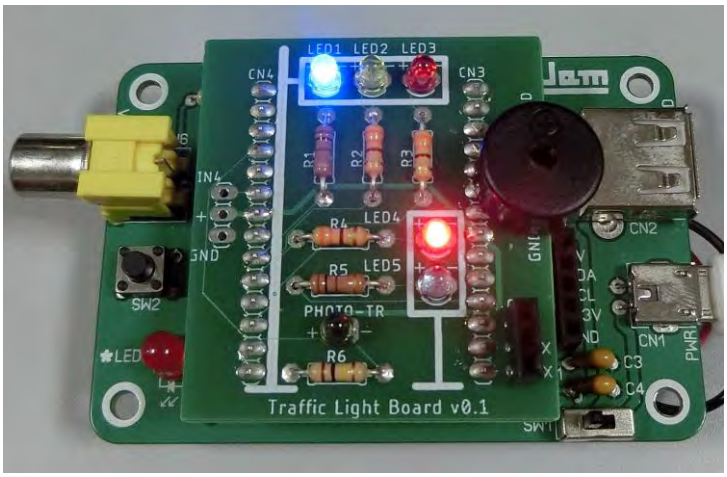
【B-1(4)】IchigoJam で信号機を作ろう（Ver.4）

●今回の目標

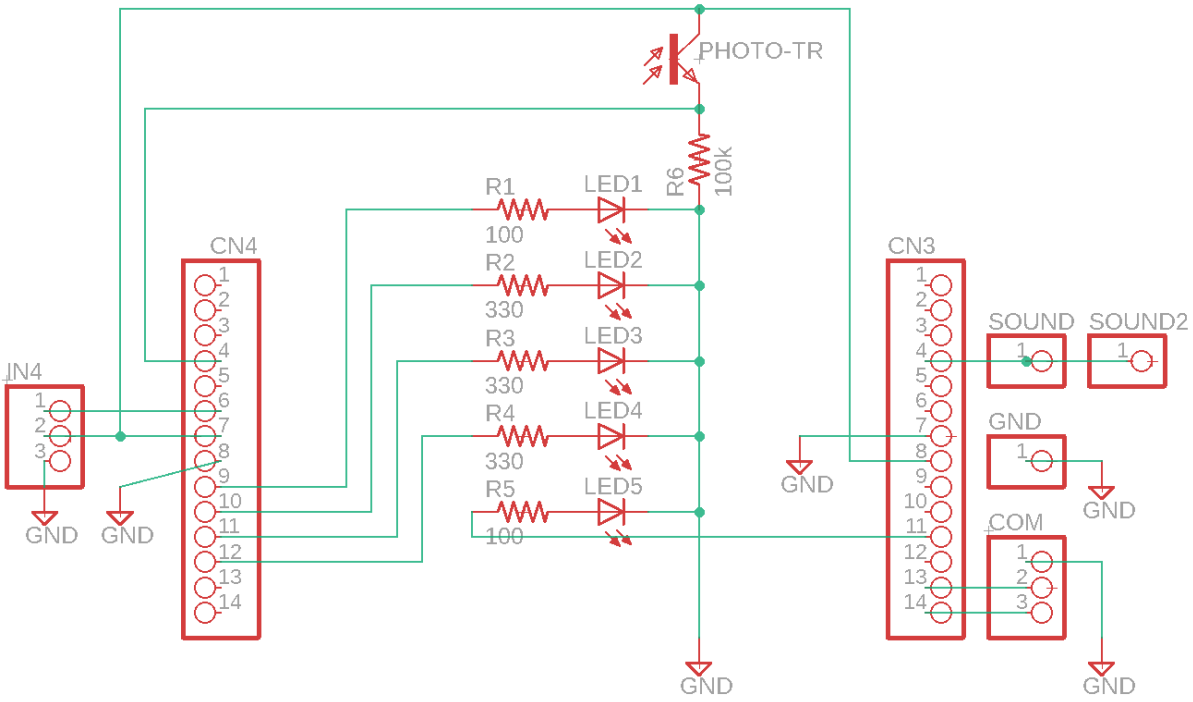
信号機基板を IchigoJam にさして、
BASIC プログラムで光らせます。



●信号機基板をさしこむ

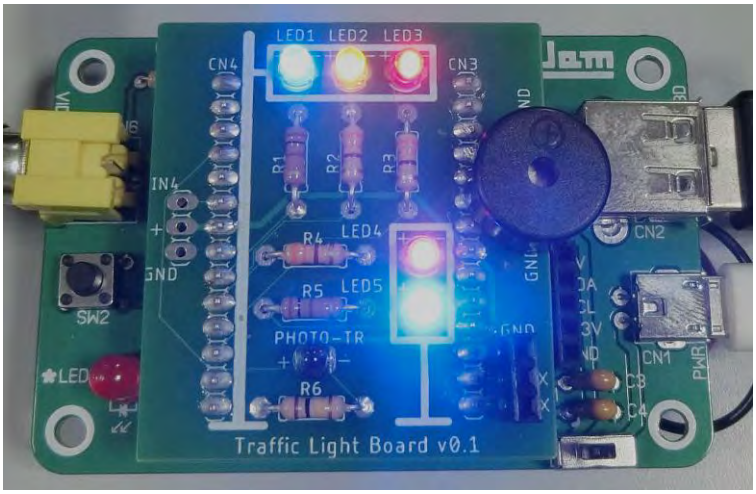
IchigoJam のソケットに、信号機基板のピンをさします。
上下をまちがえないようにしてください。
14 本のピン×2 列が、IchigoJam のソケットにささります。

回路図は以下のとおりです。



信号機基板が取り付けられたら、ちゃんとLEDが光るか、ダイレクトモードで確認しましょう。

```
OUT 1,1
OUT 2,1
OUT 3,1
OUT 4,1
OUT 5,1
```



車用と歩行者用の信号機の
青・黄・赤のLEDが光ります。
もし光らなかったら、基板の
さし方をまちがえています。
よく見直しましょう。

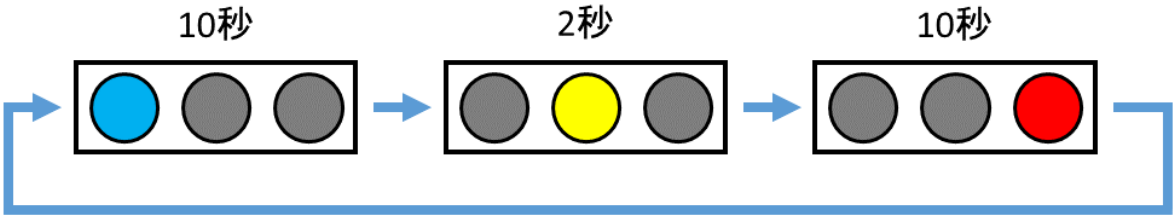
```
OUT 1,0
OUT 2,0
OUT 3,0
OUT 4,0
OUT 5,0
```

5個のLEDが消えます。

いろいろなパターンで信号機を光らせてみましょう。

●車用信号機を光らせる

まず、車用の信号機を光らせるプログラムを作ります。
信号機の動きを考えてみます。



今回は、青・黄・赤の点灯時間を上のようにしてみましょう。

```
10 '*TRAFFIC LIGHT   プログラムのタイトル
20 OUT 0             LED を全部消す
30 '*LOOP            ループの始め
40 OUT 1,1           青 LED を光らせる
50 WAIT 600          10 秒待つ
60 OUT 1,0           青 LED を消す
70 OUT 2,1           黄 LED を光らせる
80 WAIT 120          2 秒待つ
90 OUT 2,0           黄 LED を消す
100 OUT 3,1          赤 LED を光らせる
110 WAIT 600         10 秒待つ
300 OUT 3,0          赤 LED を消す
310 GOTO 30          30 行へもどる
```

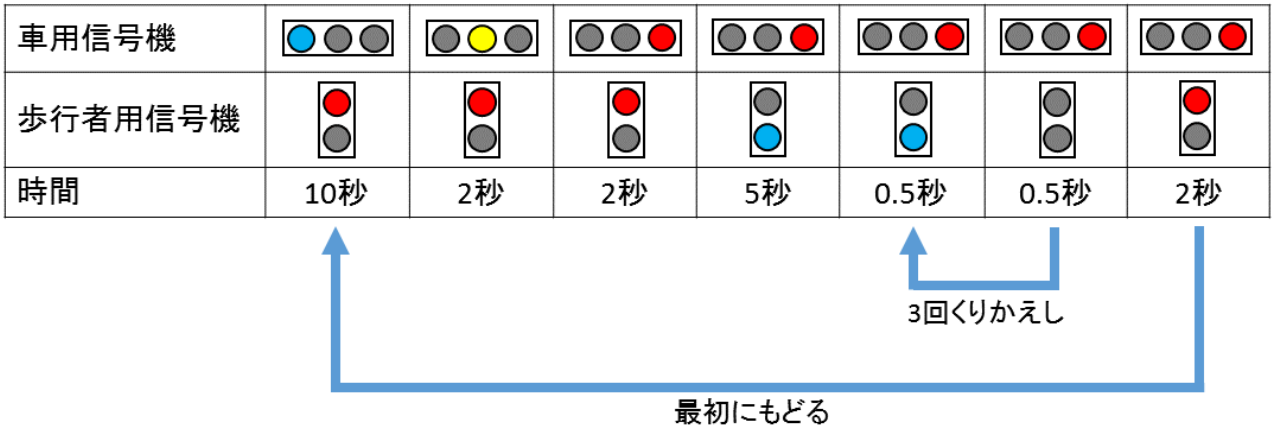
「RUN」でプログラムを実行してみましょう。
信号機が、青→黄→赤→青→…と動きます。

WAIT（ウェイト）命令の数値を変えると、それぞれの LED の点灯時間が変わります。
試してみましょう。

「SAVE 0」でプログラムを保存しておきましょう。

●歩行者用信号機を光らせる

車用信号機と歩行者用信号機を動かすプログラムを作ります。
2 つの信号機の動きをどうすればいいか、考えてみましょう。



- 車用信号機を「青→黄→赤」と動かします。この間、歩行者用信号機はずっと「赤」です。
- 車用信号機を「赤」にして、2 秒待ってから、歩行者用信号機を「青」にします。
  （すぐに「青」にしてしまうと、まだ車が来るかもしれないので危ないです）
- 歩行者用信号機の「青」を 5 秒間続けます。
- 歩行者に注意をうながすため、歩行者用信号機の「青」を 3 回点滅させます。
- 歩行者用信号機を「赤」にして、2 秒待ってから、最初にもどって車用信号機を「青」にします。
  （すぐに「青」にしてしまうと、まだ歩行者が歩いているかもしれないので危ないです）

それでは、前ページの表と説明を見ながら、プログラムを改造してみましょう。

```
10 '*TRAFFIC LIGHT
20 OUT 0
30 '*LOOP
40 OUT 1,1
45 OUT 4,1
50 WAIT 600
60 OUT 1,0
70 OUT 2,1
80 WAIT 120
90 OUT 2,0
100 OUT 3,1
110 WAIT 120
120 OUT 4,0
130 OUT 5,1
140 WAIT 300
150 FOR I=1 TO 3
160 OUT 5,1
170 WAIT 30
180 OUT 5,0
190 WAIT 30
200 NEXT
210 OUT 4,1
220 WAIT 120
300 OUT 3,0
310 GOTO 30
```

歩行者用信号の赤を光らせる
車用信号の赤の時間を 2 秒に変更
歩行者用信号の赤を消して
青を光らせる
5 秒待つ
3 回くりかえし
歩行者用信号の青を光らせる
0.5 秒待つ
歩行者用信号の青を消す
0.5 秒待つ
もどってくりかえし
歩行者用信号の赤を光らせる
2 秒待つ

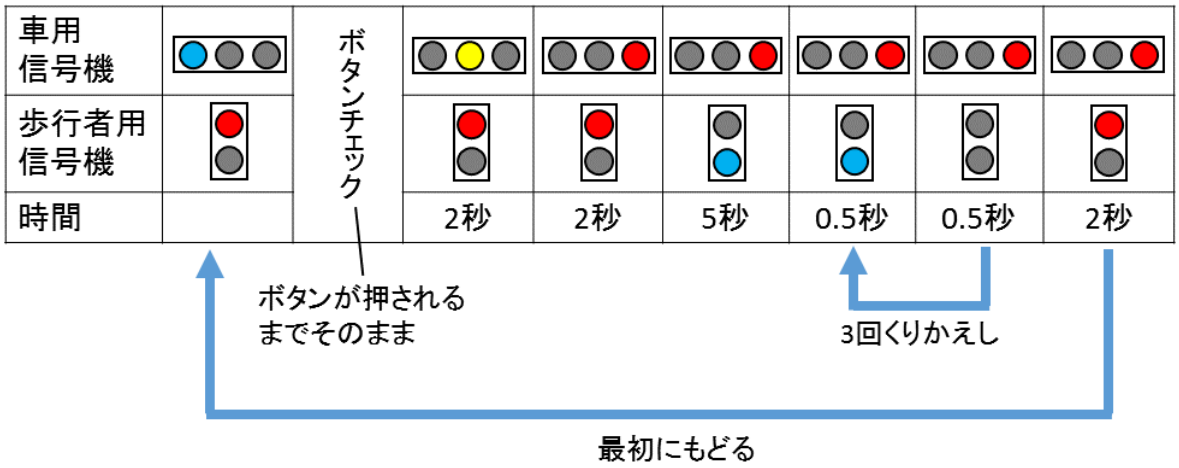
プログラムを実行してみましょう。
2 つの信号機が連動して動きます。

WAIT（ウェイト）命令の数値を変えると、それぞれの LED の点灯時間が変わります。
試してみましょう。

「SAVE 0」でプログラムを保存しておきましょう。

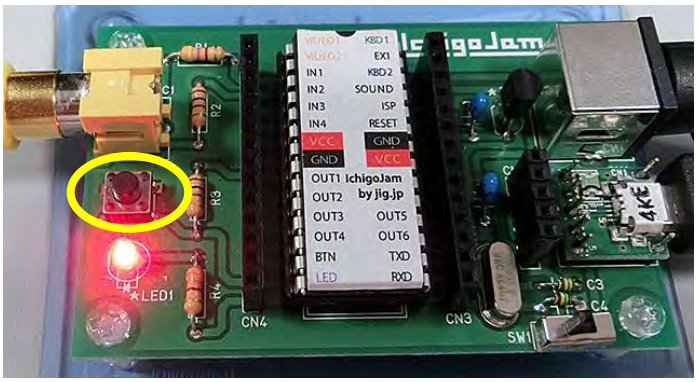
●押しボタン信号にする

プログラムを改造して、「ボタンを押したら歩行者用信号が青になる」信号にしてみましょう。
信号機の動きを表で考えます。



車用信号機が青の時に、ボタンが押されているかをチェックして、押されるまではそのまま変化させません。
ボタンが押されたら、黄色以降の処理に移ります。それ以降の動きは同じです。

ボタンは、IchigoJam の左側に付いているボタンを使います。



プログラムを改造します。
車用信号機を青にした後の、時間待ち（WAIT）の行を書きかえます。

```
10 '*TRAFFIC LIGHT
20 OUT 0
30 '*LOOP
40 OUT 1,1
45 OUT 4,1
50 IF BTN()=0 THEN GOTO 50
55 WAIT 180
60 OUT 1,0
70 OUT 2,1
80 WAIT 120
90 OUT 2,0
100 OUT 3,1
110 WAIT 120
120 OUT 4,0
130 OUT 5,1
140 WAIT 300
150 FOR I=1 TO 3
160 OUT 5,1
170 WAIT 30
180 OUT 5,0
190 WAIT 30
200 NEXT
210 OUT 4,1
220 WAIT 120
300 OUT 3,0
310 GOTO 30
```

ボタンが押されていなかったら、この行をくりかえし

3 秒待つ

プログラムを実行してみましょう。
車用信号がずっと青のままになります。
ボタンを押すと、3 秒後に黄色→赤に変わります。
※車用信号が青になった直後にボタンが押された場合、すぐに黄色になっても車は急に止まれないので、3 秒待つようにしています。

「もしボタンが押されていたら」という条件判断には、IF（イフ）、THEN（ゼン）、ELSE（エルス）命令を使います。

```
IF BTN()=0 THEN GOTO 50 ELSE ～
```

条件式　　条件が成り立つ時に実行　　条件が成り立たない時に実行

条件式	判断する条件。
条件が成り立つ時に実行	条件式が成り立つ時に、この部分を実行する。
条件が成り立たない時に実行	条件式が成り立たない時に、この部分を実行する。 ELSE 以下は省略可能。

今回は条件式で「ボタンが押されているかどうか」を判断するために、BTN（ボタン）関数を使っています。

```
BTN()
```

関数の値	ボタンが押されている＝1 ボタンが押されていない＝0

今回は、「もしボタンが押されていなかったらそのまま待つ」動きにしたいので、GOTO 命令でそのまま 50 行へもどっています。

●「おまちください」とお知らせする

ボタンを押してから 3 秒待たされると、「ちゃんとボタンが押せたのか？」と不安になります。
現実の押しボタン信号では、「おまちください」のランプが点いて歩行者にお知らせします。
IchigoJam についている LED を、お知らせランプとして点灯させましょう。

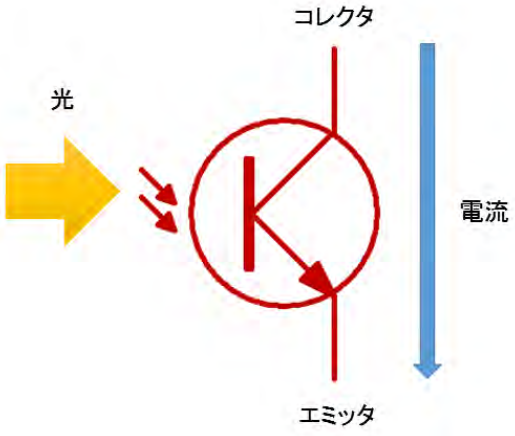
```
10 '*TRAFFIC LIGHT
20 OUT 0
30 '*LOOP
40 OUT 1,1
45 OUT 4,1
50 IF BTN()=0 THEN GOTO 50
52 LED 1          LEDを光らせる
55 WAIT 180
60 OUT 1,0
70 OUT 2,1
80 WAIT 120
90 OUT 2,0
100 OUT 3,1
110 WAIT 120
120 OUT 4,0
130 OUT 5,1
135 LED 0         LEDを消す
140 WAIT 300
150 FOR I=1 TO 3
160 OUT 5,1
170 WAIT 30
180 OUT 5,0
190 WAIT 30
200 NEXT
210 OUT 4,1
220 WAIT 120
300 OUT 3,0
310 GOTO 30
```

プログラムを実行してみましょう。
ボタンを押すと、IchigoJam の LED が光ってお知らせします。
歩行者用信号が青になると、LED が消えます。

「SAVE 0」でプログラムを保存しておきましょう。

●夜は黄色の点滅にする

夜になると、黄色の点滅になる信号機があります。
光センサーで明るさを読み取って、信号機を動かしてみましょう。
今回は光センサーとして、**フォトトランジスター**を使います。
フォトトランジスターは、光が当たると
電流が流れる部品です。



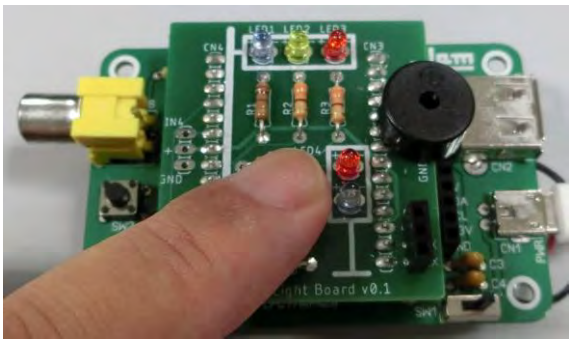
まず、フォトトランジスターがちゃんと動作するか確認しましょう。
以下のプログラムを入力します。

```
1 PRINT ANA(2):GOTO 1
```

プログラムを実行してみましょう。画面に数字が連続で表示されます。
まわりの明るさによりますが、だいたい 900 を越えたくらいの値に
なるはずです。

```
945
930
944
 :
```

フォトトランジスターに指をおいて、かげにしてみましょう。
数字が 100 以下くらいに小さくなります。
（まわりの明るさによって数字は変わります）



確認ができたら、

```
1 (Enter)
```

と入力して、1 行目を消しましょう。

「ANA( )」は、アナログ入力を読み取る関数です。

```
ANA(     2     )
```
ポート番号

ポート番号は、2（IN2 端子）、5～8（OUT1～4 端子）、0 か 9（BTN 端子）が使える。
返ってくる値は、0（0V）～1023（3.3V）。

それでは、夜になったら黄色の点滅になる信号機を作りましょう。

```
10 '*TRAFFIC LIGHT
20 OUT 0
30 '*LOOP
40 OUT 1,1
45 OUT 4,1
47 IF ANA(2)<300 THEN GOSUB 400:GOTO 60
50 IF BTN()=0 THEN GOTO 47
52 LED 1
55 WAIT 180
60 OUT 1,0
70 OUT 2,1
80 WAIT 120
90 OUT 2,0
100 OUT 3,1
110 WAIT 120
120 OUT 4,0
130 OUT 5,1
135 LED 0
140 WAIT 300
150 FOR I=1 TO 3
160 OUT 5,1
170 WAIT 30
180 OUT 5,0
190 WAIT 30
200 NEXT
210 OUT 4,1
220 WAIT 120
300 OUT 3,0
310 GOTO 30
400 '*NIGHT
410 OUT 1,0
420 OUT 2,1
430 WAIT 30
440 OUT 2,0
450 WAIT 30
460 IF ANA(2)<300 THEN GOTO 420
470 RETURN
```

もし暗かったら、400 行のサブルーチンを呼ぶ
その後は 60 行へジャンプ

もどる行を変更

ここから夜間のサブルーチン

車用の青を消す

黄を光らせる

0.5 秒待つ

黄を消す

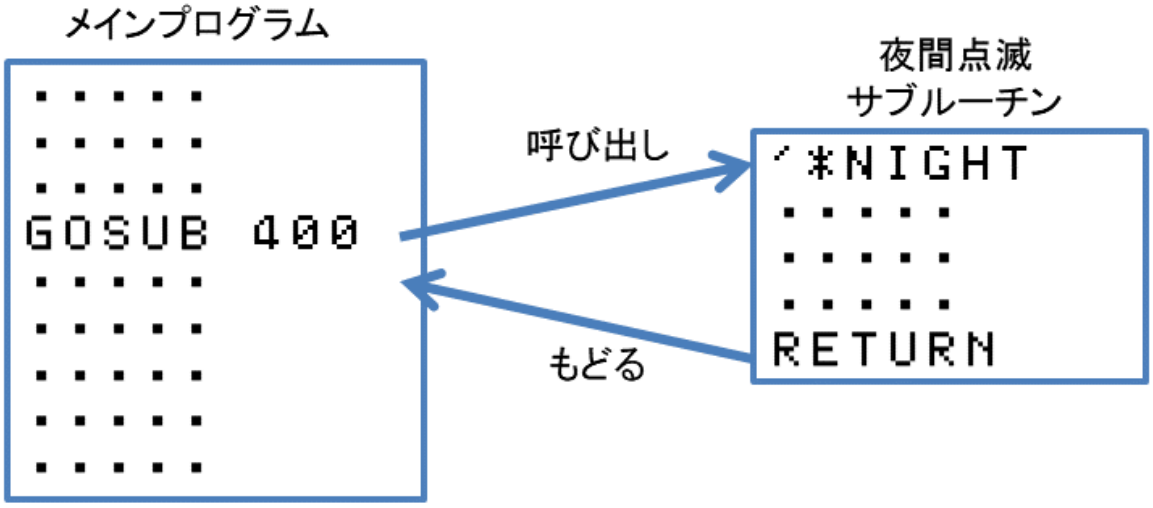
0.5 秒待つ

まだ暗かったら 420 行へジャンプ

メインプログラムへもどる

プログラムを実行してみましょう。
車用の青が光っている時にフォトトランジスターを指で暗くすると、黄色の点滅になります。
明るくすると、元に戻って黄色→赤になります。

47 行では、IF 命令でフォトトランジスタからの入力・ANA(2)の値を判断して、もし 300 以下だったら夜間点滅のサブルーチンを呼んでいます。



メインプログラムからは「GOSUB」命令でサブルーチンへジャンプし、サブルーチンからは「RETURN」(リターン)命令でもどります。もどった後は、GOSUB 命令の続きへプログラムの処理が移ります。
サブルーチンに分けると、プログラムがすっきりしてわかりやすくなります。
また、何度も同じサブルーチンを呼び出して使うことができます。

「SAVE 0」でプログラムを保存しておきましょう。

★できる人は

歩行者用信号が青になる時に、メロディを流してみましょう。
以下の部分を改造します。

```
120 OUT 4,0
130 OUT 5,1
135 LED 0                    歩行者用信号が青の時にメロディを流す
137 PLAY "O4L8 E2E4D4E4ED>B2 <F4FFA4FE
FEDDE2 FFFF4AFEFEDDE2"
140 WAIT 720                 メロディの分だけ待ち時間をのばす
150 FOR I=1 TO 3
160 OUT 5,1
165 PLAY "F4"                青の点滅の時に警告音を流す
170 WAIT 30
180 OUT 5,0
185 PLAY "B4"                青の点滅の時に警告音を流す
190 WAIT 30
200 NEXT
210 OUT 4,1
220 WAIT 120
300 OUT 3,0
310 GOTO 30
```

プログラムを実行してみましょう。
歩行者用信号が青になると、メロディが流れます。

PLAY 命令は、音階を鳴らします。
音階はアルファベットで指定します。(MML、Music Macro Language)

ド	レ	ミ	ファ	ソ	ラ	シ
C	D	E	F	G	A	B

音階の前に「>」を付けると 1 オクターブ下げ、「<」を付けると 1 オクターブ上がります。
音階の後の数字は、音符の長さを表します。(例:E4＝ミの四分音符)
最初に「L8」を指定しているので、数字の指定が無いと八分音符になります。

音階や曲をいろいろ変えてみましょう。

★できる人は

応用プログラムをいくつか紹介します。

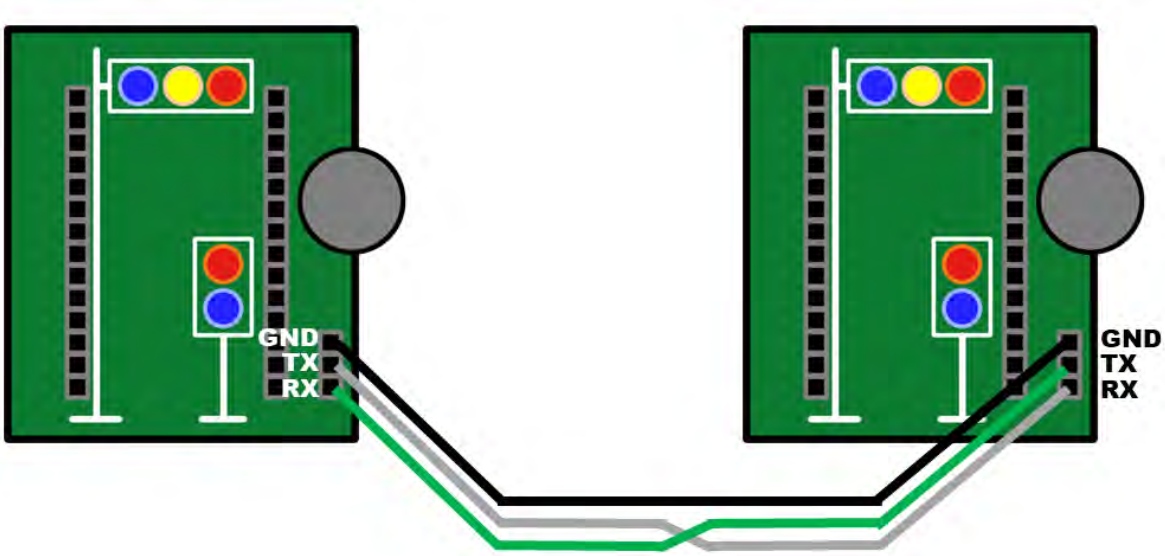
●ルーレット

プログラムを起動して IchigoJam のボタンを押すと、ルーレットが回り、LED1～5 がランダムに点滅します。
ボタンを押すと、ルーレットがゆっくりになって止まります。ボタンを押すとリプレイします。

```
10 '*Roulette
20 CLV
30 OUT 0
40 '@START
50 IF !BTN() CONT
60 IF BTN() CONT
70 '@LOOP
80 OUT 0
90 BEEP 10,2
100 L=RND(5)+1
110 OUT L,1
120 WAIT 6
130 IF !BTN() GOTO 70
140 FOR I=1 TO 5
150 OUT 0
160 BEEP 10,2
170 L=RND(5)+1
180 OUT L,1
190 WAIT 20
200 NEXT
210 GOTO 40
```

●通信対戦じゃんけん

2 台をケーブルでつないで通信対戦します。
基板右下の GND 端子同士を結線、TX と RX 端子はクロス（互い違い）で結線します。



- 2 台の IchigoJam の電源を入れて、プログラムを起動します。
- 車用信号の LED が順番に光ります。青＝グー、黄＝チョキ、赤＝パーを表します。自分が出したい手の所で、IchigoJam のボタンを押します。
- 相手が手を入力するのを待って勝負します。結果は歩行者用信号で表示します。赤＝勝ち、青＝負け、両方とも点くとあいこです。
- ボタンを押すとリプレイします。

```
10 '*Janken
20 CLV:CLK:UART 1,1
30 IF BTN() CONT
40 WAIT 30
50 '@INLOOP1
60 J=J+1:IF J>3 J=1
70 OUT 0:OUT J,1
80 CLT
90 '@INLOOP2
100 IF BTN() BEEP:GOTO 120
110 IF TICK()<60 GOTO 90 ELSE GOTO 50
```

（↓次ページに続く↓）

（↓前ページから続く↓）

```
120 '@READY
130 ?"*"
140 K=INKEY()
150 IF K<>42 GOTO 120
160 CLK
170 '@RLOOP
180 ?J
190 K=INKEY()
200 IF K<49 OR K>51 GOTO 170
210 K=K-48
220 BEEP
230 CLK
240 WAIT 30
250 '@JUDGE
260 W=5
270 IF J=K W=0
280 IF J<K OR (J=3 AND K=1) W=4
290 IF W>0 OUT W,1 ELSE OUT 4,1:OUT 5,1
300 BEEP 10+(W<>4)*20,30
310 WAIT 30
320 IF !BTN() CONT
330 RUN
```

●通信対戦スカッシュ

- 2 台の IchigoJam の電源を入れて、プログラムを起動します。
- どちらからでも IchigoJam のボタンを押して、ボールをサーブできます。
- レシーブ側では、ボールが車用信号の青→黄→赤、歩行者用信号の赤→青と飛んでくるので、歩行者用の青が点いた時に IchigoJam のボタンを押して打ち返してください。
- ボタンを押すのが早すぎても遅すぎてもミスになり、IchigoJam の LED が点いて停止します。ミスした方からサーブして再開してください。

```
10 '*Squash
20 IF BTN()=1 THEN CONT
30 '@RESTART
40 CLV
50 UART 1,1
60 CLK
70 OUT 0
80 W=30
90 '@START
100 IF BTN()=1 THEN GOTO 140
110 K=INKEY()
120 IF K>0 AND K<63 THEN GOTO 180
130 GOTO 90
140 '@GAMELOOP
150 GOSUB 290
160 CLK
170 K=0
180 '@GAMERE
190 IF K=0 OR K>62 THEN K=INKEY():CONT
200 W=K-32
210 K=0
220 GOSUB 370
230 IF M=0 THEN GOTO 140
240 LED 1
250 BEEP 30,60
260 WAIT 60
270 IF BTN()=0 THEN CONT
280 GOTO 30
```

（↓次ページに続く↓）

（↓前ページから続く↓）

```
290 '@SERVE
300 FOR X=5 TO 1 STEP -1
310 OUT X,1
320 WAIT W
330 OUT X,0
340 NEXT
350 PRINT CHR$(W+32)
360 RETURN
370 '@RECEIVE
380 B=0
390 FOR X=1 TO 4
400 OUT X,1
410 WAIT W
420 OUT X,0
430 NEXT
440 IF BTN()=1 THEN GOTO 500
450 OUT 5,1
460 CLT
470 '@WAITLOOP
480 IF BTN()=1 THEN GOTO 540
490 IF TICK()<W THEN GOTO 470
500 '@MISS
510 OUT 5,0
520 M=1
530 RETURN
540 '@STRIKE
550 BEEP
560 OUT 5,0
570 W=W-2
580 RETURN
```