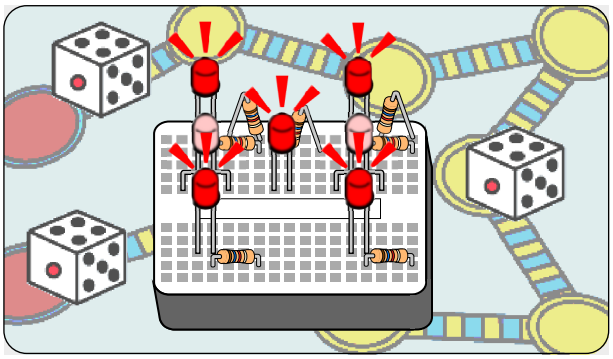
IchigoJam 電子工作パーツセット
電子さいころ
DIC-jamP

Electronic Dice Parts Set

IchigoJam 0.9.7/1.0.1対応

0 概要



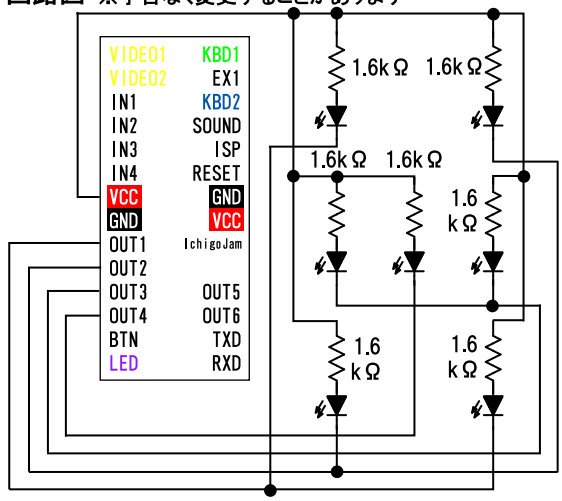
7個のLEDをさいころの目に見立てた、電子さいころの部品セットです。IchigoJam本体のタクトスイッチを押すとさいころを振ります。

本キットのプログラムはIchigoJam バージョン0.9.7及びバージョン1.0.1で動作確認しています。

部品表 ※予告なく変更することがあります

	品名/型番/値	数量	備考
1	LED 赤	7	
2	抵抗 1.6kΩ	7	茶青赤金の色帯
3	ジャンパワイヤ 10cm	12	
4	ブレッドボード	1	

回路図 ※予告なく変更することがあります



【企画・販売元】 オーディオ・マイコン・メカトロ・電子パーツ

デジット

年中無休・営業時間：AM10:00～PM8:00
〒556-0005 大阪市浪速区日本橋4-6-7

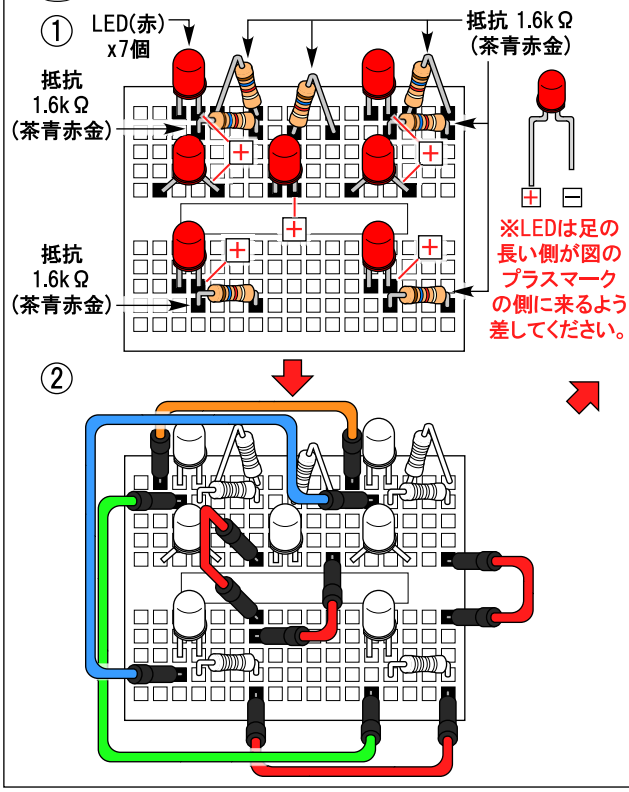
[TEL]06-6644-4555 / [FAX]06-6644-1744
[HP]http://digit.kyohritsu.com
[Blog]http://blog.digit-parts.com [Twitter]@0666444555

【販売窓口】

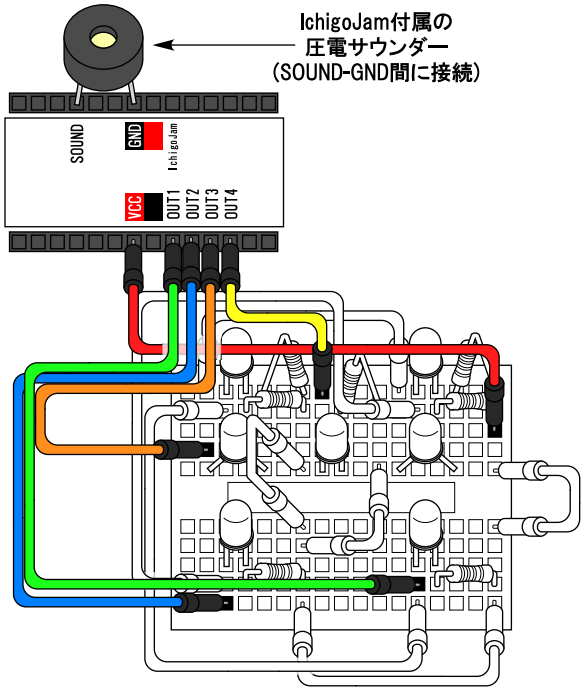
■ シリコンハウス	(大阪・日本橋店舗)	06-6644-4446
■ デジット	(大阪・日本橋店舗)	06-6644-4555
■ 法人営業部	(B2B/学校・官公庁)	06-6646-0707
■ 通販営業部	(インターネット通販)	06-6644-6116

創業1970年 KYOHRITSU 共立電子産業株式会社 共立電子 検索

1 組み立てかた



③ IchigoJamとの接続



※ジャンパワイヤの色は配線の区別のため色分けしてあります
（実際には何色の線で配線してもかまいません）

はじめのプログラム

IchigoJam本体のタクトスイッチを押すとさいころを振るプログラムです。

(1) タクトスイッチを押すと、1から6までの乱数の値を求めます。

(2) 求めた乱数の値に対応したLEDを点灯させます。

(3) タクトスイッチを押している間、(1)(2)の処理をくりかえします。

プログラムリスト

```
1 REM ﾃﾞﾝｼ ｻｲｺﾛ
2 REM 2015/07/14
10 PRINT "ｻｲｺﾛ ﾌﾟﾛｸﾞﾗﾑ"
20 REM ｽｲｯﾁ ｦ ｵｽﾏﾃﾞ ﾏﾂ
30 IF BTN()=1 THEN GOTO 50
40 GOTO 30
50 N=RND(6)+1
60 PRINT "N=";N
70 OUT 15
80 IF N<>1 THEN OUT 1,0
90 IF (N%2)=1 THEN OUT 4,0
100 IF N>=4 THEN OUT 2,0
110 IF N=6 THEN OUT 3,0
120 IF BTN()=1 THEN GOTO 50
130 GOTO 30
```

```
      ⋮
80 GOTO 30

SAVE0↵
```

プログラムを書いたあと「SAVE0↵」でプログラムを保存してください。
(↵は[ENTER]キーです)
※0番に大事なプログラムが入っている場合は、他の番号を指定して保存してください。

キーワード

- REM文(コメント)
- IF文と条件判断
- RND()関数
- 割り算の余り
- PRINT文の使い方
- BTN()関数
- 変数 N

SAVEコマンドの使い方

```
      ⋮
130 GOTO 30

SAVE0↵
```

SAVEコマンドは、作成したプログラムを保存するコマンドです。

プログラムをSAVEしないと、IchigoJamの電源をOFFにしたときに作成したプログラムが消えてしまいます。

SAVEコマンドの後ろにプログラム番号(IchigoJam本体に保存するときは0～3、EEPROMカセットに保存するときは100～131)を指定して保存します。

! SAVEすると指定した番号に入っていたプログラムは上書きされます(消えてなくなります)ので十分注意してください。

```
LOAD0↵
Loaded xxxbyte
OK
```

SAVEコマンドで保存したプログラムは、LOADコマンドでファイル番号を指定すると呼び出すことができます。

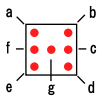
プログラムの説明

(1) プログラムの動作

IchigoJamのタクトスイッチを押している間さいころを振ります。
1から6の乱数(ランダムな数)を求めてモニタ画面に表示し、ブレッドボードのLEDでさいころの目を表示させます。

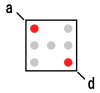
乱数を求めるにはRND(6)関数を使っています。

(2) さいころの出目の表示のしかた



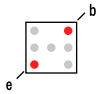
このプログラムでは、さいころの出目を表示するのにLEDを7個使用しています。説明の都合上、左図のように記号をつけます。

50行で求めたNの値とLEDの点灯状態の関係は次の図の通りです。

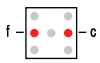


条件(1) Nが1でないとき
aとdのLEDを点灯させます

条件(2) Nが奇数のとき
(Nを2で割った余りが1のとき)
gのLEDを点灯させます



条件(3) Nが4以上のとき(80行)
bとeのLEDを点灯させます



条件(4) Nが6のとき(90行)
cとfのLEDを点灯させます

Nの値が条件(1)から条件(4)のそれぞれを満たすか満たさないかの組み合わせでさいころの出目を表現します。

たとえば、Nが5のときは次のようになります。

条件(1) 満たす　条件(2) 満たす　条件(3) 満たす　条件(4) 満たさない

キーワードの説明

REM文(コメント) (1行)

REM文は、プログラムにコメント(メモ)を入れるために使います。

(例) REM ｽｲｯﾁ ｦ ｵｽﾏﾃﾞ ﾏﾂ

「REM」の後ろはプログラムの実行時は無視されます。

PRINT文の使い方(10行)

PRINT文は変数の値(計算結果)やメッセージを表示するのに使用します。
基本的な使用方法は次のとおりです。

(例) PRINT K
この例では変数Kの値を画面に表示します。

(例) PRINT "ﾎﾝｼﾞﾂﾊ ｾｲﾃﾝﾅﾘ"
この例では「ﾎﾝｼﾞﾂﾊ ｾｲﾃﾝﾅﾘ」というメッセージの文字列を画面に表示します。文字列はダブルクオーテーション「"」で囲みます。

PRINT文で表示させる項目をセミコロン「;」記号で区切って続けて書くと、複数の項目を1行に表示させることができます。

(例) PRINT W/10;".";W%10;"V"
この例ではまず変数Wの値を10で割った値(小数点以下は切捨てとなります)を表示し、そのあとに小数点記号(ピリオド「.」記号)を表示し、そのあとに変数Wの値を10で割った余りを表示、最後に文字「V」を表示します。

PRINT文の後ろに何も指定しない場合は、何も表示せず、改行のみ行います。

IF文と条件判断 (30行)

IF文は、条件によって処理内容を変えたいときに使います。

(例) IF BTN()=1 THEN GOTO 50

条件式　条件式が真のとき実行する文

この例では、BTN()関数が返す値が1と等しいかどうか比較し、もし真ならば(等しければ)「THEN」以下の文を実行します。(偽ならば(等しくなければ)次の行に移ります)

条件式が偽のときに実行する文を書く必要があるときは、条件式が真のときに実行する文に続けて「ELSE」と書き、その後ろに条件式が偽のときに実行する文を書きます。

(例) IF A<B THEN MAX=B ELSE MAX=A

条件式　条件式が真のとき実行する文　条件式が偽のとき実行する文

この例ではAがBより小さいかどうか比較し、もし真ならば(A<Bならば)MAXにBを、偽ならば(A>=Bならば)MAXにAを代入します。

BTN()関数 (40行)

BTN()関数は、IchigoJam本体のタクトスイッチが押されていれば「1」、押されていなければ「0」を返す関数です。

(例) IF BTN()=1 THEN GOTO 50

この例ではタクトスイッチを押した場合に50行にジャンプします。

IchigoJam バージョン1.0.1では、IchigoJam本体のタクトスイッチの状態以外にキーボードの状態も取り込めます。状態を調べたいキーは、BTN()関数のかっこの中に指定します。

BTN(SPACE)：スペースキーの状態
BTN(UP)：上矢印キーの状態
BTN(DOWN)：下矢印キーの状態
BTN(LEFT)：左矢印キーの状態
BTN(RIGHT)：右矢印キーの状態

かっこの中身を省略した場合はIchigoJam本体のタクトスイッチの状態を取り込みます。(バージョン0.9.7と同じ動作をします)

RND()関数 (50行)

RND()関数は0以上かっこ「()」内の数値未満のランダムな数値を返す関数です。※RND()関数のかっこ「()」内の数値は含みません。

(例) N=RND(6)+1

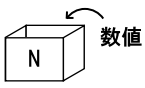
この例では、0から5までのランダムな数値を求め、それに1を足したものを変数Nに代入しています。

変数 N (50行)

変数とは、プログラム中で使用する数値を入れるための「箱」です。箱を区別するために、箱には「名前」をつけます。(これを「変数名」といいます)

IchigoJamのBASICでは、変数名は「A」から「Z」までのアルファベット1文字ですので、全部で26個の変数が使用できます。

コピー(代入)



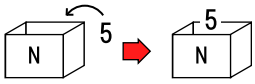
箱に式の計算結果や別の箱の中身を入れる(コピーする)ことを「代入」といいます。
(※コピー元の箱の中身は変化しません)

(参考：BASICでは変数を使用する前に「この名前の変数を使用します」と宣言する必要はありません)

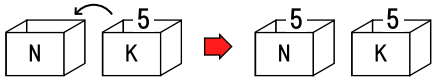
(参考2：IchigoJamのBASICで扱える数値の範囲は、-32768から32767までの整数です)

変数に数値を代入するには、次のように書きます。

(例) N = 5



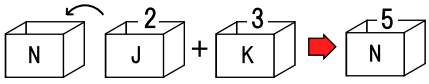
(例) N = K



※コピー元の変数(K)の値は変化しません

足し算や引き算などの計算にも使用します。

(例) N = J + K



割り算の余り (90行)

IchigoJamのBASICでは、足し算、引き算、掛け算、割り算の四則演算に加えて、割り算の余りを求めることができます。

(例) IF (N%2)=1 THEN OUT 4,0

割り算の余りを求める演算子は「%」です。この例では、Nを2で割ったときの余りを求めて、その余りが1と等しいかどうか比較しています。(奇数か偶数かを調べています)

入門プログラム

タクトスイッチを放してからも少しのあいださいころが転がっているようにはじめのプログラムを改良します。
さいころが転がる都度圧電サウンダーでビープ音を鳴らします。

タクトスイッチを放すと、プログラム中で指定した回数だけ乱数の値を求め、LEDを点灯させる処理を追加します。

プログラムリスト

```
1 REM ﾃﾞﾝｼ ｻｲｺﾛ
2 REM 2015/07/14
10 PRINT "ｻｲｺﾛ ﾌﾟﾛｸﾞﾗﾑ"
20 REM ｽｲｯﾁ ｦ ｵｽﾏﾃﾞ ﾏﾂ
30 IF BTN()=1 THEN GOTO 50
40 GOTO 30
50 BEEP 5,1
60 GOSUB 150
70 IF BTN()=1 THEN GOTO 50
80 M=64
90 BEEP 5,1
100 GOSUB 150
110 M=M-1
120 IF M>0 THEN GOTO 90
130 GOTO 30
140 REM ｻｲｺﾛ ﾉ ﾒ ｦ ﾋｮｳｼﾞ
150 OUT 15
155 N=RND(6)+1:PRINT "N=";N
160 IF N<>1 THEN OUT 1,0
170 IF (N%2)=1 THEN OUT 4,0
180 IF N>=4 THEN OUT 2,0
190 IF N=6 THEN OUT 3,0
200 RETURN
```

キーワード

- BEEP文
- ループ構造
- サブルーチン(GOSUBとRETURN)

キーワードの説明

BEEP文 (50行)

BEEP文は、IchigoJam本体に接続した圧電サウンダーを鳴らす命令です。

(例) BEEP 5,1

この例では、圧電サウンダーを音の高さ5、長さ1で鳴らします。BEEP文で指定する最初の数値は音の高さ(1～255、大きくなるほど音が低くなる)、2番目の数値は鳴らす長さ(1/60秒単位)です。

(例) BEEP

BEEP文の後ろの数値を省略することも可能です。(デフォルトの音の高さと長さで鳴ります)

ループ構造 (80～120行)

プログラム中で同じ処理を繰り返し行わせることがよくあります。
同じ処理を繰り返し行わせるには、次のようにループ構造を書きます。

(例)

```
80 M=64 ←(1)残り回数セット
90 BEEP 5,1
100 GOSUB 150 ←(2)ループ本体
110 M=M-1 ←(3)残り回数を減らす
120 IF M>0 THEN GOTO 90 ←(4) 残り回数が
                            0より大きければ
                            ループ本体を実行
```

(1) ループの残り回数を保持するための変数(M)にループを回る回数(残り回数)をセットします。

(2) ループ本体の処理(繰り返し行わせる処理)を書きます。

(3) 残り回数を保持している変数(M)の値を1減らします。

(4) 残り回数を保持している変数(M)の値が0より大きいならば、ループ本体の先頭にジャンプします。(Mの値が0の場合はジャンプせず、繰り返しループから抜けます)

IchigoJam バージョン1.0.1では、上に説明した方法以外に、FOR/NEXT文を使った方法でもループ構造を作れます。

FOR/NEXT文がないIchigoJam バージョン0.9.7でも同じプログラムで実験できるように、本キットではFOR/NEXT文は使用していません。

サブルーチン(GOSUBとRETURN) (150～200行)

プログラム中の複数の箇所で同じ処理を行うことがよくあります。

このような場合は、その処理をサブルーチンプログラムとしてまとめ、サブルーチンプログラムを呼び出すようにするとプログラムがすっきりします。

サブルーチンプログラムを呼び出すには、「GOTO」文の代わりに「GOSUB」文を使います。(※GOTO文で呼び出さないでください)

(例) GOSUB 150

この例では、150行以降にあるサブルーチンプログラムを呼び出します。

サブルーチンプログラムの例は次のとおりです。

(例)

```
150 OUT 15
160 N=RND(6)+1:PRINT "N=";N
170 IF N<>1 THEN OUT 1,0
180 IF (N%2)=1 THEN OUT 4,0
190 IF N>=4 THEN OUT 2,0
200 IF N=6 THEN OUT 3,0
210 RETURN ←RETURN文でサブルーチンプログラムから復帰します
```

この例では、150行からRETURN文のある210行までがサブルーチンプログラムになります。

GOSUBで呼び出したサブルーチンプログラムの最後には、RETURN文を置きます。GOSUB文とRETURN文は常にペアで使用してください。

RETURN文を実行すると、サブルーチンプログラムから復帰します。

4 上級プログラム

入門プログラムを改良して、スイッチを放したあとさいころが転がるスピードが少しずつ遅くなるようにします。
WAIT文を使うことで待ち時間を作っています。

プログラムリスト

```
1 REM ﾃﾞﾝｼ ｻｲｺﾛ
2 REM 2015/07/14
10 PRINT "ｻｲｺﾛ ﾌﾟﾛｸﾞﾗﾑ"
20 REM ｽｲｯﾁ ヲ ｵｽﾏﾃﾞ ﾏﾂ
30 IF BTN()=1 THEN GOTO 50
40 GOTO 30
50 BEEP 5,1
60 GOSUB 150
70 IF BTN()=1 THEN GOTO 50
80 M=28
90 BEEP 5,1
100 GOSUB 150
105 WAIT 29-M
110 M=M-1
120 IF M>0 THEN GOTO 90
125 BEEP 5,60
130 GOTO 30
140 REM ｻｲｺﾛ ﾉ ﾒ ヲ ﾋｮｳｼﾞ
150 OUT 15
155 N=RND(6)+1:PRINT "N=";N
160 IF N<>1 THEN OUT 1,0
170 IF (N%2)=1 THEN OUT 4,0
180 IF N>=4 THEN OUT 2,0
190 IF N=6 THEN OUT 3,0
200 RETURN
```

キーワード

- WAIT文

プログラムの説明

タクトスイッチを放したあとさいころが転がるスピードが少しずつ遅くなるように、タクトスイッチを放したあとのループの部分(90行～120行)にWAIT文を入れています。

このプログラムでは少しずつ遅くなるようにしたいので、ループの回数(変数を含んだ式)をWAIT文の指定時間として指定しています。

キーワードの説明

WAIT文 (105行)

WAIT文は、プログラムの実行を指定した時間だけ一時停止させる文です。一時停止させる時間は1/60秒(約16.7ミリ秒)単位で指定します。

(例) WAIT 60

この例では、プログラムの実行を1秒(1/60秒 x 60 = 1秒)だけ一時停止させます。

一時停止させる時間として、定数の代わりに変数を含んだ式を指定することもできます。

(例) WAIT 29-M

この例のように変数を含んだ式を指定すると、待ち時間を可変にすることも可能です。