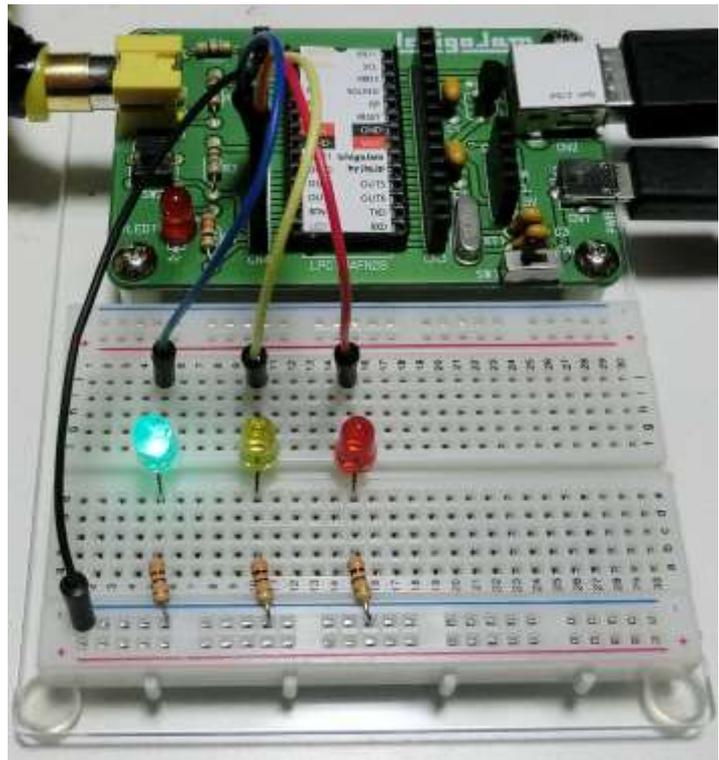


【B-1(3)】IchigoJam で信号機を作ろう (Ver.3)**●今回の目標**

LED 信号機を作って
IchigoJam につないで、
BASIC プログラムで光らせます。



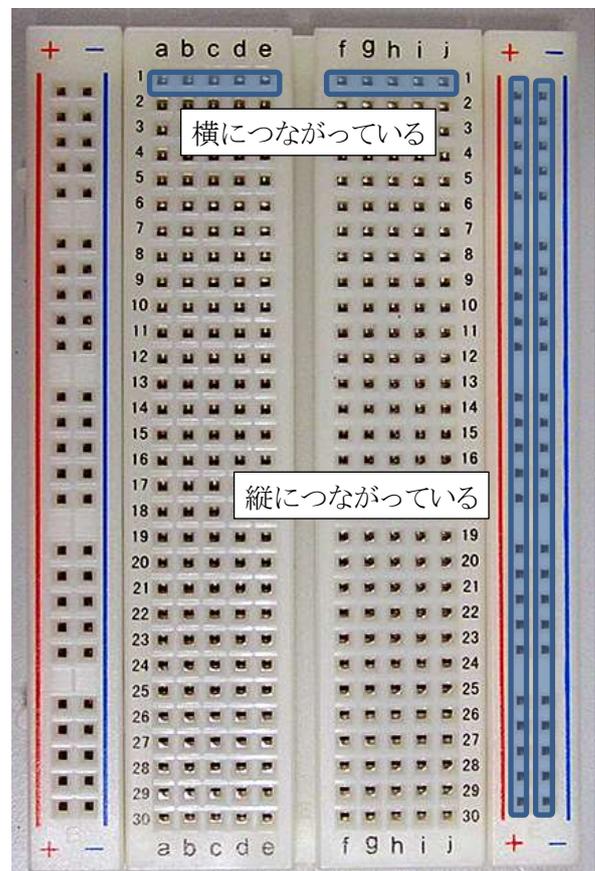
今回は「ブレッドボード」を使って信号機を作ります。

はんだ付けをしなくても電気回路が作れる、便利なボードです。

ボードには、穴がいくつも開いています。
この穴に、部品をさしていきます。

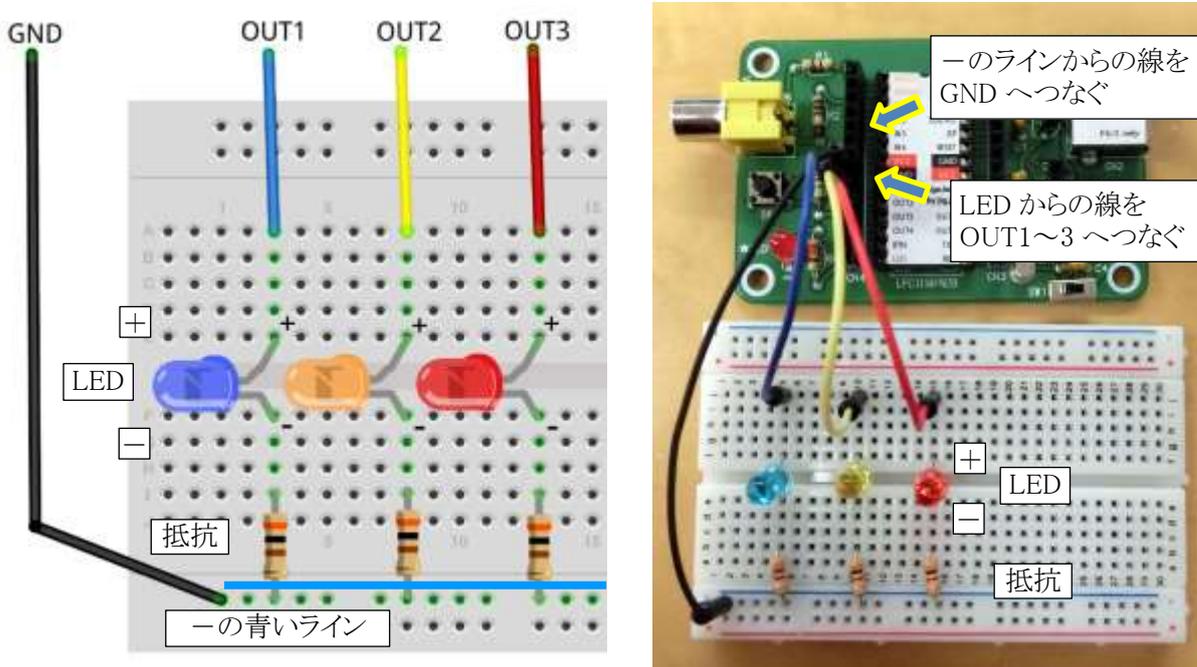
中央の「abcde」「fghij」の穴は、ボードの中で横につながっています。

左右にある「+」「-」の穴は、ボードの中で縦につながっています。

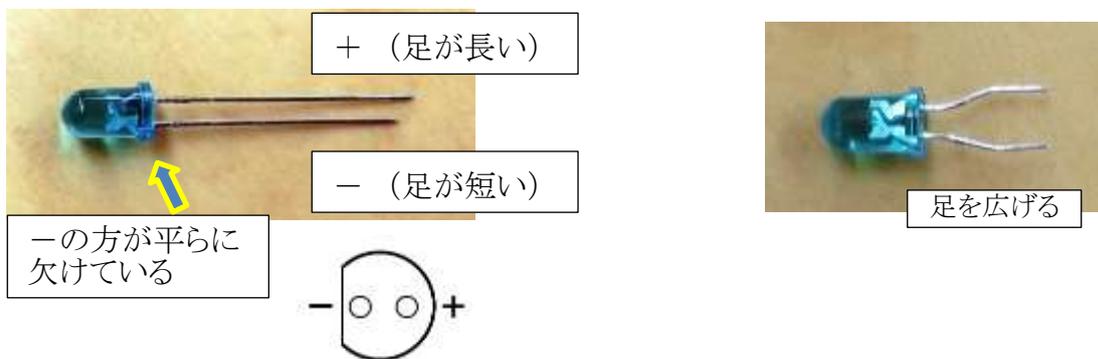


●ブレッドボードで信号機を作る

ブレッドボードに LED と抵抗を差して、IchigoJam とワイヤでつなぎます。



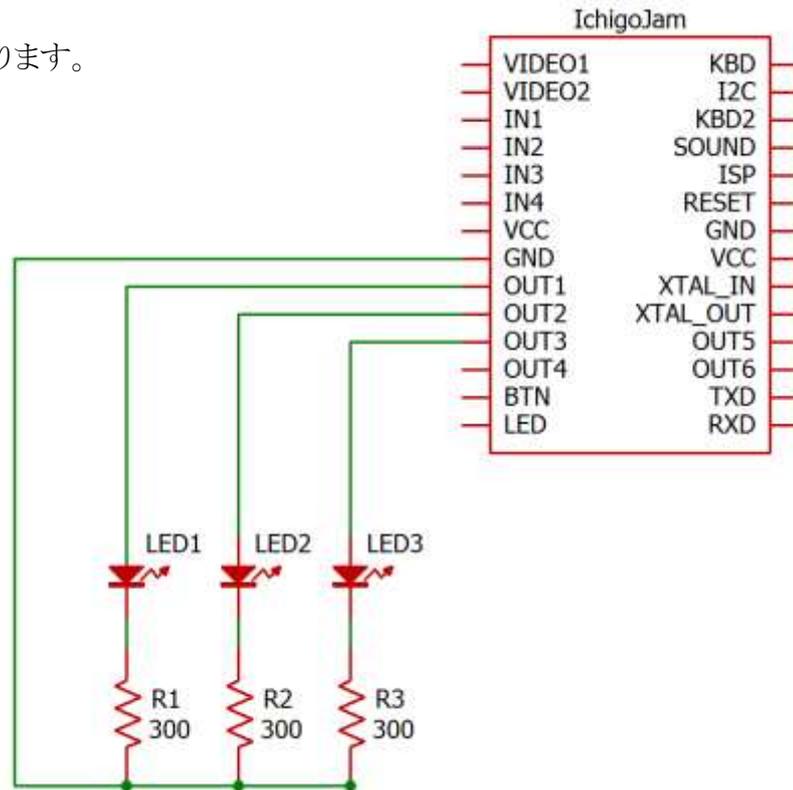
LED は、電流を流すと光る部品です。
端子は、「+」と「-」があります。向きを間違えると光らないので、注意してください。
ブレッドボードにさすために、足をみじかく切って広げます。



抵抗は、電流を流れにくくする部品です。LED に流れる電流を少なくするために入れます。
向きはどちら向きでもいいです。
ブレッドボードにさすために、足をみじかく切って曲げます。



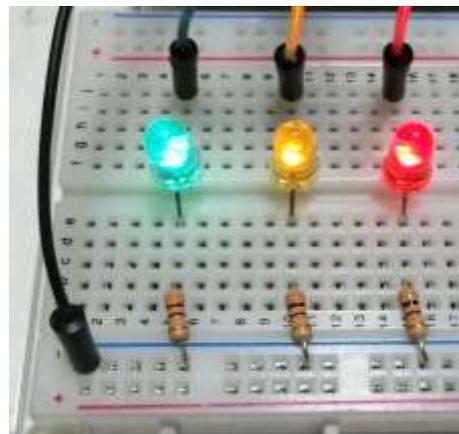
回路図はこのようになります。



回路ができたなら、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
```

信号機の青・黄・赤のLEDが光ります。
もし光らなかったら、部品や線の
さし方をまちがえています。
よく見直しましょう。

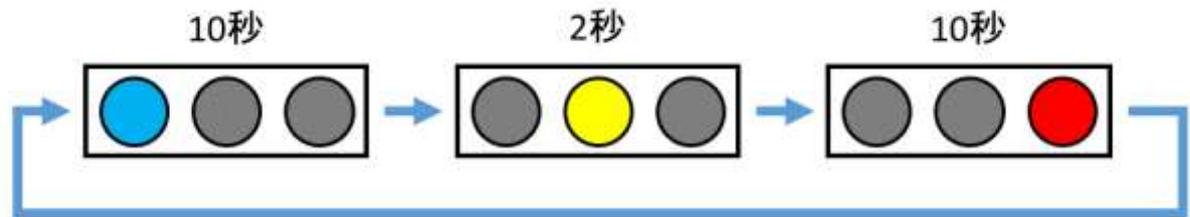


```
OUT 1,0
OUT 2,0
OUT 3,0
```

3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。

では、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ならせます。

10	OUT	1, 1	青 LED を光らせる
20	WAIT	600	10 秒待つ
30	OUT	1, 0	青 LED を消す
40	OUT	2, 1	黄 LED を光らせる
50	WAIT	120	2 秒待つ
60	OUT	2, 0	黄 LED を消す
70	OUT	3, 1	赤 LED を光らせる
80	WAIT	600	10 秒待つ
90	OUT	3, 0	赤 LED を消す
100	GOTO	10	10 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。

ためしてみよう

1. **WAIT** (ウェイト) 命令の数値を変えて、それぞれの LED の点灯時間を変えてみよう。
2. ふつうの「青→黄→赤→青→…」とはちがうパターンで、信号機を光らせてみよう。

「**SAVE 0**」でプログラムを保存しておきましょう。

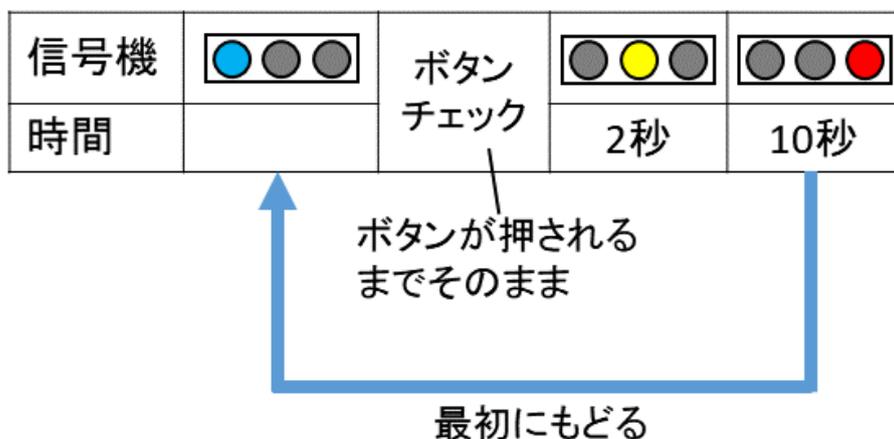
●押しボタン信号にする

プログラムを改造して、「ボタンを押したら信号が黄→赤になる」信号にしてみましょう。

ボタンは、IchigoJam の左側に付いているボタンを使います。



信号機の動きを図で考えます。



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。

プログラムを改造します。

信号機を青にした後の、20行を書きかえます。

```

10 OUT 1,1
20 IF BTN(<)<math>=0</math> THEN GOTO 20
30 OUT 1,0
40 OUT 2,1
50 WAIT 120
60 OUT 2,0
70 OUT 3,1
80 WAIT 600
90 OUT 3,0
100 GOTO 10
    
```

もしボタンが押されていないかったら、この行をくりかえし

「**RUN**」でプログラムを実行してみましょう。
 信号機が青になって、そのまま動きません。
 IchigoJam のボタンを押すと、→黄→赤…と動きます。

「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  20  ELSE      ~
      条件式          条件が成り立つ      条件が成り立た
                        時に実行          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 20 行へもどっています。

ためしてみよう

ボタンを押してすぐ信号が黄色になると、青の時間が短くなるので、車の人は困ります。少し待ってから黄色になるように、プログラムを変えてみよう。

「SAVE 0」でプログラムを保存しておきましょう。