

## スティック形コンピュータを作ろう

### ●今回作る物

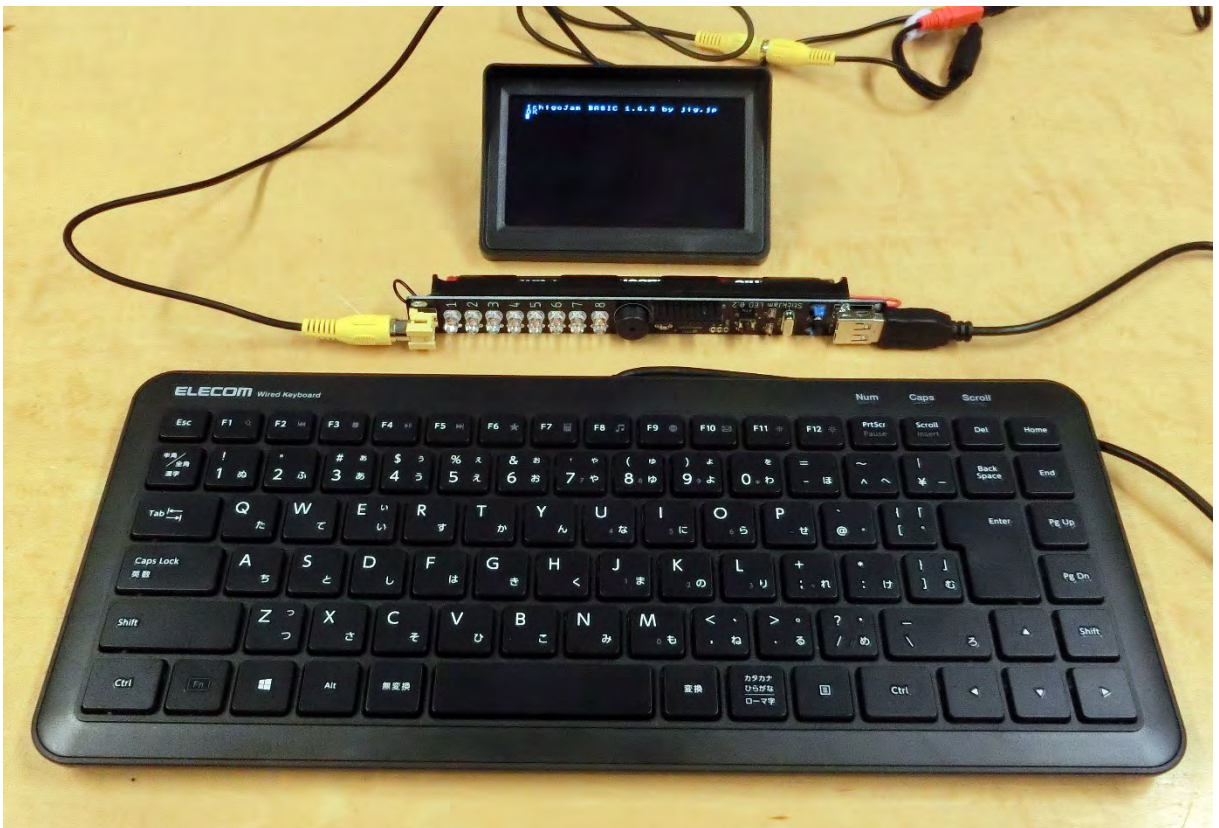
LED を 8 個のせたスティック形の IchigoJam です。



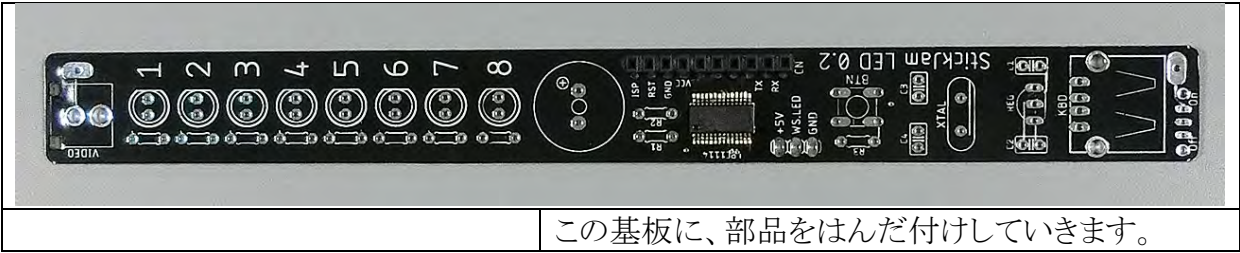
電源は裏面の単 3 電池 3 本です。



キーボードとモニターをつなぐと、普通の IchigoJam と同じようにプログラミングできます。



●はんだ付けの手順



1. 抵抗 R1 470Ω

黄・紫・茶・金

R1 の場所にはんだ付けします。  
向きはありません。

2. 抵抗 R2 100Ω

茶・黒・茶・金

R2 の場所にはんだ付けします。  
向きはありません。

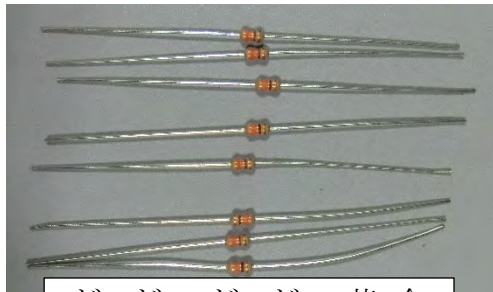
3. 抵抗 R3 100kΩ

茶・黒・黄・金

R3 の場所にはんだ付けします。  
向きはありません。



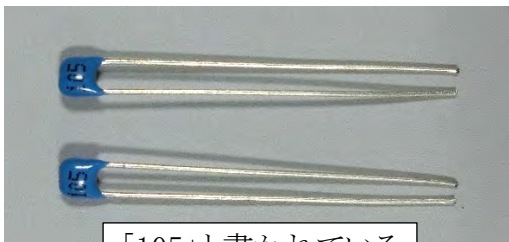
#### 4. 抵抗 330Ω × 8 本



だいたい・だいたい・茶・金

8本ならべてはんだ付けします。  
向きはありません。

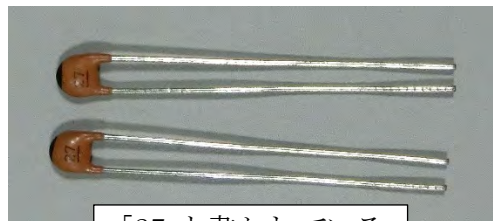
#### 5. セラミックコンデンサ C1,C2 1μF × 2 本



「105」と書かれている

C1,C2 の場所にはんだ付けします。  
向きはありません。

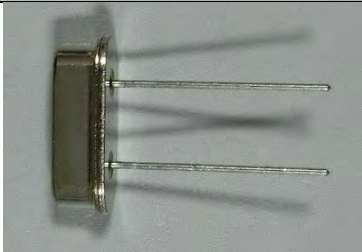
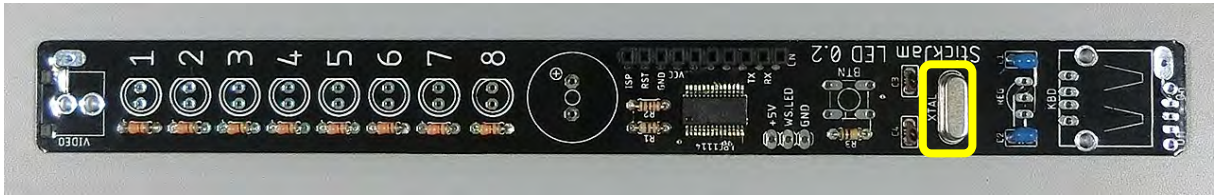
#### 6. セラミックコンデンサ C3,C4 27pF × 2 本



「27」と書かれている

C3,C4 の場所にはんだ付けします。  
向きはありません。

### 7. クリスタル 12MHz



XTAL の場所にはんだ付けします。  
向きはありません。

### 8. タクトスイッチ



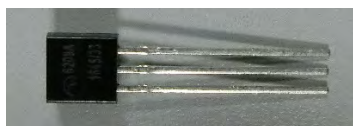
BTN の場所にはんだ付けします。  
4本の足が入る向きにさしてください。

### 9. USB キーボード端子



KBD の場所にはんだ付けします。

### 10. レギュレータ

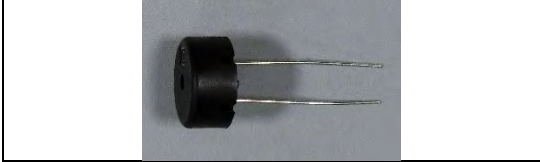
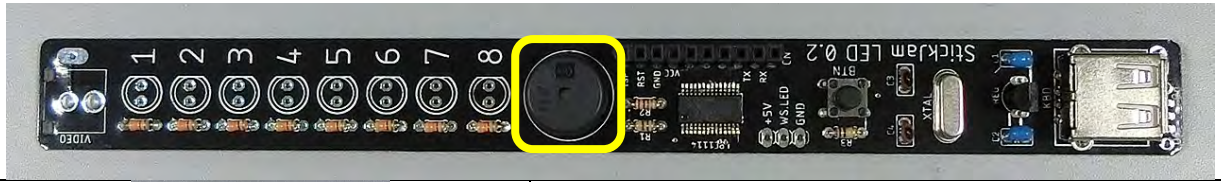


REG の場所にはんだ付けします。  
レギュレータの平らな面と、基板の直線が合う向きにさしてください。





### 11. 圧電ブザー



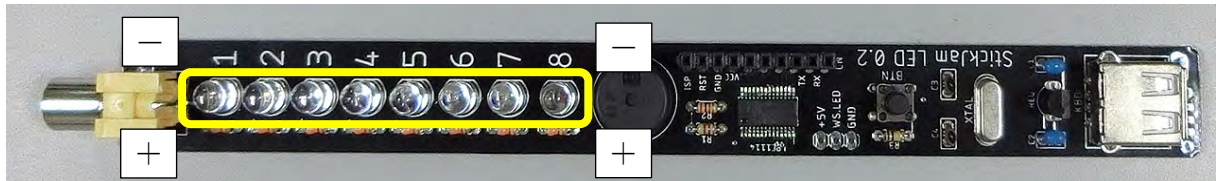
円が描いてある場所にはんだ付けします。  
向きはありません。

### 12. ビデオ端子



VIDEO の場所にはんだ付けします。  
銀色の端子が外へ向くようにはんだ付けします。

### 13. LED × 8 個



数字の 1~8 の場所にはんだ付けします。  
プラス(+)とマイナス(-)の向きがあるので注意してください。

### 14. スライドスイッチ



基板のうらがわにはんだ付けします。  
向きはありません。

## 15. 電池ケース



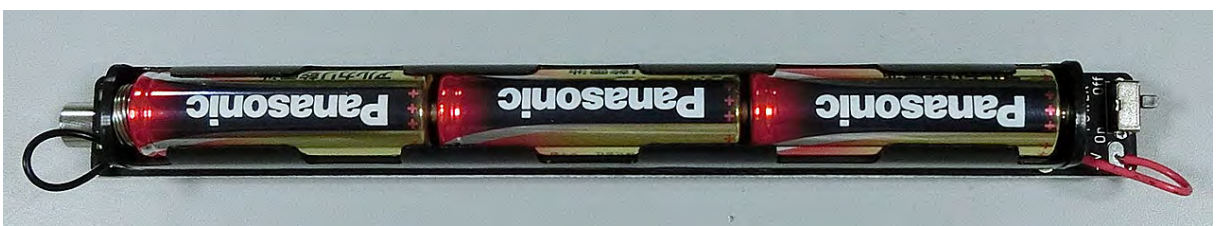
基板のうら面にはんだ付けします。  
赤の線をプラス(+)に、黒の線をマイ  
ナス(GND)へはんだ付けします。



電池ケースのうら面に、両面テープをはじと真ん中の3か所貼り付けます。



電池ケースを、基板のうら面に貼り付けます。



単3電池×3本を、電池ケースに入れます。これで完成です。

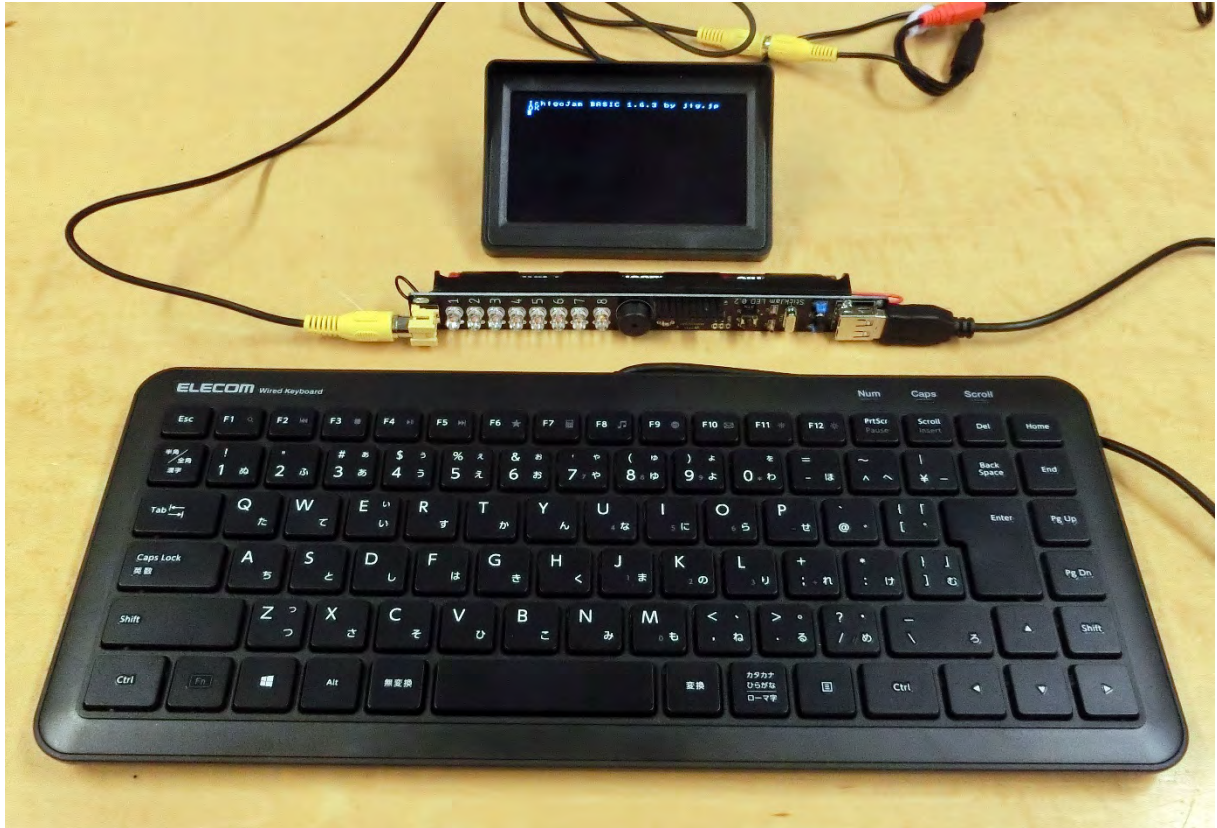


USB 端子にキーボードをつなげます。

ビデオ端子に液晶モニターをつなげます。(モニターの電源をコンセントから取ってください)

スライドスイッチをONにすると、IchigoJamの画面が表示されて、キーボードで文字が打てればOKです。

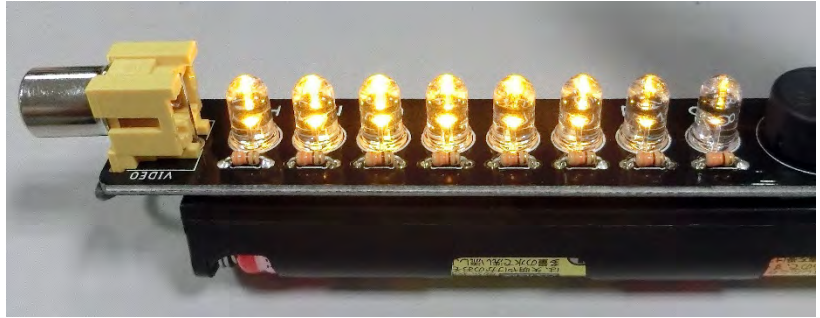
動かなかったら、電源を切って、はんだ付けを確認しましょう。



●LED を光らせる

まずは LED を光らせてみましょう。  
ダイレクトモードで、以下のプログラムを打ちます。

```
OUT 1,1
OUT 2,1
OUT 3,1
OUT 4,1
OUT 5,1
OUT 6,1
OUT 7,1
OUT 8,1
```



8 個の LED が光ります。  
もし光らない LED があったら、電源を切って、LED のはんだ付けを確認しましょう。

```
OUT 1,0
OUT 2,0
OUT 3,0
OUT 4,0
OUT 5,0
OUT 6,0
OUT 7,0
OUT 8,0
```

LED が消えます。

OUT(アウト)命令は、出力ポートに値を出力する命令です。

```
OUT 1 ,1
    ポート  値
    番号
```

ポート番号	出力するポートの番号(1~6)。
値	デジタル出力なので、「0」か「1」を指定する。

LED を光らせるには「OUT (ポート番号), 1」と入力して、OUT ポートに「1」(電圧が ON)を出力します。LED は抵抗を通して GND(電源のマイナス)につながっているので、電流が流れて LED が光ります。

「OUT (ポート番号), 0」と入力して、OUT ポートに「0」を出力すると、GND も電圧が「0」(OFF)なので、電流が流れず、LED は消えます。



●LED を連続で光らせる

LED を連続で光らせるプログラムを作ります。  
 まず、1 番の LED を光らせるプログラムを入力します。

10 P=1	変数 P を 1 にする
20 OUT P, 1	LED を光らせる
30 WAIT 6	10 分の 1 秒待つ
40 OUT P, 0	LED を消す

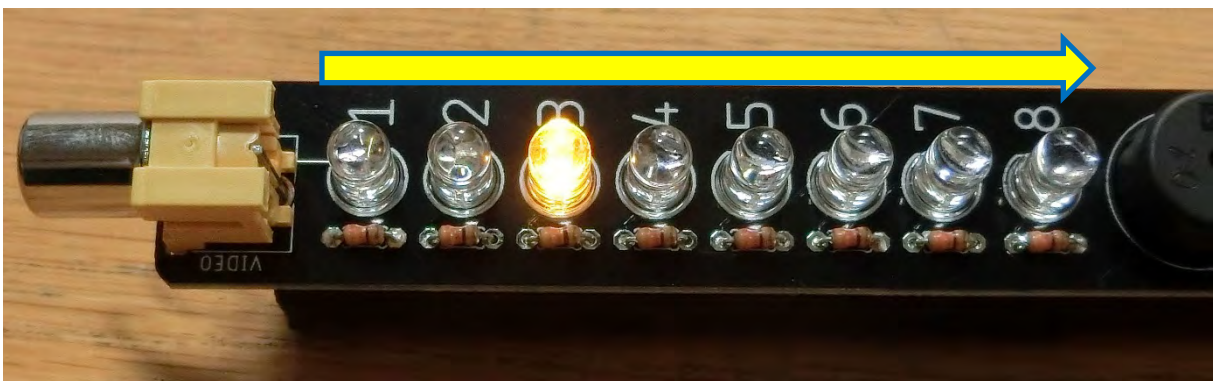
RUN 命令でプログラムを実行してみましょう。1 番の LED が 10 分の 1 秒光ります。  
 WAIT 命令の数字を変えて、光る時間をいろいろ変えてみましょう。

今回は**変数**(へんすう)Pを使って、出力するポートを指定しています。  
 変数は、数字を入れる箱(いれもの)だと思ってください。  
 小学生の人は算数で習う「□」(四角)、中学生以上の人は「x」などの文字と思えばいいです。

さて、同じように 2 番～8 番の LED を光らせるのですが、8 個分のプログラムを入力するのは大変です。  
 変数 P を自動的に 1～8 まで変化させて、**くりかえし**のプログラムにします。  
 ※青い色をつけた、10 行と 50 行だけ入力します。

10 FOR P=1 TO 8	変数 P を 1 から 8 まで、1 ずつ増やす
20 OUT P, 1	
30 WAIT 6	
40 OUT P, 0	
50 NEXT	FOR にもどってくりかえし

RUN 命令でプログラムを実行してみましょう。1 番～8 番の LED が順に光ります。



新しく追加した FOR(フォー)、NEXT(ネクスト) 命令について説明します。

```
FOR   P=1   TO   8   STEP   1
      変数の  最後の値  変化させる値
      最初の値
```

変数の最初の値	指定した変数へ最初にセットする値。
最後の値	この値まで変数を変化させる。
変化させる値	この値ずつ変数を変化させる。 STEP 以下を省略すると1 ずつ増やす。

```
NEXT  ここまでくり返し
```

「FOR P=1 TO 8」とすると、変数 P を1から始めて 8 になるまで1ずつ増やして、「NEXT」までのプログラムをくり返し実行します。

最後に GOTO 命令を追加して、最初にもどってくりかえすようにしましょう。

```
10 FOR P=1 TO 8
20 OUT P,1
30 WAIT 6
40 OUT P,0
50 NEXT
60 GOTO 10
```

10 行にもどる

RUN 命令でプログラムを実行してみましよう。LED が 1 番～8 番までくり返し光ります。

**ここで、プログラムを保存します。**

```
SAVE 1
```

★できる人は

FOR 命令の最初の値、最後の値、変化させる値をいろいろ変えて、LED がどう光るか試してみましよう。



**●8 個の LED を使ったプログラム**

8 個の LED を使ったプログラムをいくつか紹介します。入力して動かしてみてください。

※「NEW」命令を実行するか、StickJam の電源を一度切って、プログラムを消去してから、新しいプログラムを入力してください。

**●ルーレット**

実行すると、8 個の LED がランダムに光って止まります。

StickJam のボタンを押すと、もう一度回ります。

```
10  ' *ROULETTE
20  FOR I=1 TO 10
30  OUT 0
40  P=RND(8)+1
50  OUT P,1
60  WAIT 6
70  NEXT
80  IF !BTN() CONT
90  GOTO 20
```

**●サイモン（記憶ゲーム）**

実行すると、8 個の LED のどれか 1 個が光ります。キーボードでその番号の数字キーを押してください。次は 2 個が順番に光ります。その順番に数字キーを押してください。

1 個ずつ出題が増えていきます。解答をミスするとゲームオーバーです。

```
10  ' *SIMON
20  CLV:OUT 0
30  WAIT 30:M=M+1:[M]=RND(8)+1
50  FOR I=1 TO M
60  N=[I]
70  OUT N,1:WAIT 20:OUT N,0:WAIT 10
80  NEXT
90  A=0
100 K=INKEY():IF !K CONT
110 A=A+1:N=K-48
120 IF N!=[A] OUT 255:END
130 OUT N,1:WAIT 20:OUT N,0
140 IF A<M GOTO 100
150 GOTO 30
```

●バーサライタ (POV)

プログラムを実行すると、8 個の LED が細かく光ります。

StickJam を左右に振ると、眼の残像効果でハート形が見えます。

```

10  ' *POV
20  CLV:OUT 8,0
30  [0]=`00001110
40  [1]=`00011111
50  [2]=`00111111
60  [3]=`01111110
70  [4]=`11111100
80  [5]=`01111110
90  [6]=`00111111
100 [7]=`00011111
110 [8]=`00001110
120 L=9
130 FOR I=0 TO L
140 OUT [I]
150 NEXT
160 GOTO 130
    
```

★30～110 行の 0,1 のパターンが、横にして見るとわかりますがハート形になっています。

(0=消灯、1=点灯) パターンを変えると、見える文字が変わります。

下の表で、0,1 のパターンを考えて、改造してみましょう。

行	30	40	50	60	70	80	90	100	110
データ									



★できる人は:スカッシュゲーム

8 個の LED を連続で光らせるプログラムを改造して、スカッシュゲームにします。

※青い色がついた行を入力します。

```

1 CLV
5 IF !BTN() CONT
6 IF BTN() CONT
7 OUT 0
10 FOR P=1 TO 8
15 IF BTN() M=1
20 OUT P,1
30 WAIT 8-S
40 OUT P,0
50 NEXT
60 IF !BTN() OR M GOTO 140
70 BEEP:S=S+1
80 FOR P=7 TO 1 STEP -1
90 OUT P,1
100 WAIT 8-S
110 OUT P,0
120 NEXT
130 GOTO 10
140 OUT 255:BEEP 30,60:WAIT 60
150 OUT 1<<(S-1):RUN
    
```

変数クリア  
 ボタンが押されるまで待つ  
 ボタンが離されるまで待つ  
 LED 全消灯  
 お手つきチェック  
 スコアが上がるとボールが速くなる  
 お手つき・ミスチェック  
 打ち返し音:スコア+1  
 壁方向へボール移動  
 10行へもどる  
 LED 全点灯:ミス音  
 スコア表示:リプレイ

- プログラムを実行して、StickJam のボタンを押すと、ボールがサーブされて LED1→LED8 が順番に光ります。
- LED8 が光った時にボタンを押すと打ち返せます。(LED8 が光った後にボタンを押し始めて、消えるまで押し続ける) 打ち返すごとにボールが速くなって難しくなります。
- LED8 が消えるまでにボタンを押さなかった場合(遅すぎ)、LED8 が光る前にボタンを押した場合(早すぎ、お手つき)はミスになってゲームオーバーです。LED が8 個全部光った後、打ち返した回数が LED で表示されます。ボタンを押すとリプレイします。

**完成したら、プログラムを保存します。**

SAVE 1

★できる人は:通信対戦スカッシュ

となりの人の StickJam とケーブルでつないで、通信対戦します。

2 台の StickJam で、スカッシュのプログラムを対戦用に改造します。

```

1 CLV
5 IF BTN() OUT 0:GOTO 80
6 K=INKEY():IF K<48 GOTO 5
7 S=K-48:OUT 0
10 FOR P=1 TO 8
15 IF BTN() M=1
20 OUT P,1
30 WAIT 8-S
40 OUT P,0
50 NEXT
60 IF !BTN() OR M GOTO 140
70 BEEP:S=S+1
80 FOR P=7 TO 1 STEP -1
90 OUT P,1
100 WAIT 8-S
110 OUT P,0
120 NEXT
130 ?CHR$(48+S);:GOTO 5
140 OUT 255:BEEP 30,60:WAIT 60
150 RUN
    
```

ボタンが押されたらサーブへ

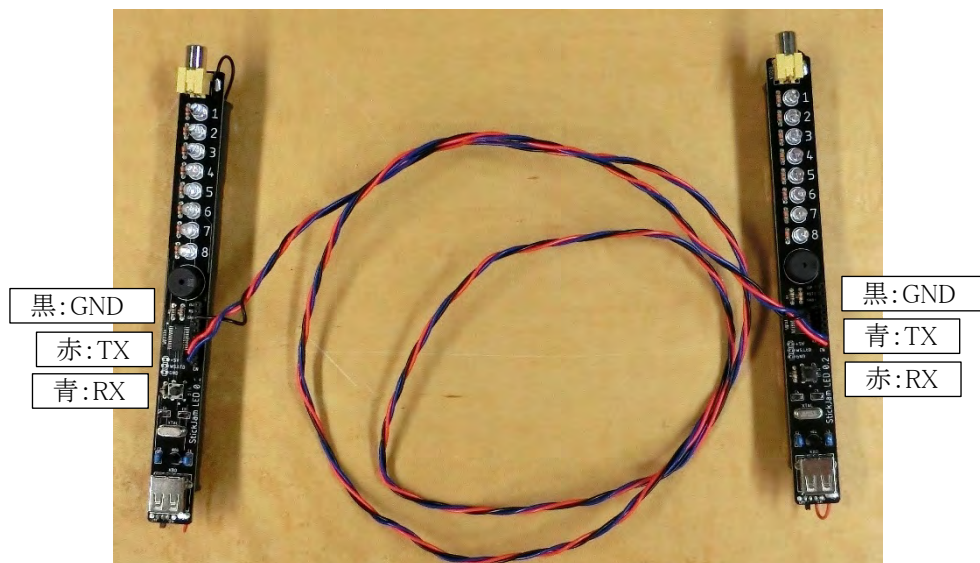
相手からのボールをチェック

ボールの速度設定:LED 全消灯

相手へボールを送る:5行へもどる

最初へもどる

自分となりの人の StickJam を、ケーブルでつなぎます。



黒:GND  
赤:TX  
青:RX

黒:GND  
青:TX  
赤:RX

赤と青の線は、2台で逆につなぎます。



- プログラムを実行して、片方の StickJam でボタンを押すと、ボールがサーブされます。LED7→LED1 が順番に光った後、相手の StickJam へボールが行きます。
- 相手側でボールが LED1→LED8 へ移動します。LED8 が光った時にボタンを押すと打ち返せます。(LED8 が光った後にボタンを押し始めて、消えるまで押し続ける)
- 打ち返したボールが LED7→LED1 へ移動して、相手の StickJam へ行きます。同じように打ち返してください。打ち返すごとにボールが速くなって難しくなります。
- LED8 が消えるまでにボタンを押さなかった場合(遅すぎ)、LED8 が光る前にボタンを押した場合(早すぎ、お手つき)はミスになってゲームオーバーです。LED が8個全部光って止まります。ミスになった方の StickJam でボタンを押すとリプレイします。

★できる人は:プログラムセクタ

このプログラムを入力して、「SAVE 0」でファイル0番に保存します。

StickJam の電源を入れると、このプログラムが自動起動します。

LED1~3 が順番に光るので、実行したい番号の LED が光った時に StickJam のボタンを押すと、ファイル1~3番のプログラムを実行できます。

```
10 @ARUN: ^*SELECTER
20 CLV:OUT 8,0:WAIT 30
30 G=G+1:IF G>3 G=1
40 CLT
50 OUT G,1
60 IF BTN() BEEP:WAIT 30:OUT 0:LRUN G
70 IF TICK()<60 GOTO 50
80 OUT G,0
90 GOTO 30
```