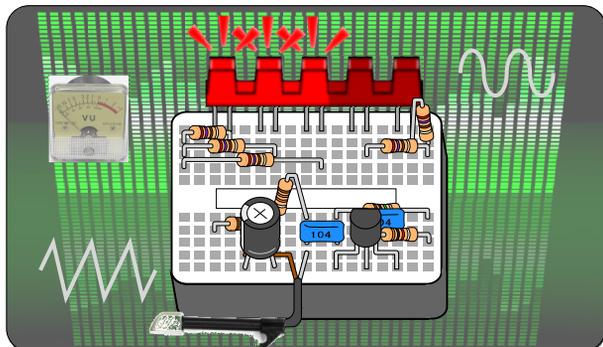


0 概要



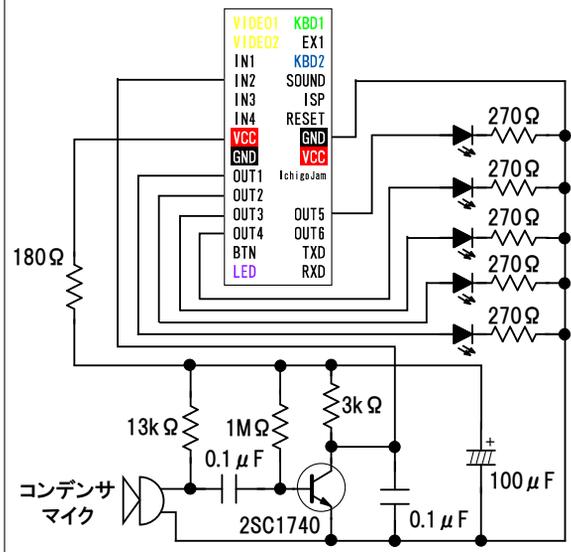
マイクで拾った音の大きさを5ドットのLEDで表示します。
コンデンサマイクの出力をトランジスタ1石で検波してサウンドレベルに
に応じた直流電圧に変換し、IchigoJamのアナログ入力に入力します。

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。

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		抵抗 13kΩ	1	茶橙緑金の色帯
トランジスタ 2SC1740	1		抵抗 1MΩ	1	茶黒緑金の色帯
コンデンサマイク	1		積層セラミックコンデンサ 0.1μF	2	104の表記
抵抗 180Ω	1	茶灰茶金の色帯	電解コンデンサ 100μF	1	
抵抗 270Ω	5	赤黒茶金の色帯	ジャンプワイヤー 長さ10cm	12	
抵抗 3kΩ	1	橙黒赤金の色帯	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: AM11:00~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

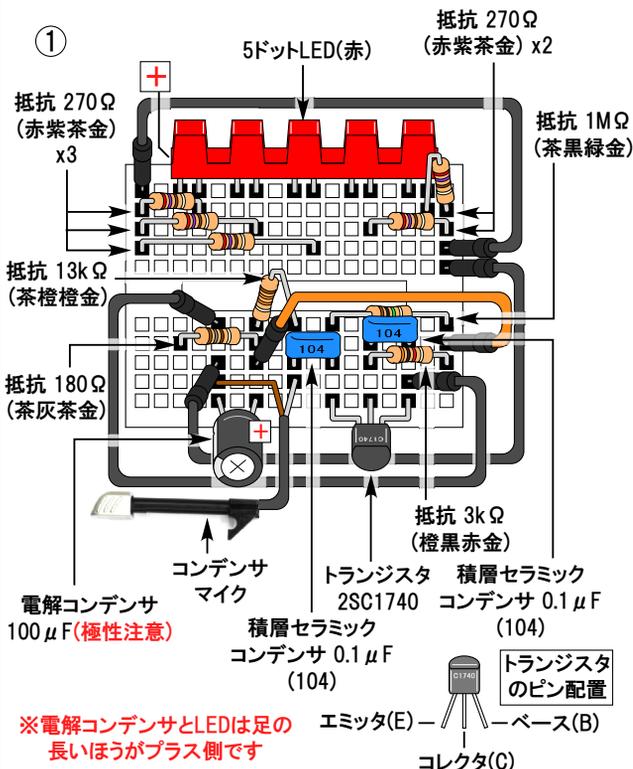


KYOHITSU
共立電子産業株式会社

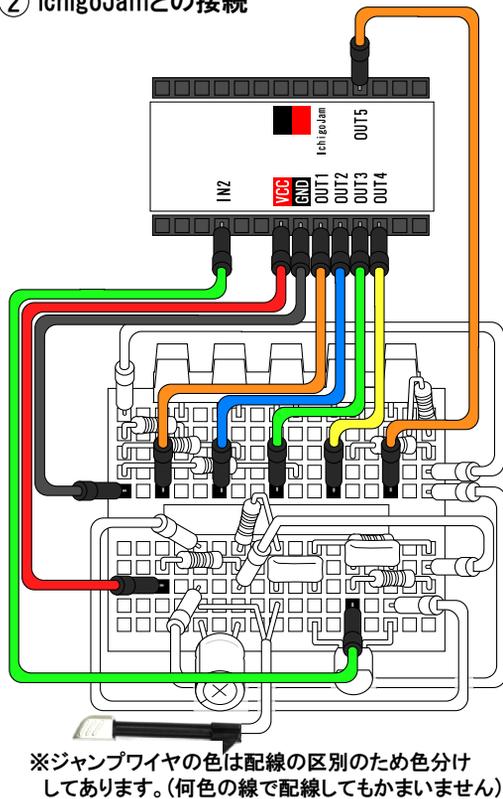
共立電子

検索

1 組み立てかた



2 IchigoJamとの接続



② はじめのプログラム

サウンドレベルを一定時間ごとに取り込み、モニタ画面にその値を表示するプログラムです。
マイクに向かって話しかけると、モニタ画面に表示される値が変化します。

プログラムリスト

```
1 REM サウト レベル メタ VER. 1.0
2 REM 2015/07/13
10 PRINT "サウト レベル メタ"
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
```

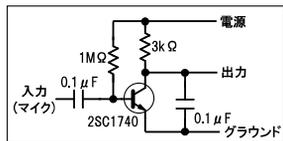
プログラムを1行だけ変更したいときは、変更したい行番号で文を新しく書き、最後に[ENTER(↵)]キーを押すと新しく書いた文で置き換わります。

キーワード

- 🔑 REM文(コメント)
- 🔑 ANA(2)関数
- 🔑 変数 V
- 🔑 WAIT文

📖 部品と回路の説明

トランジスタ1石で信号レベルをはかる



本キットの回路では、コンデンサマイクの音声信号出力をIchigoJamで扱えるレベルにするのに、トランジスタ1石による検波回路を使用しています。

マイクから入力信号が入ると、コレクタ電流の平均値が変化します。出力電圧が信号レベルに応じて変化するため、信号レベルをはかることができます。

📖 プログラムの説明

アナログ電圧の取り込み

IchigoJamにはアナログ電圧を取り込む機能(ANA(2)関数)があります。(ANA(2)関数については「キーワードの説明」を参照してください)

プログラムをRUNしてマイクに向かって話しかけると、一定時間(約167ミリ秒)ごとにサウンドレベルに応じたアナログ電圧を取り込み、モニタ画面に表示します。

📖 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 スイッチが押されたら

「REM」の後ろはプログラムの実行時は無視されます。

```
⋮
40 WAIT 10
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。
(↵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。

📖 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コマンドでプログラム番号を指定すると呼び出すことができます。

🔑 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。

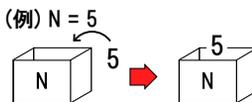
📦 コピー(代入)

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)

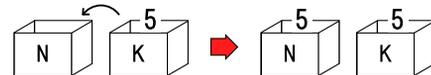
(参考) :BASICでは変数を使用する前に「この名前の変数を使用します」と宣言する必要はありません)

(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)

「N」という名前の変数に数値を代入するには、次のように書きます。



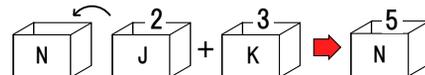
(例) N = K



※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



🔑 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、かっこ内の数字(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。



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)の値を調べるにより、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(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。

③ 入門プログラム

はじめのプログラムを改良して、サウンドレベルを5ドットのLEDで表示するようにします。

サウンドレベルを取り込み、そのレベルに応じて5ドットのLEDを棒グラフ状に点灯させます。

プログラムリスト

```
1 REM サウンドレベルメータ VER. 1.0
2 REM 2015/07/13
10 PRINT "サウンドレベルメータ"
20 REM ショートノセッテイ
30 K=640
40 REM メインループ
50 V=ANA(2)
60 W=V-K
70 PRINT W
80 IF W>=10 THEN OUT 1,1 ELSE OUT 1,0
90 IF W>=20 THEN OUT 2,1 ELSE OUT 2,0
100 IF W>=30 THEN OUT 3,1 ELSE OUT 3,0
110 IF W>=40 THEN OUT 4,1 ELSE OUT 4,0
120 IF W>=50 THEN OUT 5,1 ELSE OUT 5,0
130 WAIT 5
140 GOTO 50
```

Kの値(補正值)の設定方法については、「プログラムの説明」をご覧ください。

※IchigoJam バージョン0.9.7を使用する場合は、プログラムの50行を次のように変更してください。

50 V=ANA() ← **変更箇所**

キーワード

🔑 IF文と条件判断

📖 プログラムの説明

プログラムをRUNしてマイクに向かって話しかけると、サウンドレベルを5ドットのLEDで棒グラフ風に表示します。

はじめのプログラムで実験したように、マイクに向かって話しかけると、サウンドレベルに応じてアナログ入力の値が無信号時の値から増えます。

そこで、このプログラムでは音がしていないときのアナログ入力の値を補正值としてあらかじめ用意しておき、一定時間ごとにはかったアナログ入力の電圧との差をサウンドレベルとして扱っています。

補正值(K)の値の決め方(30行)

本キットに使用している部品(特にトランジスタ)の特性のばらつきにより、画面に表示されるサウンドレベルの値がずっとマイナスのままの場合や、大きすぎる場合があります。

このような場合はLEDの点灯がうまくいきませんので、プログラム30行のKの値を変更する必要があります。

```
PRINT ANA(2)
635
OK
30 K=635
```

プログラム30行で設定するKの値を求めるには、マイクになるべく音が入らないようにして、「PRINT ANA(2)」と入力します。
(行番号は不要です)

画面に補正值を表す数字が表示されますので、プログラム30行の数字を表示された数字に変更してください。

参考(BASICの文の即時実行)

行番号なしでBASICの文を入力し、[ENTER]キーを押すと、入力した文を直ちに実行することができます。

📖 キーワードの説明

🔑 コロン(:)記号(30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) V=ANA():PRINT V

🔑 ループ構造(30行~60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。繰り返しのループ構造を書くには、次のように書きます。

(例)

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

🔑 BTN()関数(180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」、押されていない場合は「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)

📖 5 上級プログラム・改

上級プログラムを改良して、LEDの点灯具合でVUメータの針の振れ具合をまねてみます。



VUメータは左の写真のようなメータで、音声や音楽の信号レベルを表示するのに使います。針の振れ具合が実際の音量の感じに合うよう、針が少しゆっくり振れるように工夫してあります。

プログラムリスト

```
1 REM サウンドレベルメータ VER. 1.0
2 REM 2015/07/13
10 PRINT "サウンドレベルメータ"
20 REM ヨコキ / セッテイ
25 W=0
30 I=16:K=0:PRINT "ヨコキ シタイズ"
40 K=K+ANA(2)
50 WAIT 10
60 I=I-1:IF I>0 THEN GOTO 40
70 K=K/16
80 REM メインループ
90 V=ANA(2)
100 X=V-K
105 W=(W+X)/2
110 PRINT W
120 IF W>=10 THEN OUT 1,1 ELSE OUT 1,0
130 IF W>=20 THEN OUT 2,1 ELSE OUT 2,0
140 IF W>=40 THEN OUT 3,1 ELSE OUT 3,0
150 IF W>=80 THEN OUT 4,1 ELSE OUT 4,0
160 IF W>=160 THEN OUT 5,1 ELSE OUT 5,0
170 WAIT 1
180 IF BTN()=1 THEN GOTO 25 ELSE GOTO 90
```

※IchigoJam バージョン0.9.7を使用する場合は、プログラムの40行と90行を次のように変更してください。

40 K=K+ANA() ← 変更箇所

90 V=ANA() ← 変更箇所

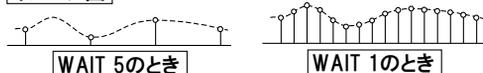
📖 プログラムの説明

表示の変化をゆっくりにするには

このプログラムでは、170行のWAIT文の待ち時間を「1」に指定しています。(上級プログラムでは待ち時間を「5」に指定していました)これは、拍手の音などの突発的な音もできるだけうまく拾えるようにと考えてのことです。

一定時間内になるべく多くの回数ANA(2)関数を呼び出すようにすると、突発的な音も拾えるチャンスが増えることになります。

イメージ図



上図の○印のところでIN2端子のアナログ電圧がIchigoJamのプログラムに取り込まれます。

このプログラムでは、サウンドレベルをきめ細かく取り込みながらも、LEDの点灯具合をVUメータ風になめらかに変化するようにしています。プログラムの105行がこのための計算です。

105 W=(W+X)/2

(プログラムより抜粋)

変数Xには、サウンドレベルの値が入っています。変数Wは記憶用の変数で、フィルタした結果が入っています。

新しい値を取り込むごとに、変数Wの値とサウンドレベルの値Xとの平均を取り(足して2で割る)、それを結果とします。結果はまた次回値を取り込んだときに使いますので、変数Wに記憶しておきます。

この処理を行うと、表示の変化がなめらかになり、アナログ式(針式)のVUメータ風になります。