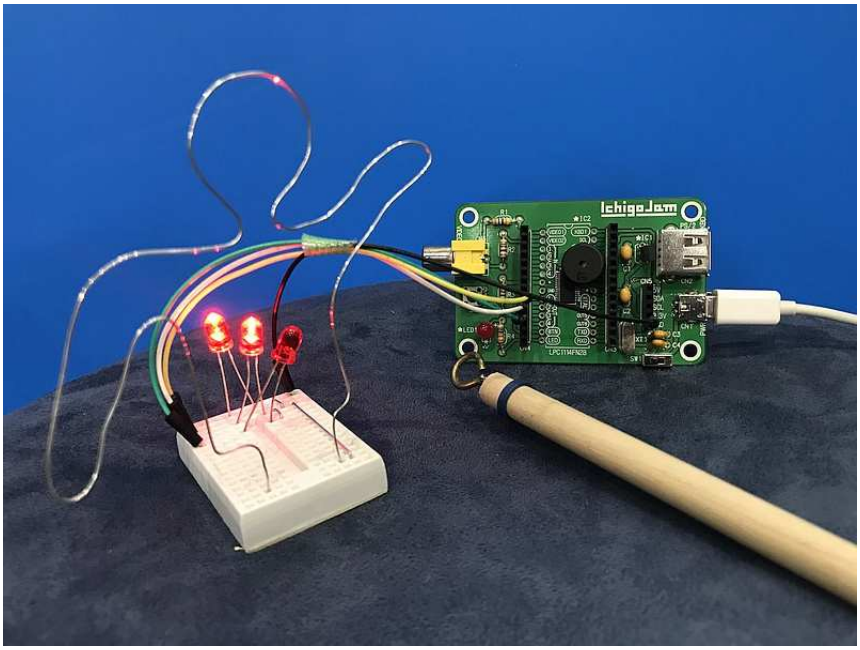
PCN 北九州
プログラミング クラブ ネットワーク キタキュウシュウ

IchigoJam 初級プログラミング ～ゲーム工作編～



【講師】ロボット工房 RoBoCoBo　中野 司 ・ 佐藤 誉夫

【会場】北九州パレス(勤労青少年文化センター) 2F

【日時】2019/3/21(木・祝) 10:00～15:00（昼休み 12:00~13:00）

０．はじめに

◆コース概要

- ねらい

・プログラムを「書く」

・コンピュータ、わかる

・頭、よくなる

・英語になれる

・国産イチゴを守る

- 本日のスケジュール

０．はじめに

１．ブレッドボード実験

２．ワイヤーアートの制作

３．ゲーム回路の配線

４．ゲームプログラミング

５．おわりに

◆スタートアップおさらい

- コマンドとプログラム

- プログラム編集と入力テクニック

  ・スクロールアウトしたら、さようなら...また **ＬＩＳＴ** してね

  ・プログラム行のコピーと削除

  ・もう忘れて... **ＮＥＷ**

- **ＬＩＳＴ** とスクロール停止

  ・F4 キー　→すぐに→　ESC キー

  ・**ＬＩＳＴ－１００** ... ～100 行目まで

  ・**ＬＩＳＴ１００，０** ... 100 行目～

- ファイル表示（**ＦＩＬＥＳ**）と保存（**ＳＡＶＥ　０～３**）

  ・プログラム１行目はコメントにしておく ... **１０　'ﾌﾟﾛｸﾞﾗﾑ ﾒｲ**

◆本体メモリ容量 (1kB ×４バンク)

- セーブする

  SAVE0　　IchigoJam 本体に 0～3 番の４つのプログラムを保存できます

- ロードする

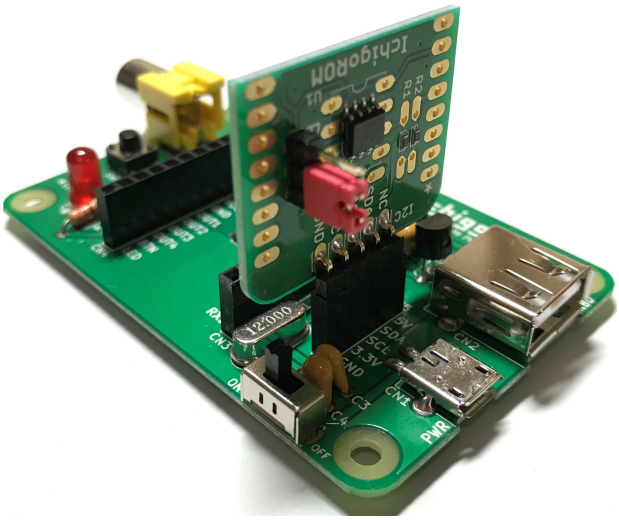
  LOAD0　　読み出すプログラムの番号 0～3 を指定

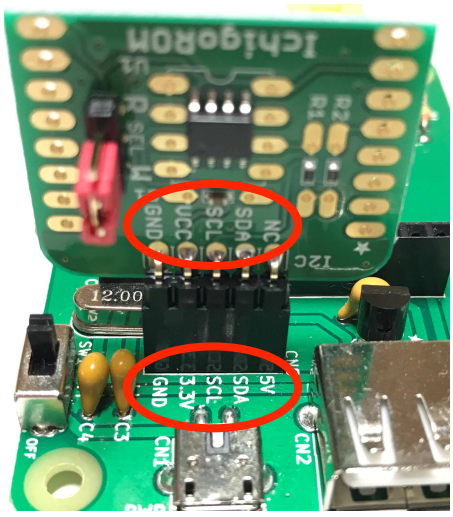
- セーブした番号を見る

  FILES

◆EEPROM (64kB = 1kB ×64 バンク)

- さらに 64 のプログラムを保存できるカード　＊EEPROM を挿し込む向きに注意！





- メモリバング 0～3 を研究用に使おう

- EEPROM バング 100～109 も君たちが使おう

- 全ファイル表示（FILES 0）と保存（SAVE 100～163）

- 今回のファイル表示コマンド FILES129（FILES 0,129 に同じ）

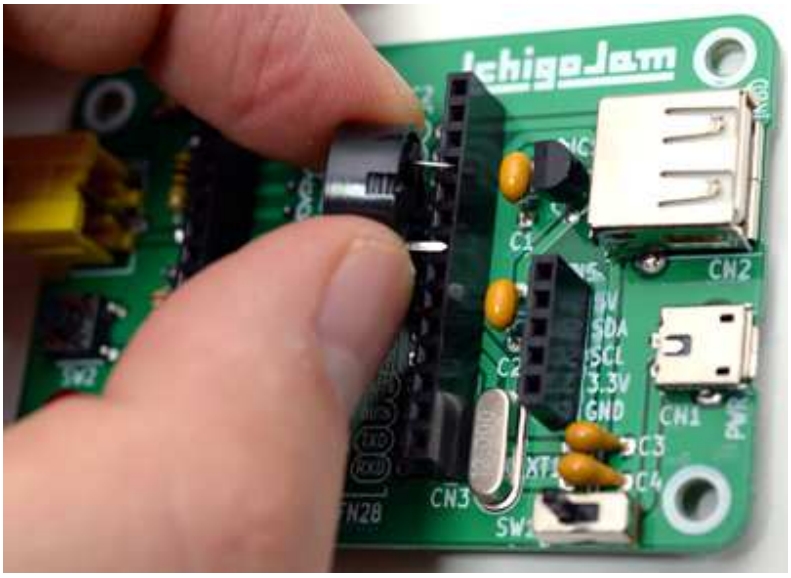
◆圧電サウンダー接続方法

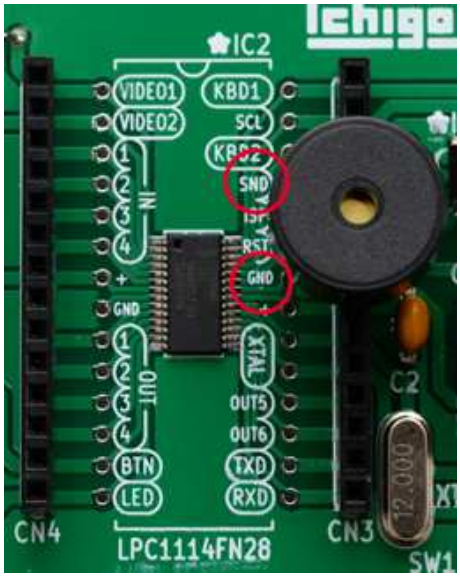
- 圧電サウンダーの両足をそれぞれ次の穴に挿し込んでください

  右列、上から４番目の SND ソケット

  右列、上から７番目の GND ソケット

  ＊極性はありません





◆BEEP 命令

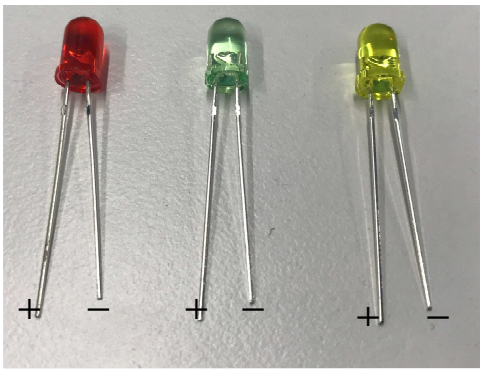
◆PLAY 命令と MML

- なんか音痴…

1．ブレッドボード実験

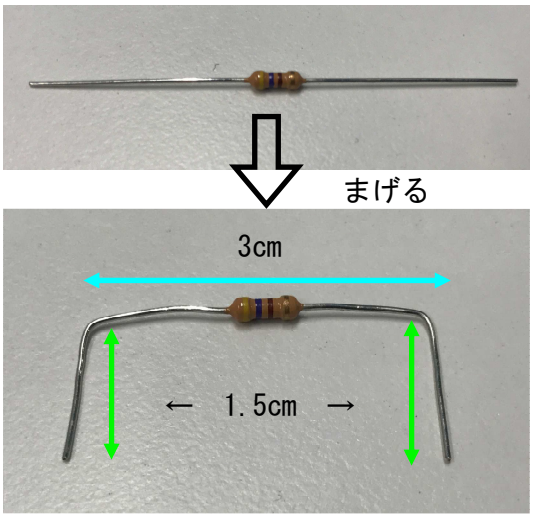
◆電子部品を準備しよう

・LED

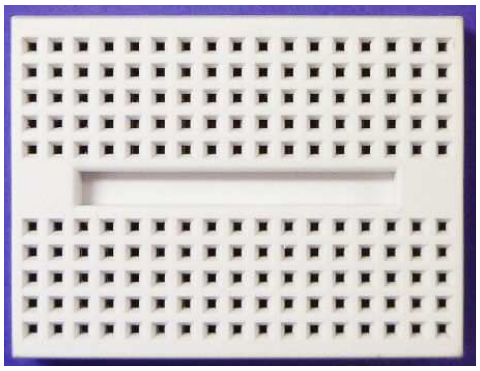


長い方の足（リード）は＋
短い方の足（リード）は－

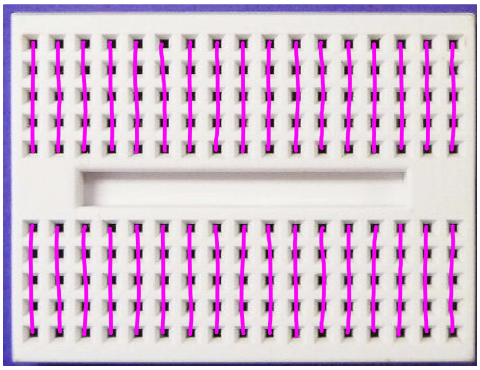
・抵抗器



・ブレッドボード

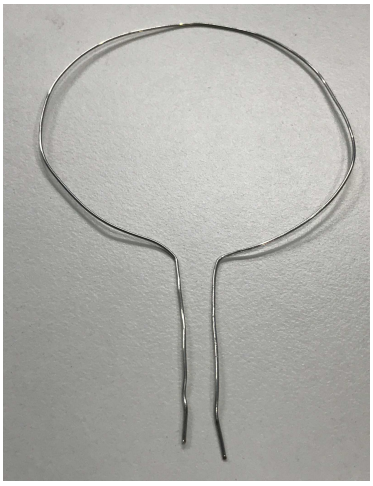


＊内部接続図



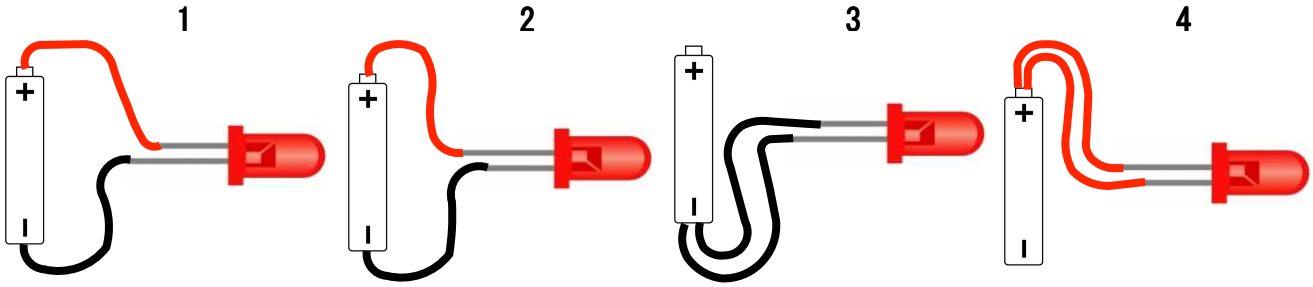
・イライラ棒

・ワイヤーアート

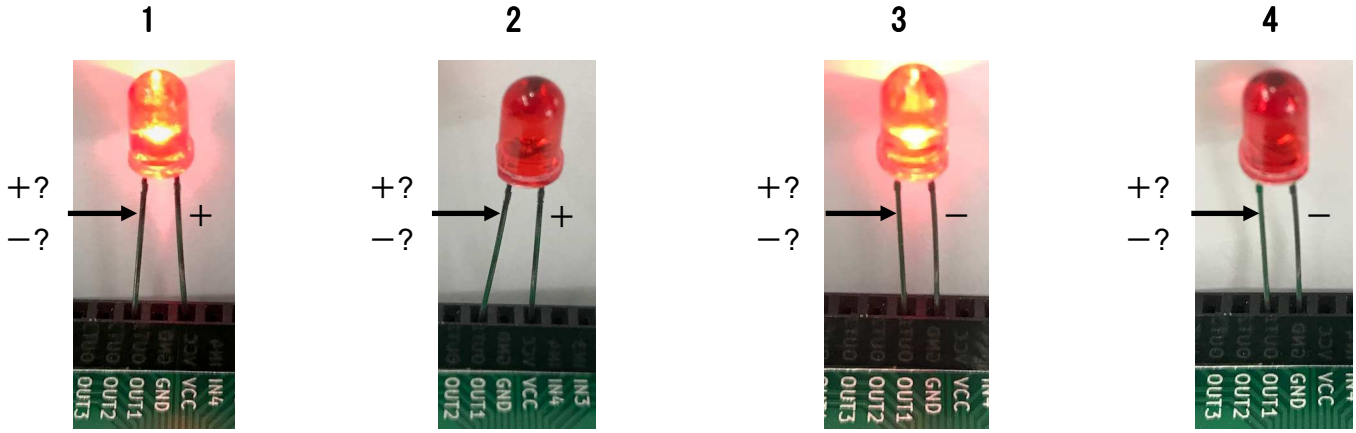


◆LED 点灯クイズ

【第１問】　LED が点灯するのは何番の回路？



【第２問】　LED の左足は　プラス(＋)？　マイナス(－)？



VCC はプラス(＋)、GND はマイナス(－)を表します。

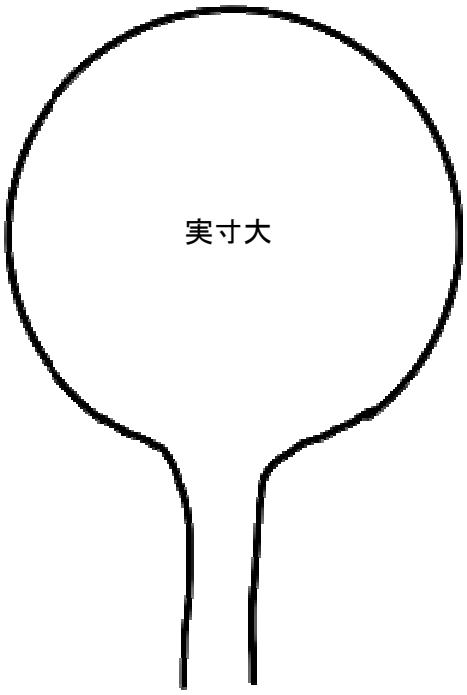
ポート OUT1 はプログラムから ＋ か － を決められます。

◆ワイヤーを曲げて輪っかを作ろう

銀色のワイヤーを曲げて、輪っかを作ります。

ラジオペンチなどを使うときれいに作ることができます。

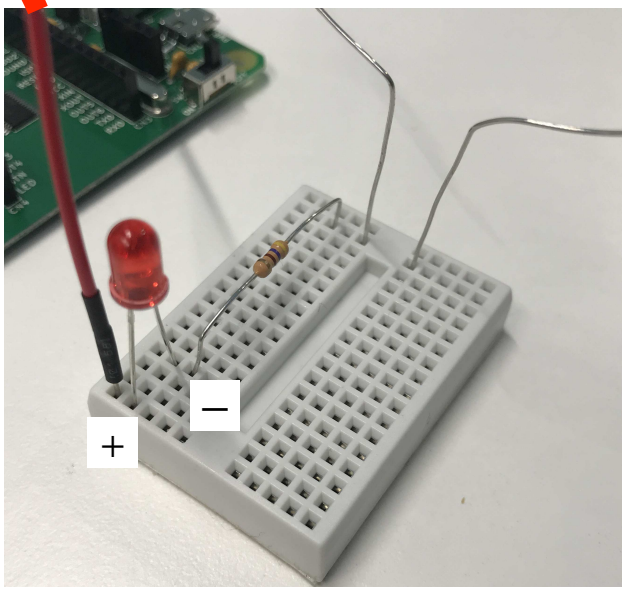
最後に不要な線を切り取ります。

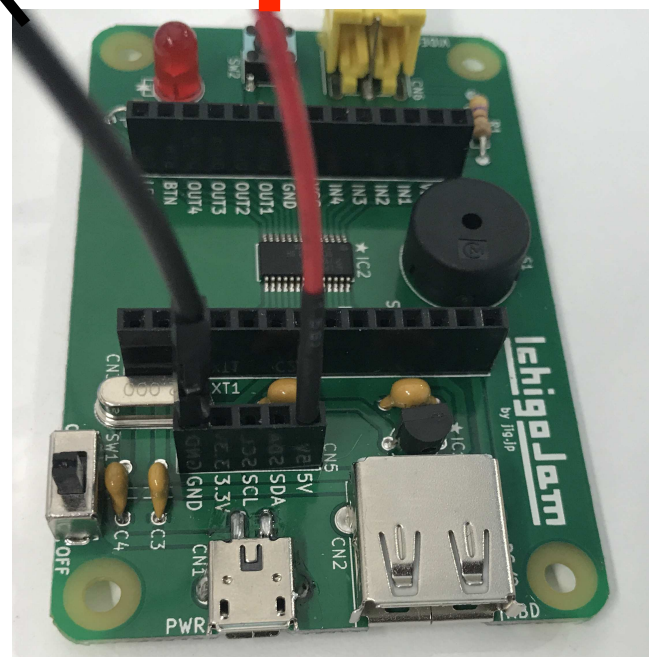


◆かんたんイライラ棒ゲーム回路を組もう

写真を参考に、各部品をブレッドボードに差し込みます。

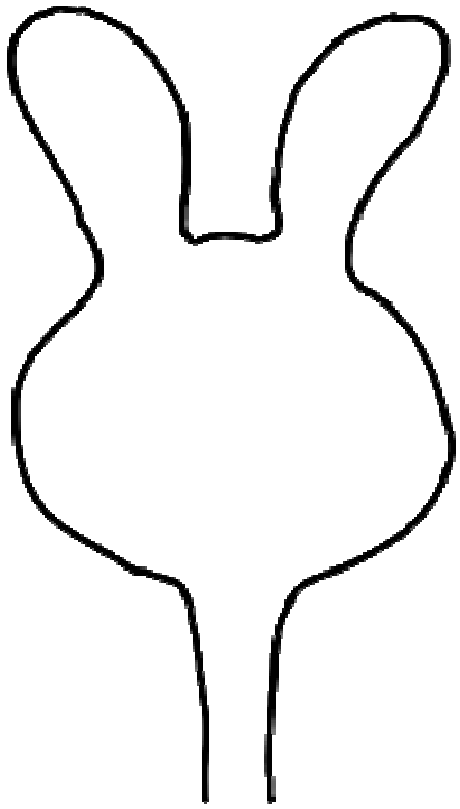
LED の極性（向き）に注意！

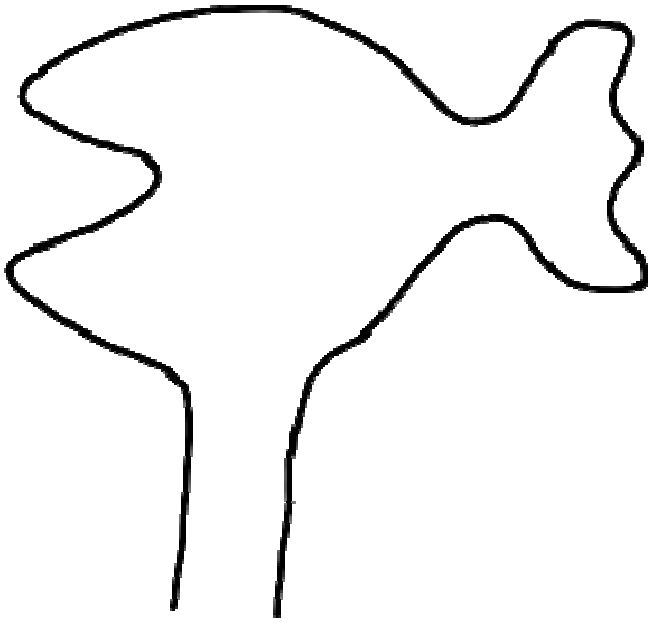


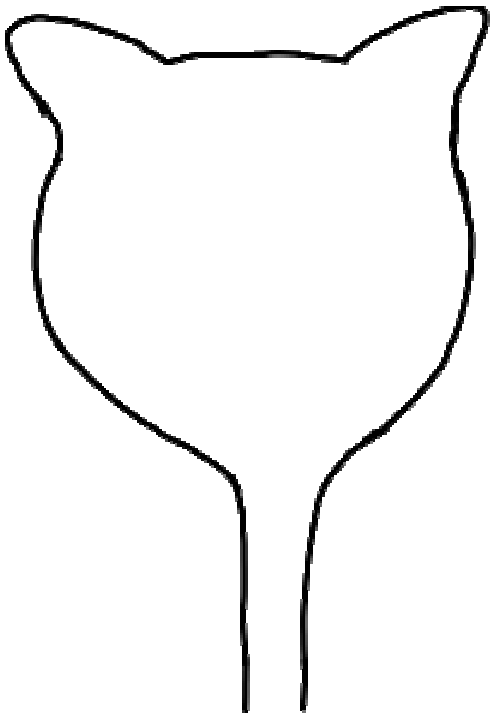


２．ワイヤーアートの制作

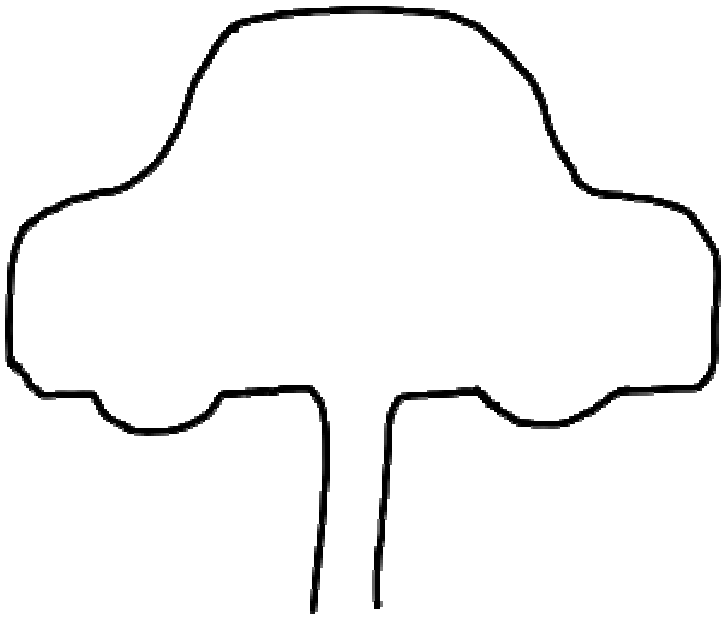
◆ワイヤーアートサンプル図　いきもの編

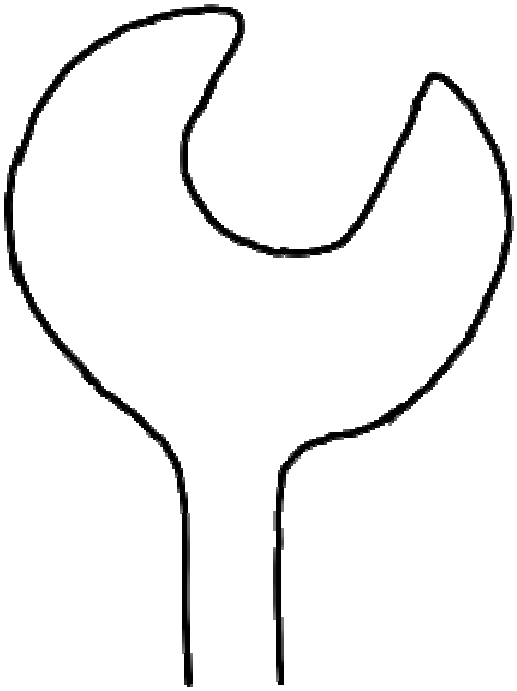






◆ワイヤーアートサンプル図　機械編



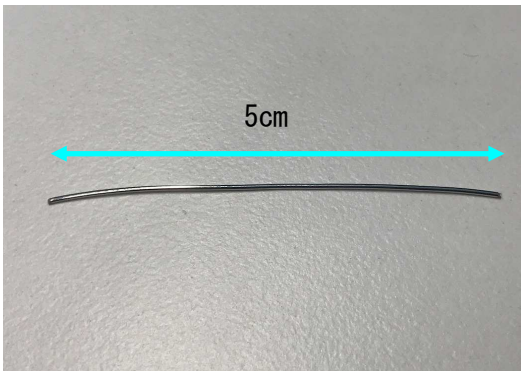


３．ゲーム回路の配線

◆部品を準備しよう

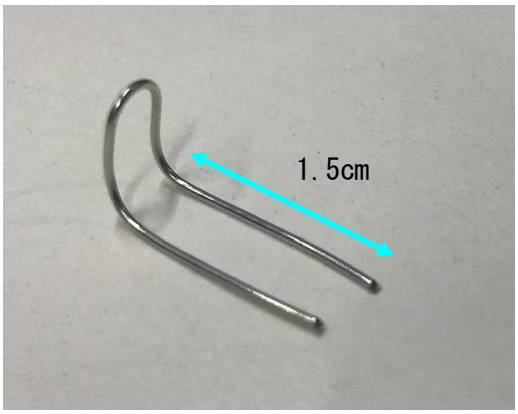
- ゴール接点を作ります

1. ワイヤーを 5cm にカットします

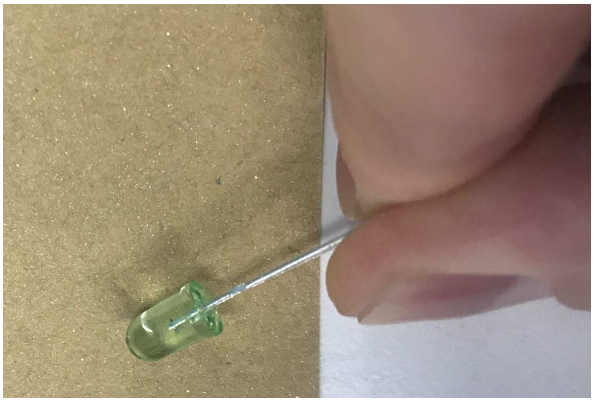


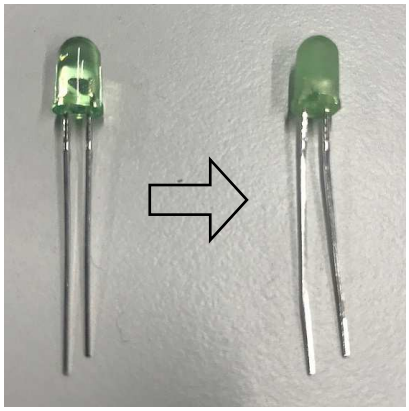
2. 中心を曲げます

3. 端から 1.5cm のところを起こします



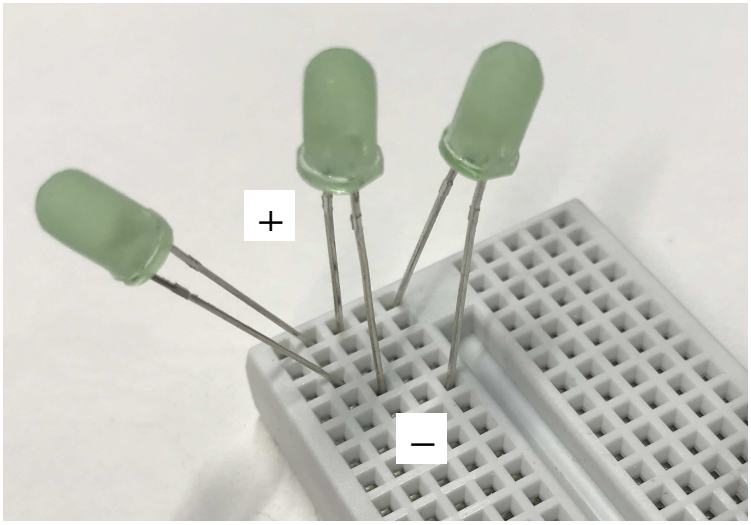
- LED の表面をくもらせて光が拡散するようにします　　＊LED は３個／どの色でも OK



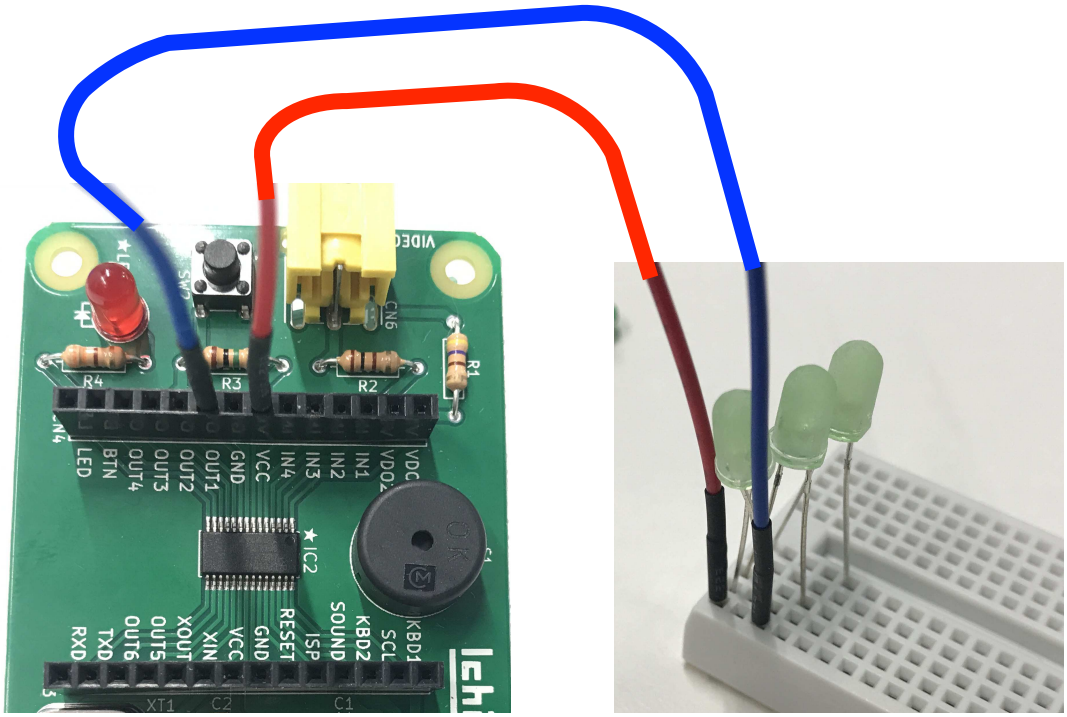


◆電子部品を IchigoJam に接続しよう

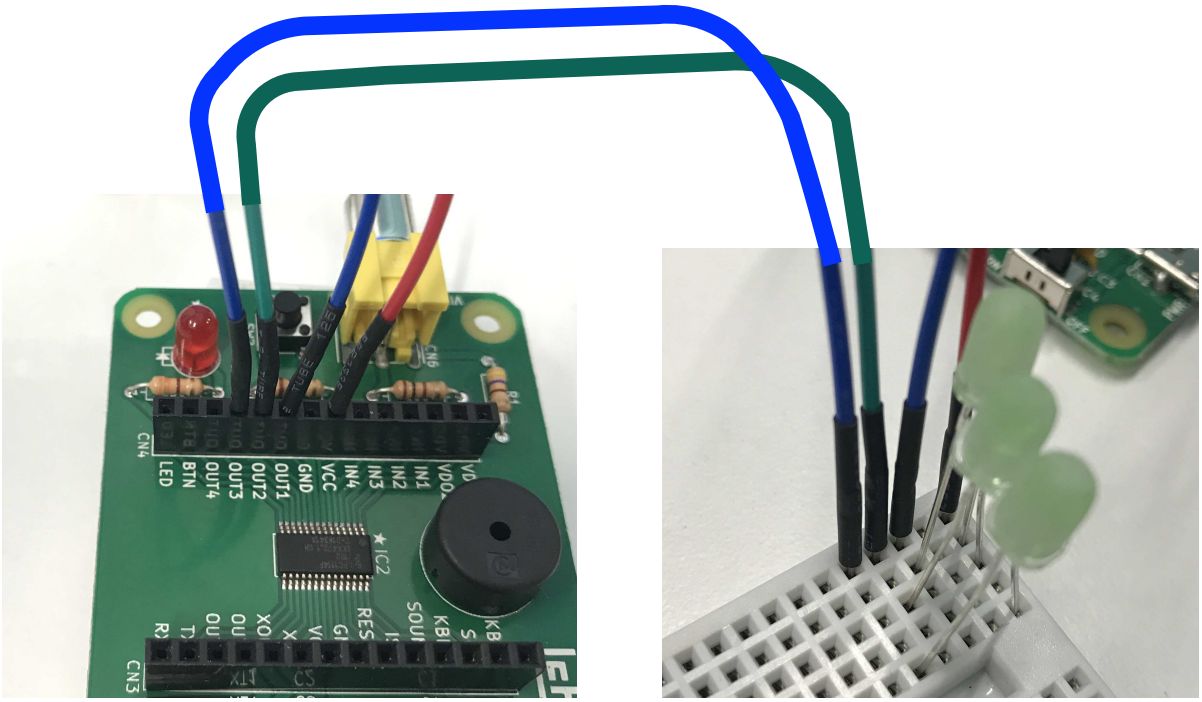
- ブレッドボードに LED を挿し込み、IchigoJam とつなぎます



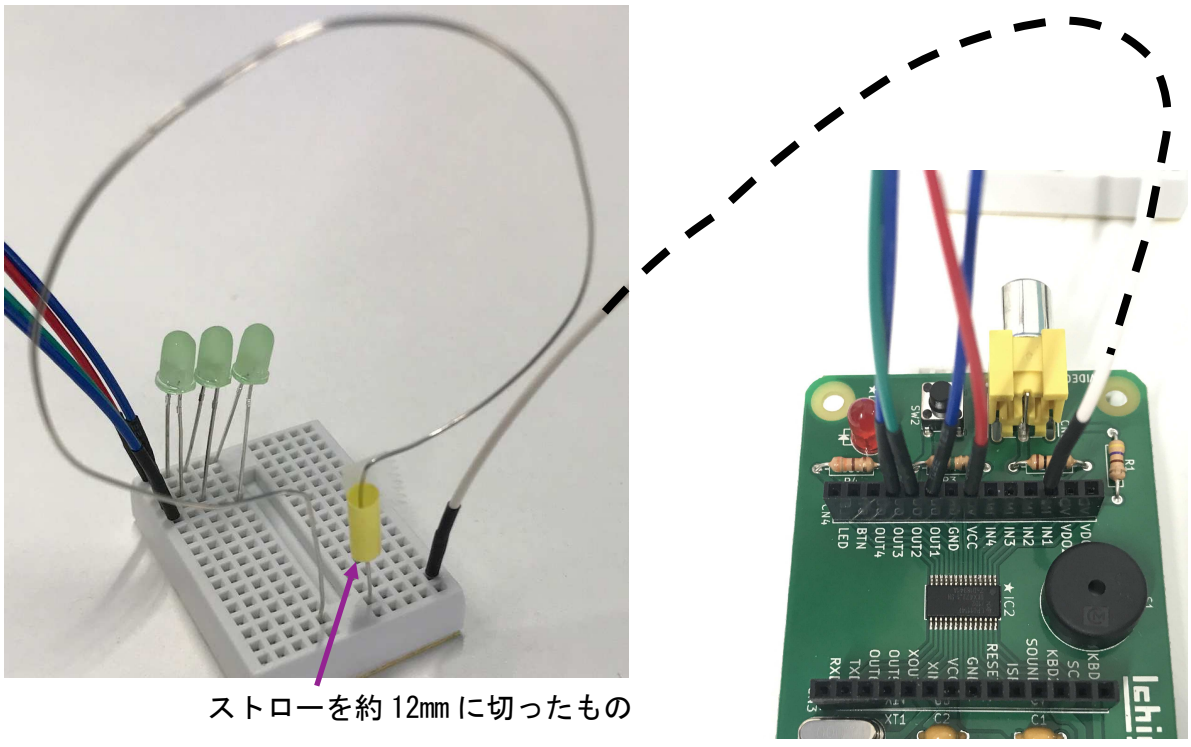
- 赤のジャンパ線を VCC に、青のジャンパ線を OUT1 につなぎます



- 緑のジャンパ線を OUT2 に、青のジャンパ線を OUT3 につなぎます

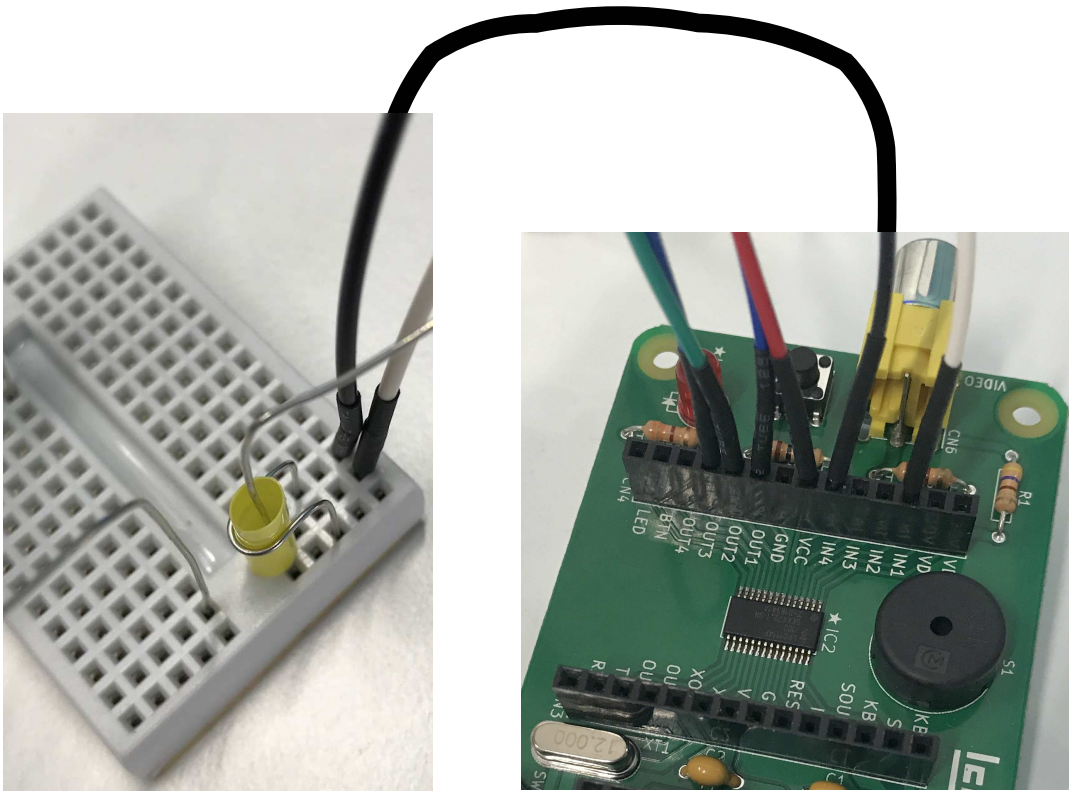


- ワイヤーアートにストローを通し、ブレッドボードに挿し込みます
- 白のジャンパ線を IN1 につなぎます

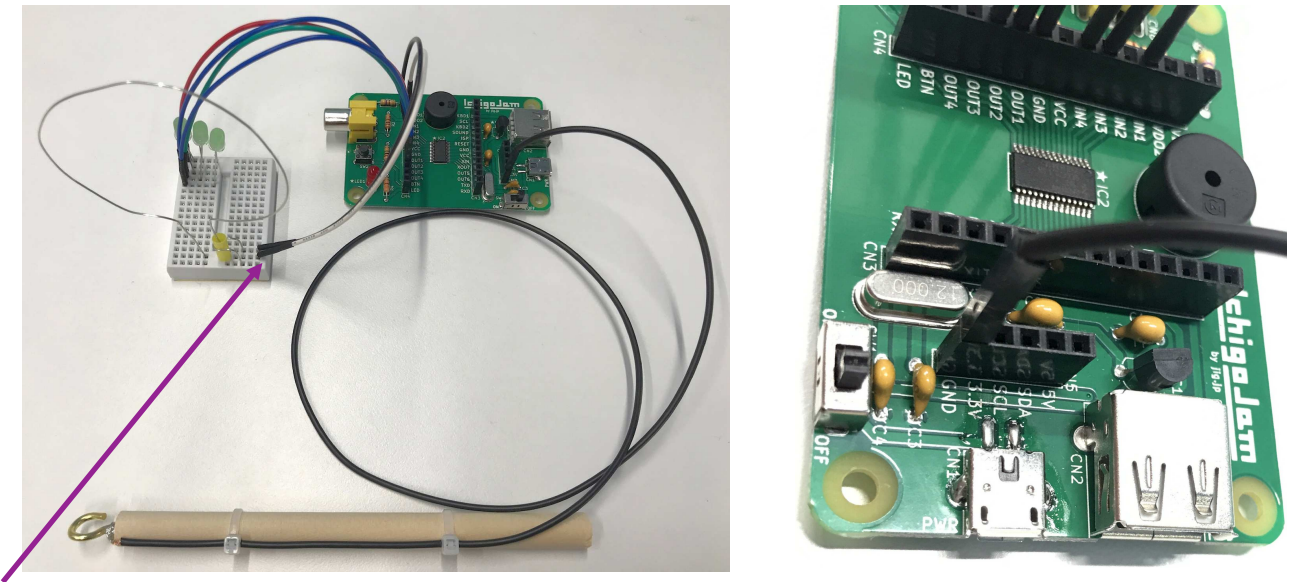


ストローを約 12mm に切ったもの

- ゴール接点をブレッドボードに挿し込みます
- 黒のシャンパ線を IN4 につなぎます



- イライラ棒の黒いコードを GND へ接続します



ワイヤーアートに当たらないよう、配線を寝かせておきましょう

完成！　次はプログラミング！

４．ゲームプログラミング

【プログラム No.0】

◆イライラ棒がワイヤーに当たると LED ピカッ☆

```
1 ’Iraira-Bo 0      コメント行（FILES で表示されるから書いておくと便利！）

10 @START           この行にラベル名“START”をつける
20 LED 0            LED を消す（消しておく）

110 IF IN(1)=0 LED 1
      入力ピン IN1（ワイヤー）が GND(-)レベルなら LED を点ける

330 GOTO @START          10 行に戻る（くり返す）
```

－ＮＥＷ してプログラム入力

－ １行目に ’ プログラム名

－ 行に名前をつける @ラベル名

－ もし… ＩＦ 文

－ もし入力ピン IN1（ワイヤー）が GND(−)レベルなら… ＩＦ ＩＮ（１）＝０

－ 入力ピン IN1（ワイヤー）はプルアップ（何もつながないと VCC(+)レベルに）されている

－ＧＯＴＯ 行番号 ／ ＧＯＴＯ @ラベル名

－ＳＡＶＥ １２０ で保存

【プログラム No.1】

◆LED 点灯（エラー処理）ルーチンを後にまとめる

```
1 ’Iraira-Bo 1

10 @START
20 LED 0

100 @CHECK          この行にラベル名“CHECK”をつける
110 IF IN(1)=0 GOTO @OVER
      入力ピン IN1（ワイヤー）が GND(−)レベルなら 300 行に飛ぶ
180 GOTO @CHECK          100 行に戻る（チェックをくり返す）

300 @OVER           この行にラベル名“OVER”をつける
320 LED 1
330 GOTO @START          10 行まで戻る（また LED を消す）
```

－ ナニモ カワラナイ ネ…

－ これでエラー処理をたくさん書けるゾ！

－ SAVE 121 で保存

【プログラム No.2】

◆ゲームオーバーの文字とサウンドを出す

```
1 ´Iraira-Bo 2

10 @START
20 CLS: LED 0          画面と LED を消す（消しておく）

100 @CHECK
110 IF IN(1)=0 GOTO @OVER
180 GOTO @CHECK

300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"     ブブー
330 WAIT 300: GOTO @START       5 秒後にスタートに戻る
```

- ＣＬＳ で画面クリア

- ＰＬＡＹ 文の ＞ は1オクターブ下、ＣＤＥＦＧＡＢ はドレミファソラシ、Ｒ は休符

- ＷＡＩＴ　６０ で1秒待つ

- ＳＡＶＥ　１２２ で保存

【プログラム No.3】

◆ゴール判定する

```
1 ’Iraira-Bo 3

10 @START
20 CLS: LED 0

100 @CHECK
110 IF IN(1)=0 GOTO @OVER
120 IF IN(4)=0 GOTO @GOAL
```

入力ピン IN4（ゴール接点）が GND(−)レベルなら 400 行に飛ぶ

```
180 GOTO @CHECK

300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
```

ゴール処理の始まり

```
410 ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
```

タラララン♪

```
430 WAIT 300: GOTO @START
```

5 秒後にスタートに戻る

- 入力ピン IN4（ゴール接点）はプルアップ（何もつながないと VCC(+)レベルに）されている

- PLAY 文の < は1オクターブ上、8 は8分音符、長さを指定しなければ4分音符

- SAVE 123 で保存

【プログラム No.4】

◆スタートボタン欲しい！

```
1 ’Iraira-Bo 4

10 @START
20 CLS: LED 0
30 ?"PUSH BUTTON"
40 ?"  TO START"

50 @PUSH    スタートボタン処理の始まり
60 IF BTN()=0 GOTO @PUSH
                    ボタンが押されてなければ 50 行に戻る（チェックをくり返す）
70 CLS: BEEP  ボタンが押されたら、画面を消して、ピッ

100 @CHECK
110 IF IN(1)=0 GOTO @OVER
120 IF IN(4)=0 GOTO @GOAL
180 GOTO @CHECK

300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

－ＢＴＮ（）または ＢＴＮ（０）で本体ボタン、ＢＴＮ（ＵＰ）で↑キー、
ＢＴＮ（ＳＰＡＣＥ）でスペースキーなどをチェックできる

－ＳＡＶＥ １２４ で保存

【プログラム No.5】

◆エラーカウントしてみよう！

```
1 'Iraira-Bo 5

10 @START
20 CLS: LED 0: H=0          エラーカウント用の変数 H を 0 にしておく
30 ?"PUSH BUTTON"
40 ?"  TO START"

50 @PUSH
60 IF BTN()=0 GOTO @PUSH
70 CLS: BEEP

100 @CHECK
110 IF IN(1)=0 GOTO @MISS   ワイヤーに当たったら、カウントアップ処理へ飛ぶ
120 IF IN(4)=0 GOTO @GOAL
180 GOTO @CHECK

200 @MISS                   カウントアップ処理の始まり
240 H=H+1: ?"HIT=";H        カウントアップして表示
260 PLAY ">B"               ブー
270 GOTO @CHECK             100 行に戻る（チェックをくり返す）

300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

－ 1ふやす Ｈ＝Ｈ＋１

－ チャタリングして、あっという間にスコア使い切っちゃうよ～

－ＳＡＶＥ　１２５ で保存

【プログラム No.6】

◆チャタリング対策するナリ

```
1 'Iraira-Bo 6

10 @START
20 CLS: LED 0: H=0
30 ?"PUSH BUTTON"
40 ?"   TO START"

50 @PUSH
60 IF BTN()=0 GOTO @PUSH
70 CLS: BEEP

100 @CHECK
110 IF IN(1)=0 GOTO @MISS
120 IF IN(4)=0 GOTO @GOAL
180 GOTO @CHECK

200 @MISS
240 H=H+1: ?"HIT=";H
260 PLAY ">B"
270 WAIT 30: GOTO @CHECK
```

ワイヤーに当たったら 0.5 秒待つ（この間チェックしない）

```
300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

– ソフトウェア（プログラム）によるチャタリング対策の基本は、少しの間「無視する」

– SAVE 126 で保存

【プログラム No.7】

◆残りライフを LED 点灯数で表わす

```
1 'Iraira-Bo 7

10 @START
20 CLS: OUT 0: H=3          残りライフ H を 3 にしておく
30 ?"PUSH BUTTON"
40 ?"  TO START"

50 @PUSH
60 IF BTN()=0 GOTO @PUSH
70 CLS: BEEP
80 ?"LIFE: ";H

100 @CHECK
110 IF IN(1)=0 GOTO @MISS
120 IF IN(4)=0 GOTO @GOAL
180 GOTO @CHECK

200 @MISS
210 IF H=3 OUT 3,1          ライフ 3→2 で OUT3 を消す
220 IF H=2 OUT 2,1          ライフ 2→1 で OUT2 を消す
230 IF H=1 OUT 1,1          ライフ 1→0 で OUT1 を消す
240 H=H-1: ?"LIFE: ";H          H をカウントダウンして表示
250 IF H=0 GOTO @OVER       もしライフが 0 になったら、ゲームオーバー処理へ飛ぶ
260 PLAY ">B"
270 WAIT 30: GOTO @CHECK

300 @OVER
310 ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

- OUT 0 で出力ピンが全て GND(−)レベルにリセットされる（本体 LED も消える）

- OUT 3，1 で OUT3 が VCC(+)レベルになり、
LED の両極が VCC(+)レベルとなるので、電気が流れず消える

- OUT 0 または OUT 3，0 で OUT3 が GND(−)レベルになり、
LED の VCC(+)レベル極から電気が流れて点く

- SAVE 127 で保存

【プログラム No.8】

◆デカ文字でカッコよくする

```
1 'Iraira-Bo 8

10 @START
20 CLS: OUT 0: VIDEO 3: H=3
30 LC 2,4: ?"PUSH BUTTON"
40 LC 5,6: ?"TO START"

50 @PUSH
60 IF BTN()=0 GOTO @PUSH
70 CLS: BEEP
80 LC 4,4: ?"LIFE: ";H

100 @CHECK
110 IF IN(1)=0 GOTO @MISS
120 IF IN(4)=0 GOTO @GOAL
180 GOTO @CHECK

200 @MISS
210 IF H=3 OUT 3,1
220 IF H=2 OUT 2,1
230 IF H=1 OUT 1,1
240 H=H-1: LC 10,4: ?H
250 IF H=0 GOTO @OVER
260 PLAY ">B"
270 WAIT 30: GOTO @CHECK

300 @OVER
310 LC 3,8: ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 LC 2,8: ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

文字サイズ2倍モード

－ＶＩＤＥＯ　３ で文字をタテ・ヨコとも２倍に

－ＶＩＤＥＯ　１ で元のサイズに

－ＬＣ　２，４ で左から3マス目・上から5マス目に位置決め

－ＳＡＶＥ　１２８ で保存

【プログラム No.9】

◆カウントダウンタイマーでもっとイライラ

```
1 'Iraira-Bo 9

10 @START
20 CLS: OUT 0: VIDEO 3: H=3: T=40
30 LC 2,4: ?"PUSH BUTTON"
40 LC 5,6: ?"TO START"

50 @PUSH
60 IF BTN()=0 GOTO @PUSH
70 CLS: CLT: BEEP
80 LC 4,4: ?"LIFE: ";H
90 LC 4,6: ?"TIME:";DEC$(T,2)

100 @CHECK
110 IF IN(1)=0 GOTO @MISS
120 IF IN(4)=0 GOTO @GOAL
130 IF TICK()<60 GOTO @CHECK
140 CLT: T=T-1
150 LC 9,6: ?DEC$(T,2)
160 IF T=0 GOTO @OVER
170 IF T<11 BEEP
180 GOTO @CHECK

200 @MISS
210 IF H=3 OUT 3,1
220 IF H=2 OUT 2,1
230 IF H=1 OUT 1,1
240 H=H-1: LC 10,4: ?H
250 IF H=0 GOTO @OVER
260 PLAY ">B"
270 WAIT 30: GOTO @CHECK

300 @OVER
310 LC 3,8: ?"GAME OVER"
320 LED 1: PLAY ">B16.R32B"
330 WAIT 300: GOTO @START

400 @GOAL
410 LC 2,8: ?"GOAL! CLEAR!"
420 PLAY "<C8E8G8<C"
430 WAIT 300: GOTO @START
```

40 秒リミット設定

内部タイマーリセット

設定した秒数(40)を表示

1秒たってなければチェックに戻る

1秒たったので、タイマーをリセットし、T を1へらす

残り秒数2けた表示

もし秒数が 0 だったら、ゲームオーバー処理へ飛ぶ

もし秒数が 10 以下だったら、警告音ピッ・ピッ・ピッ…

- タイマー命令 ＣＬＴ でリセット、ＴＩＣＫ（） で時間(1 秒で 60)を返す

- ＤＥＣ＄（数，２） で2けた右そろえ表示

- ＳＡＶＥ　１２９ で保存

５．おわりに

◆次回開催予定

- 2019/6/30(日)．．．場所・内容は「ひ・み・つ．．．」

ありがとうございました．またね！