

# IchigoJam 電子工作パーツセット

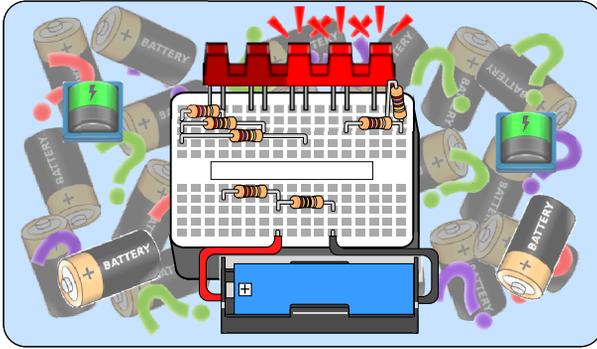
DBC-jamP

## 乾電池チェッカ

### Battery Checker Parts Set

0 概要

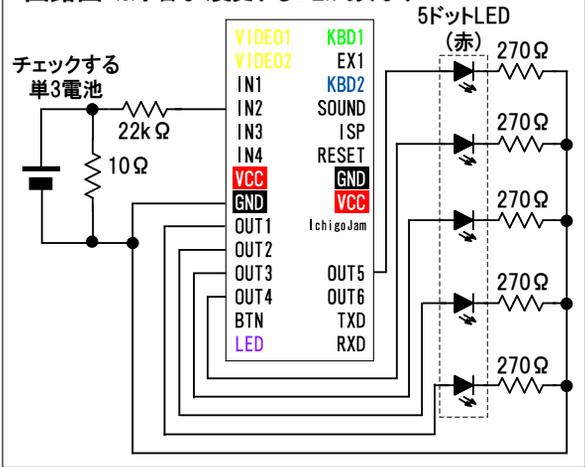
IchigoJam 0.9.7 / 1.0.1対応



電子工作をしていると、使い古しの乾電池がたまります。まだ新しい電池と寿命が来た電池が混ざってしまうと、簡単には見分けられません。この乾電池チェッカは、電池に所定の負荷抵抗(10Ω)を接続した状態の電圧を読み取り、電池の能力を5ドットのLEDで棒グラフ状に表示します。

本キットのプログラムはIchigoJam バージョン1.0.1用です。プログラムの一部変更でバージョン0.9.7でも実行できます。

### 回路図 ※予告なく変更することがあります



### 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	品名/型番/値	数量	備考
5ドットLED 赤	1		ジャンプワイヤー 10cm	9	
抵抗 10Ω	1	茶黒黒金の色帯	電池ホルダ 単3x1用	1	
抵抗 270Ω	5	赤紫茶金の色帯	ブレッドボード	1	
抵抗 22kΩ	1	赤赤橙金の色帯			

【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**

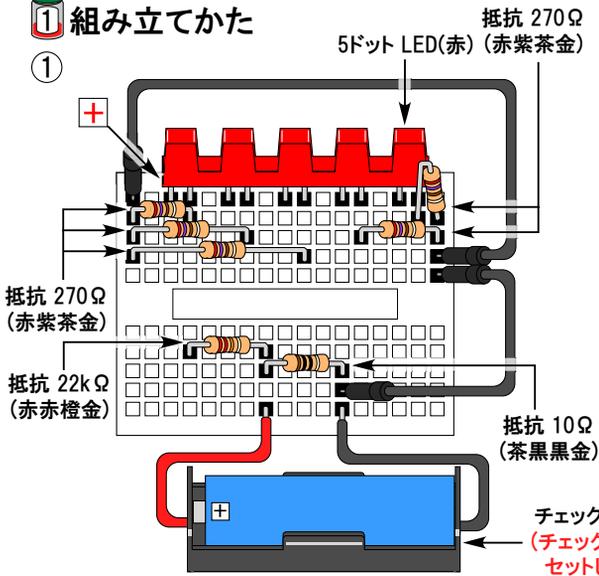
年中無休・営業時間: AM11:00~PM8:00  
〒566-0006 大阪府浪速区日本橋4-6-7  
TEL:06-6644-4555 / FAX:06-6644-1744  
[E]http://digit.yohritsu.com  
[Blog]http://blog.digit-parts.com [Twitter]@0666444555

【販売窓口】

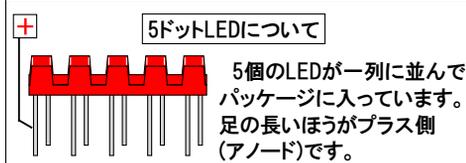
- シリコンハウス (大阪・日本橋店舗) 06-6644-4446
- デジット (大阪・日本橋店舗) 06-6644-4555
- 法人営業部 (G2日/学校/管代行) 06-6646-0707
- 通販営業部 (インターネット通販) 06-6644-6116

共同電子 検索

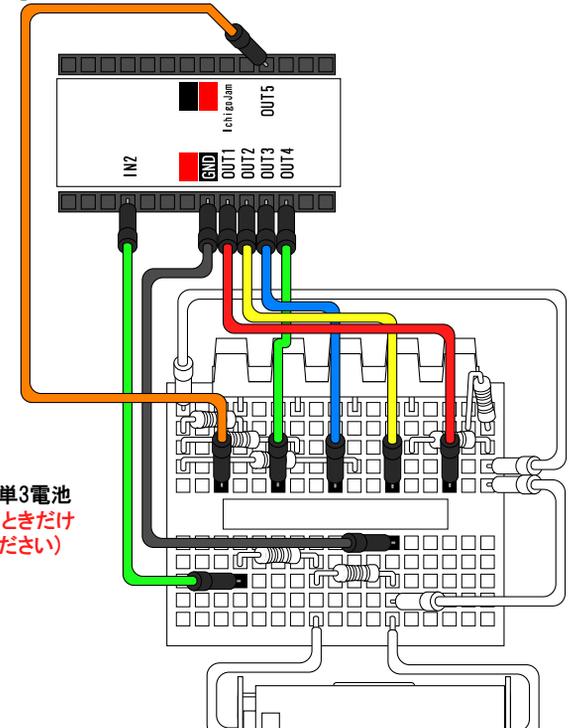
### 1 組み立てかた



※LEDは足の長い側が図のプラスマークの側に来るように差してください。



### 2 IchigoJamとの接続



※ジャンプワイヤーの色は配線の区別のため色分けしてあります。何色の線で配線してもかまいません。

## 2 はじめのプログラム

### 注意!!

本キットは実際に電池から電流を取り出して能力をチェックしています。電池ホルダに電池をセットし放しにすると電池がなくなってしまうので、チェックするときだけ電池をセットするようにしてください。

電池ホルダに電池をセットすると、アナログ入力(IN2端子)の電圧に対応した0から1023の数字をモニタ画面に表示します。

### プログラムリスト

```
1 REM カデ'ナ チェカ VER. 1. 0
2 REM 2015/07/09
10 PRINT "カデ'ナ チェカ"
20 V=ANA(2)
30 PRINT V
40 WAIT 10
50 GOTO 20
```

### IchigoJam バージョン0.9.7の場合

IchigoJam バージョン0.9.7を使用する場合は、プログラムの20行を次のように変更してください。

```
20 V=ANA( )
```

赤の下線部分が  
変更箇所です

プログラムを1行だけ変更したいときは、変更したい行番号で文を新しく書き、最後に[ENTER(↵)]キーを押すと新しく書いた文で置き換わります。

## 部品と回路の説明

### 電池の能力チェックのしかた

モータで動くおもちゃの電池が古くなると、しだいに動きが遅くなってきます。なぜでしょうか？モータの回転が電源の電圧が高いほど速くなることは皆さんも理科の実験などで体験していると思いますので、きっと「電池は古くなると電圧が下がってきて、それでモータの動きが遅くなるんだな」と予想することでしょう。

その予想は半分当たっているのですが、本当は「電池は古くなると電流を取り出す能力が落ちてきて、モータに電流を流したときの電池の電圧が下がってしまうから」です。おうちに「テスター」という測定器がありましたら、モータをつながない状態で電池単体の電圧を測ってみましょう。よほど電池が古くない限り、新品の電池とそれほど変わらない電圧が出ているはずですよ。

このように、電池単体の電圧を測っても電池の能力チェックはできません。電池の能力をチェックするには、電池から実際に電流を取り出し、そのときの電圧を測る必要があります。

本キットでは、電池に10Ωの抵抗(※)をつないで電流を取り出し、そのときの電圧を測ることで電池の能力チェックをしています。(※単3電池の能力を測るときは負荷抵抗としてJIS規格で定められています)

### プログラムの説明

プログラムをRUNすると、約167msおきにIchigoJamのアナログ入力(IN2端子)の値を取り込み、0から1023の数字をモニタ画面に表示します。

アナログ入力の値を取り込むには、ANA(2)関数を使います。この機能は、IchigoJamのIN2端子とGND端子間のアナログ電圧を0～1023の数値で返します。詳細は「キーワードの説明」で説明します。

```
!
50 GOTO 20
```

```
SAVEO
```

プログラムを書いたあと「SAVEO」でプログラムを保存してください。(O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)
- 変数 V
- WAIT文
- PRINT文の使い方
- ANA(2)関数



### SAVEコマンドの使い方

```
!
130 GOTO 30
```

```
SAVEO
```

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。

```
LOADO
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。

## 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)を入れるために使用します。

(例) REM スイッチが閉じると

「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います。

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 "ホンゾウ セイテンナリ"

この例では「ホンゾウ 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。文字列はダブルクォーテーション「"」で囲みます。

PRINT文で表示させる項目をセミicolon「;」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。

## 🔑 変数 V (20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)

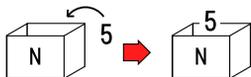
箱に式の計算結果や別の箱の中身を入れる(コピーする)ことを「代入」といいます。  
(※コピー元の箱の中身は変化しません)

(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)

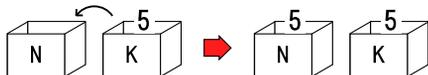
(参考2: IchigoJamのBASICで扱える数値の範囲は、-32768から32767までの整数です)

変数に数値を代入するには、次のように書きます。

(例) N = 5



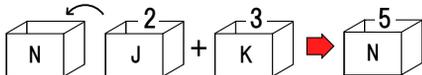
(例) N = K



※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います。

(例) N = J + K



## 🔑 ANA(2)関数 (20行)

ANA(2)関数は、IchigoJamのアナログ入力端子(IN2端子)とGND端子間のアナログ電圧を0~1023の整数で返します。

IchigoJam バージョン0.9.7のプログラムでANA()関数を使用しているプログラムをIchigoJam バージョン1.0.1で動作させるには、次のように書き換えてください。

<バージョン1.0.1>                      <バージョン0.9.7>  
(例) V=ANA (2)                      V=ANA ()

IchigoJam バージョン1.0.1では、かつこ内の数字(2)を省略するとBTN端子のアナログ電圧を取り込みます。  
**IN2端子のアナログ電圧を取り込むにはANA(2)と書く必要があります。**

「ANA(2)」という名前の箱の中にIN2端子とGND端子間のアナログ電圧を0~1023の整数で表したものがいつも入っています。箱の中身はプログラム中でいつでも取り出すことができます。

アナログ電圧  
(0~1023)

IN2の電圧(V)	ANA(2)の値
3.300	1023
3.296	1022
3.294	1021
⋮	⋮
0.006	2
0.003	1
0.000	0

IN2端子の電圧とANA(2)の値の関係は左の表のようになっていますので、ANA(2)の値を調べることで、IN2端子の電圧を知ることができます。

(例) N=ANA(2)

この例は、IN2端子のアナログ電圧に対応した値を変数Nに代入しています。

## 🔑 WAIT文 (40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(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。

### 3 入門プログラム

はじめのプログラムを改良して、LEDのドットの数で電池の放電能力を表示させます。

#### プログラムリスト

```
1 REM カデ`ンチ チェッカ VER. 1. 0
2 REM 2015/07/09
10 PRINT `カデ`ンチ チェッカ`
20 V=ANA(2)
30 PRINT `V=`;V
40 IF V>=186 THEN OUT 1, 1 ELSE OUT 1, 0
50 IF V>=248 THEN OUT 2, 1 ELSE OUT 2, 0
60 IF V>=310 THEN OUT 3, 1 ELSE OUT 3, 0
70 IF V>=372 THEN OUT 4, 1 ELSE OUT 4, 0
80 IF V>=434 THEN OUT 5, 1 ELSE OUT 5, 0
90 WAIT 10
100 GOTO 20
```

※IchigoJam バージョン0.9.7を使用する場合は、プログラムの20行を次のように変更してください。

```
20 V=ANA() ← 変更箇所
```

#### キーワード

IF文と条件判断

### キーワードの説明

#### 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用します。

```
(例) IF V>=186 THEN OUT 1, 1
```

条件式      条件式が真のとき  
実行する文

この例では、Vの値が186以上かどうか比較し、もし真ならば(以上ならば)「THEN」以下の文を実行します。(偽ならば(未満ならば)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 V>=186 THEN OUT 1, 1 ELSE OUT 1, 0
```

条件式      条件式が真のとき      条件式が偽のとき  
実行する文      実行する文

この例ではVの値が186以上かどうか比較し、もし真ならば(V>=186ならば)OUT1端子に「1」を、偽ならば(V<186ならば)OUT1端子に「0」を出力します。

### プログラムの説明

電池の電圧(V)	取り込み結果の数値
1.4	434
1.2	372
1	310
0.8	248
0.6	186

このプログラムでは、IN2端子のアナログ電圧の値(ANA(2)の値)をもとに、電池の放電能力をLEDのドット数で表現します。

電池の電圧は0.6Vから1.4Vまで、0.2V刻みで5段階の表示ができます。

電池の電圧とANA(2)の値の関係は左表の通りです。

### 4 上級プログラム

入門プログラムを改良して、電池電圧の測定結果を5ドットのLEDで表示する以外に、画面に電圧値を表示するようにします。

電池の電圧を0.1V刻みで表示するため、電圧値の計算のしかたを工夫しています。

#### プログラムリスト

```
1 REM カデ`ンチ チェッカ VER. 1. 0
2 REM 2015/07/09
10 CLS
20 PRINT `カデ`ンチ チェッカ`
30 V=ANA(2)
40 REM デン`アツ ニ カザ`ン
50 W=(V*30)/512+(V*3)/512
60 IF (W%2)=0 THEN W=W/2 ELSE W=W/2+1
70 LOCATE 0, 1:PRINT ` `
75 LOCATE 0, 1:PRINT V
80 PRINT W/10;`. `:W%10;`V`
90 IF W>=10 THEN OUT 1, 1 ELSE OUT 1, 0
100 IF W>=11 THEN OUT 2, 1 ELSE OUT 2, 0
110 IF W>=12 THEN OUT 3, 1 ELSE OUT 3, 0
120 IF W>=13 THEN OUT 4, 1 ELSE OUT 4, 0
130 IF W>=14 THEN OUT 5, 1 ELSE OUT 5, 0
140 WAIT 10:GOTO 30
```

※IchigoJam バージョン0.9.7を使用する場合は、プログラムの30行を次のように変更してください。

```
30 V=ANA() ← 変更箇所
```

#### キーワード

CLS文

LOCATE文

割り算の余り

コロン(「:」)記号

### プログラムの説明

#### (1) ANA(2)関数の値を電圧に換算する

30行のANA(2)関数は、IN2端子とGND端子間のアナログ電圧を0~1023の値で返します。

このプログラムでは、画面に電圧を表示するため、ANA(2)関数で得られた値を0.1V刻みの数字に換算してから処理しています。

IchigoJamのBASICは整数しか扱えないため、このプログラムでは次のように数を扱っています。

(例) 1.5の表記

```
015
```

▲ 小数点の位置      左の例は1.5をこのプログラムで表現したものです。プログラムを書くときに小数点が1の桁の左にあるものと考えています。固定小数点数の1.5を10倍して「15」という整数として扱っています。

ANA(2)関数で得られた値を0.1V刻みの数字に換算するには、次の式を用います。

$$\text{電圧(0.1V単位)} = \frac{\text{ANA(2)の値} \times 3}{1024}$$

ANA(2)関数の値を実際の電圧に換算するには、上の式をIchigoJamのBASICで書けばよいのですが、IchigoJamのBASICで扱える数は、-32768~32767の範囲の整数のみです。上の式をそのままプログラムに書くと計算途中の結果が32767を超えてしまい(オーバーフローといいます)正しく計算できないことがあります。

そこでプログラム上では50行のように30倍と3倍の項に分けて計算し、結果を足しています。(30倍したあとに512で割っているところが計算方法のポイントです)

※1024ではなく512で割っているのは、60行で四捨五入の処理を行うためです

## (2) 四捨五入

(1)の方法でANA(2)関数の値を実際の電圧値に換算しました。(0.1V刻み)このプログラムでは実際の電圧値に換算するときに四捨五入の処理を行っています。

たとえば、実際の電圧が1.37Vであると仮定します。この電圧を小数第1位までの数で表現する必要があるとき、どう表現するのが自然でしょうか？ おそらくは「1.4V」と表現するのが自然だと思います。(1.3V台後半だから)

プログラム上で四捨五入の処理を行うため、電圧値に換算するときに次のようにしています。

```
50 W=(V*30)/512+(V*3)/512
60 IF (W%2)=0 THEN W=W/2 ELSE W=W/2+1
                                (プログラムより抜粋)
```

(1) 値が最終の値の2倍になるようWを計算します。(50行)

(2) Wを2で割った余りが「0」ならばWの値を単純に2で割り、「1」ならばWの値を2で割ったものに1足したものを求めるWの値とします。(60行)

IchigoJamのBASICは整数しか扱えないため、(1)では値が最終の値の2倍になるようWを計算しています。これにより電圧を0.05V単位で計算します。

このWの値を2で割った余りが「0」のときは電圧の小数第2位が「5未満」であることを表し、「1」のときは「5以上」であることを表します。

最終的に求めるWの値は0.1V単位ですので、(2)でWの値を2で割っています。このとき電圧の小数第2位が5未満であれば小数第2位を切り捨て、5以上であれば小数第2位を切り上げるようにします。

## 📖 キーワードの説明

### 🔑 CLS文 (10行)

CLS文は、画面の表示をクリアする(何も表示されていない状態にする)ためのコマンドです。

(例) CLS

CLS文がクリアするのは画面の表示だけです。プログラムや変数の値などはクリアされません。

### 🔑 割り算の余り (60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 (W%2)=0 THEN W=W/2 ELSE W=W/2+1

割り算の余りを求める演算子は「%」です。この例では、K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。

### 🔑 LOCATE文 (70行)

LOCATE文は、次に実行するPRINT文の画面上の表示位置を指定するのに使います。

表示する位置は、LOCATEの後ろに「横の位置」「縦の位置」の順で指定します。表示位置は横の位置は画面左端、縦の位置は画面上端を「0」として数えます。

(※「1」からではありません。間違えやすいので注意してください)

(例) LOCATE 0, 1

この例は、次に実行するPRINT文の表示位置を、画面の2行目の一番左端(1文字目)にします。

### 🔑 コロン(「:」)記号(70行)

BASICのプログラムでは、1行に複数の文を書くことができます。1行に複数の文を書くときは、文と文の間をコロン(「:」)記号で区切って書きます。

(例) N=RND(6)+1:REM ランクヲ 卦ル