

# IchigoJam 電子工作パーツセット

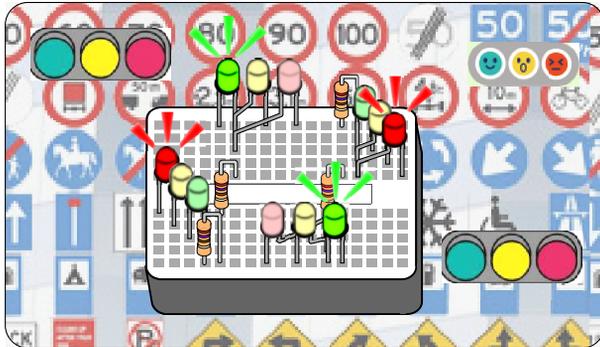
## LED信号機

SIG-jamP

### LED Traffic Signal Parts Set

IchigoJam 0.9.7 / 1.0.1対応

#### 概要



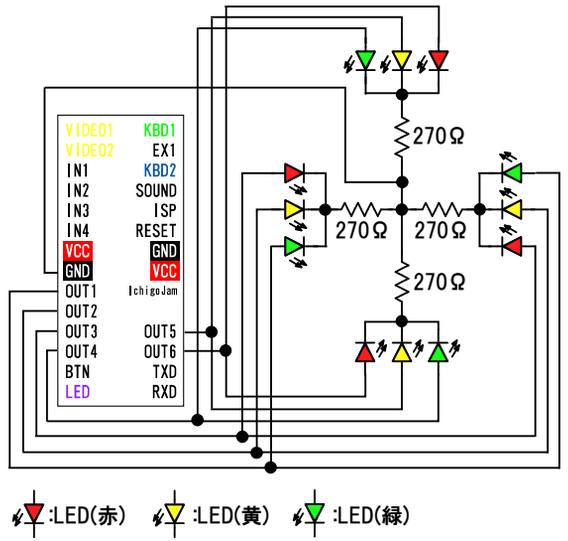
赤、緑、黄のLEDを交差点の信号機に見立てたLED信号機の部品セットです。プログラムをRUNすると、LEDの信号が赤→緑→黄の順で点灯し、交差点の信号機を再現します。

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	品名/型番/値	数量	備考
1 LED 赤	4		4 抵抗 270Ω	4	赤紫茶金の色帯
2 LED 黄	4		5 ジャンパワイヤ 10cm	14	
3 LED 緑	4		6 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・メトロ電子パーツ

## デジタル

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.kyofritsu.com

[Blog]http://blog.digit-parts.com [Twitter]@0666444555

【販売窓口】

- シリコハウス (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



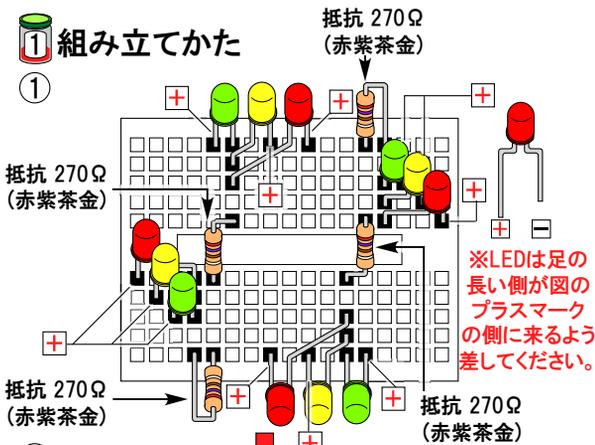
KYOFRITSU

共立電子産業株式会社

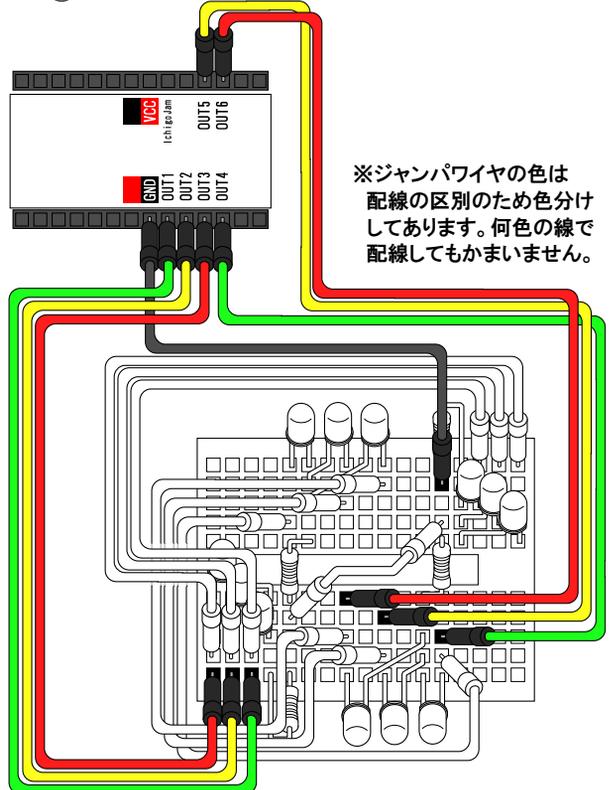
共立電子

検索

#### ① 組み立てかた



#### ③ IchigoJamとの接続



## 2 はじめのプログラム

組み立てた回路の動作チェックを兼ねて、6個のLEDを順番に点滅させてみます。

### プログラムリスト

```
1 REM LED ショウキ V1.0
2 REM 2015/07/10
10 PRINT "LED テスト"
15 PRINT "トウガイ モトリ"
20 OUT 1, 1:OUT 2, 0:OUT 3, 0
30 OUT 4, 0:OUT 5, 0:OUT 6, 0
40 WAIT 60
45 PRINT "トウガイ キ"
50 OUT 1, 0:OUT 2, 1:OUT 3, 0
60 WAIT 60
65 PRINT "トウガイ アガ"
70 OUT 1, 0:OUT 2, 0:OUT 3, 1
80 WAIT 60
85 PRINT "ナンボク モトリ"
90 OUT 1, 0:OUT 2, 0:OUT 3, 0
100 OUT 4, 1:OUT 5, 0:OUT 6, 0
110 WAIT 60
115 PRINT "ナンボク キ"
120 OUT 4, 0:OUT 5, 1:OUT 6, 0
130 WAIT 60
135 PRINT "ナンボク アガ"
140 OUT 4, 0:OUT 5, 0:OUT 6, 1
150 WAIT 60
160 GOTO 15
```

```
⋮
160 GOTO 15
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。

### 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前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。

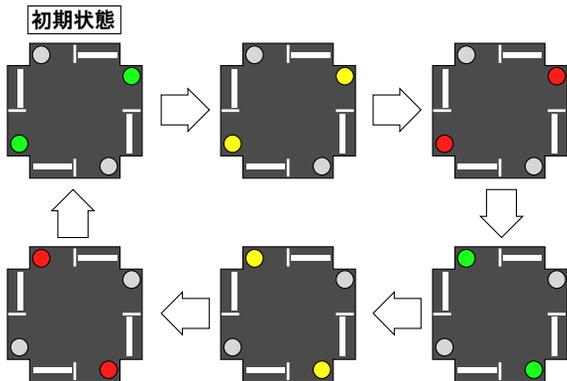
### キーワード

- REM文(コメント)
- PRINT文の使い方
- コロソ(「:」)記号
- WAIT文

### プログラムの説明

#### LEDの点滅について

プログラムが書いたら、SAVEしてからRUNしましょう。組み立てとプログラムに間違いがなければ、下図のように信号機のLEDが順番に点滅するはず。 (うまくいかない場合は、組み立てとプログラムの入力に間違いがないか、もう一度チェックしましょう)



本キットの回路では、IchigoJamの出力端子(OUT1~OUT6)を「1」にしたときLEDが点灯し、「0」にしたときLEDが消灯します。

プログラムでは、OUT1端子からOUT6端子まで順番に「1」にしてLEDを順番に点灯させています。LEDの点灯具合が見やすいように、LEDを点灯させたあとWAIT文で約1秒待っています。

### 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 スイッチヲ 扱マシマツ

「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 "ホンジツハ セイテンナリ"

この例では「ホンジツハ 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コロソ「:」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。

## 🔑 コロン「:」記号 (20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

## 🔑 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」で動作チェックができれば、交差点の信号機の動作を再現してみましょう。

### プログラムリスト

```
1 REM LED シグナル機
2 REM 2015/07/10
10 PRINT "LED シグナル機"
```

右上に続きます



## プログラムの続きです

```
20 I=1
30 IF I=1 THEN GOTO 110
40 IF I=2 THEN GOTO 150
50 IF I=3 THEN GOTO 180
60 IF I=4 THEN GOTO 210
70 IF I=5 THEN GOTO 240
80 IF I=6 THEN GOTO 270
90 I=I+1:IF I<=6 THEN GOTO 30
100 GOTO 20
110 PRINT "I=";I;"トウザイ=カ ナホク=カ"
120 OUT 1,0:OUT 2,0:OUT 3,1
130 OUT 4,0:OUT 5,0:OUT 6,1
140 WAIT 60:GOTO 90
150 PRINT "I=";I;"トウザイ=ミドリ ナホク=カ"
160 OUT 1,1:OUT 2,0:OUT 3,0
170 WAIT 180:GOTO 90
180 PRINT "I=";I;"トウザイ=キ ナホク=カ"
190 OUT 1,0:OUT 2,1:OUT 3,0
200 WAIT 60:GOTO 90
210 PRINT "I=";I;"トウザイ=カ ナホク=カ"
220 OUT 1,0:OUT 2,0:OUT 3,1
230 WAIT 60:GOTO 90
240 PRINT "I=";I;"トウザイ=カ ナホク=ミドリ"
250 OUT 4,1:OUT 5,0:OUT 6,0
260 WAIT 180:GOTO 90
270 PRINT "I=";I;"トウザイ=カ ナホク=キ"
280 OUT 4,0:OUT 5,1:OUT 6,0
290 WAIT 60:GOTO 90
```

## キーワード

🔑 変数 I

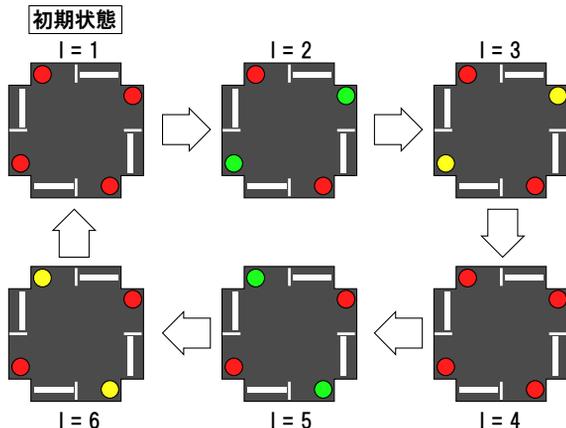
🔑 IF文と条件判断

## 📖 プログラムの説明

### (1) 信号機の表示について

このプログラムでは、赤緑黄の3個のLEDを使って交差点の信号機を再現しています。(本物の信号機の「青信号」は青味の緑ですが、本キットではLEDの緑(黄緑)で代用しています)

信号機の動作は、下の図のようになります。(説明の都合上、図の上側を北側とします)



- (I = 1) 初期状態(デフォルトの状態)では全ての信号が赤です。
- (I = 2) 東西側の信号を緑にします。
- (I = 3) 東西側の信号を黄にします。
- (I = 4) 東西側の信号を赤にします。
- (I = 5) 南北側の信号を緑にします。
- (I = 6) 南北側の信号を黄にします。

このプログラムでは、交差点の信号機の動作を表現するために、信号機の状態を表す変数を用意し、その変数の値(状態を表す1~6の数)に基づいてLEDの点灯/消灯の操作を行っています。

このように状態を表す変数を用意し、その変数の値に基づいていろいろな操作を行う手法は、プログラミングやデジタル回路の設計で広く使われています。(専門的には「状態(ステート)」に基づいて動く機械」ということで「ステートマシン」と呼ばれています)

### (2) 待ち時間について

このプログラムでは、信号の待ち時間は固定にしています。黄信号の時間(I=3またはI=6)と東西方向/南北方向ともに赤の時間(I=1またはI=4)は短いので1秒待ち、東西方向/南北方向のどちらかが緑信号の時間(I=2またはI=5)は3秒待つようになっています。

プログラム中170行と260行のWAIT文で指定している待ち時間を変更することで、東西方向と南北方向の信号待ち時間を変えることができます。実験してみてください。

#### 📖 参考(プログラムを1行だけ変更するには)

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。

#### もとのプログラム

```
170 WAIT 180:GOTO 90
```

#### 新しく文を書きます

```
170 WAIT 240:GOTO 90
```

[ENTER]キーを押すと180行が新しく書いた文で置き換わります

## 📖 キーワードの説明

### 🔑 変数 I (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。

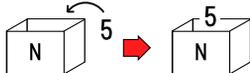
※コピー元の箱の中身は変化しません

(参考1: 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)

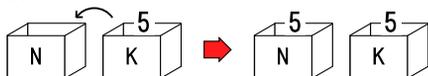
(参考2: IchigoJamのBASICで扱える数値の範囲は、-32768から32767までの整数です)

「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 = 5



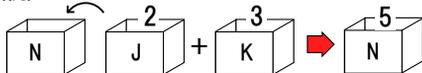
(例) N = K



※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



## 🔑 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。

## 📄 4 上級プログラム

はじめのプログラムでは、緑信号の長さは一定でした。交通量の少ないときは緑信号の時間を少し短くし(信号待ち時間を減らし)、交通量の多いときは緑信号の時間を延長すると、交通の流れがスムーズになりそうです。

はじめのプログラムを改造して、IchigoJam上のタクトスイッチを押すと緑信号の時間を延長する機能を追加します。

### プログラムリスト

```
1 REM LED シグナル
2 REM 2015/07/10
10 PRINT "LED シグナル"
20 I=1
30 IF I=1 THEN GOTO 110
40 IF I=2 THEN GOTO 150
50 IF I=3 THEN GOTO 180
60 IF I=4 THEN GOTO 210
70 IF I=5 THEN GOTO 240
80 IF I=6 THEN GOTO 270
90 I=I+1:IF I<=6 THEN GOTO 30
100 GOTO 20
110 PRINT "I=";I;"トウザイ=ア ナホク=アガ"
120 OUT 1,0:OUT 2,0:OUT 3,1
130 OUT 4,0:OUT 5,0:OUT 6,1
140 WAIT 60:GOTO 90
150 PRINT "I=";I;"トウザイ=ミドリ ナホク=アガ"
160 OUT 1,1:OUT 2,0:OUT 3,0
170 GOSUB 310:GOTO 90
```

右上に続きます

### プログラムの続きです

```
180 PRINT "I=";I;"トウザイ=キ ナホク=アガ"
190 OUT 1,0:OUT 2,1:OUT 3,0
200 WAIT 60:GOTO 90
210 PRINT "I=";I;"トウザイ=ア ナホク=アガ"
220 OUT 1,0:OUT 2,0:OUT 3,1
230 WAIT 60:GOTO 90
240 PRINT "I=";I;"トウザイ=ア ナホク=ミドリ"
250 OUT 4,1:OUT 5,0:OUT 6,0
260 GOSUB 310:GOTO 90
270 PRINT "I=";I;"トウザイ=ア ナホク=キ"
280 OUT 4,0:OUT 5,1:OUT 6,0
290 WAIT 60:GOTO 90
300 REM イチヨウキノウ ツキ ウェイト
310 W=12
320 IF BTN()=1 THEN W=W+1
330 WAIT 10
340 W=W-1:IF W>0 THEN GOTO 320
350 RETURN
```

### キーワード

- 🔑 サブルーチン(GOSUB/RETURN)
- 🔑 BTN()関数
- 🔑 ループ構造

## プログラムの説明

### 緑信号の時間延長機能

このプログラムでは、東西/南北のどちらかの信号機が緑のとき、タクトスイッチを押すことで緑信号の時間を延長できます。

この機能を実現するため、延長機能付きのウェイト用サブルーチンプログラムを用意し、緑信号のとき呼び出しています。  
(黄信号のときと、東西方向/南北方向ともに赤信号のときは時間延長は不要ですので時間固定のWAIT文を使用しています)

```

300 REM インチヨウノ タ ウェイト
310 W=12
320 IF BTN()=1 THEN W=W+1
330 WAIT 10
340 W=W-1:IF W>0 THEN GOTO 320
350 RETURN (プログラムより抜粋)

```

上のリストは、延長機能付きウェイト用サブルーチンプログラムの部分を抜粋したものです。

ループを1回まわると変数Wの値を1ずつ減らし、Wの値が0になつたらサブルーチンから戻るという構造になっています。(詳細は「キーワードの説明」をご覧ください)

ループの中で、IchigoJamのタクトスイッチを押しているかどうかを調べ(BTN()関数)、もし押していればWの値を1増やしています。もしタクトスイッチを押したままであれば、ループを何回まわっても変数Wの値は変わらず(W+1-1=Wとなるため)、時間延長ができることとなります。

## キーワードの説明

### サブルーチン(GOSUBとRETURN) (310~350行)

プログラム中の複数の箇所でも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」文を使用します。(※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
155 N=RND(6)+1
160 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 ←

```

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200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。

### BTN()関数 (320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みます。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)

### ループ構造 (310~340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
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。