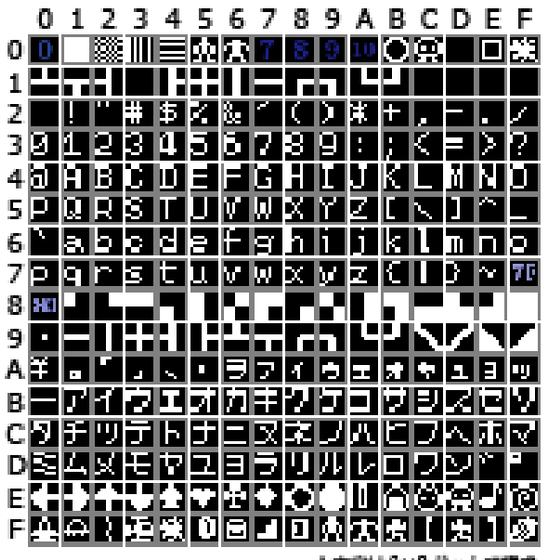
BEEP 周期 , 長さ　BEEP を鳴らす
PLAY【MML】　MML なしで演奏停止
SOUND()　再生中なら 1 を返す
TEMPO テンポ　テンポを指定

■MML (Music Macro Language)

[音]　音 (C D E F G A B R)
[音]n　音長 (. を付けると 1.5 倍長)
[音]+　半音上げる
[音]-　半音下げる
Tn　テンポ (初期値 :120)
Ln　デフォルトの音長 (初期値 :4)
On　オクターブ指定 (1～5)
>　1 オクターブ上げる
<　1 オクターブ下げる
$　以後の MML を繰り返す
Nn　音の高さを指定
　(音長で指定可能な値 : 1,2,3,4,8,16,32)

●制御（コントロール）コード

08(#08)　バックスペース（後退）
13(#0D)　リターン
14(#0E)　インサート（挿入）
127(#7F)　デリート（削除）



1 文字は 8×8 ドットで構成

「IchigoJam 1.2 チートシート　by OpenSpace」

http://www.openspc2.org/reibun/IchigoJam/etc/cheat-sheet/0002/

いま をつくる！

make think create

n:bit TM
－ micro:bit拡張ボード －

NPO法人　**NEXTDAY**　は

子供たちの学びを支援しています

お問い合わせ先　nextday@ict.skr.jp

未来 を創る！

搭載するセンサー基板の開発と発射実験を続けています。
2023年秋までに打ち上げ・体験教室の開催を目指します。

子どもたちに 創る Creative + I nformation T echnology & C ommunicatior ollaboration 楽しさを

https://nextday-kids.com/