## LED テープを光らせよう



今回は IchigoJam に LED テープ(WS2812B) をつないで、光らせてみます。 クリスマスのイルミネーションなどに使えます。



●LED テープをつなぐ

IchigoJam に LED テープの線をつなぎます。





- ▶ 赤の線→VCC 端子
- ▶ 黒の線→GND 端子
- ▶ 白の線→LED 端子

## ●LED を 1 個光らせる

まずは LED を 1 個光らせてみましょう。 以下のプログラムを入力します。

LET [0],50,0,0:WS.LED 1

1個目のLED が緑色に光ります。

LET [0],0,50,0:WS.LED 1

1個目のLEDが赤色に光ります。

LET [0],0,0,50:WS.LED 1

1個目の LED が青色に光ります。

LET [0],0,0,0:WS.LED 1

LED が消えます。

色の数字は0~255の間で設定します。

(あまり大きくするとまぶしいので、大きくても 100 くらいまでにした方がいいです) 3 個の数字をいろいろ変えて、LED がどう光るか試してみましょう。

#### ★配列変数の値

LET(レット)命令で、配列変数に値をセットします。

LET	[0]	,50	,0	,0
	先頭の	1個目	2個目	3個目
	配列変数	の値	の値	の値

先頭の配列変数	値をセットする先頭の配列変数
値	セットしたい値を1個目,2個目,…とコンマ区切りで書く

「LET [0],50,0,0」とすると、配列変数[0]に50、[1]に0、[2]に0がセットされます。

今回は配列変数に、LED テープを光らせるための値を設定します。 緑色の値・赤色の値・青色の値の3個1組で指定します。 [0]・[1]・[2]が LED1 個目、[3]・[4]・[5]が LED2 個目…の値になります。



配列変数

3つの色の組み合わせでどんな色になるかは、「光の3原色」の規則で決まります。



#### <u>★LED の点灯</u>

WS.LED (ダブリューエス・エルイーディー)命令は、配列変数の値を使って LED テープを光 らせます。

**WS.LED 1 ,30** LED 個数 くりかえし回数

D 個数 くりかえし回数

LED 個数	光らせたい LED の個数を指定する。
	※これは LED の個数なので、配列変数はこの3 倍の個数が必要。
くりかえし回数	LED 個数で指定したパターンを、ここで指定した回数だけくりかえす。

●LED をたくさん光らせる

LED を1個だけでなく、何個も光らせてみましょう。

LET	[0],5	0,0,0	1 個目の LED を緑にする
LET	[3],0	,50,0	2 個目の LED を赤にする
LET	[6],0	1,0,50	3個目の LED を青にする
WS.L	ED 3	LED を 3 個光	らせる

3個の LED が、指定した色で光ります。

LED を光らせる個数を増やしたり、配列変数へ設定する値を変えて色を変えたり、いろいろ 試してみましょう。

## ●LED を同じ色で 30 個光らせる

WS.LED 命令のくりかえし機能を使って、LED を 30 個光らせてみましょう。

LET [0],50,0,0:WS.LED 1,30

30 個の LED が緑色に光ります。

LET [0],0,50,0:WS.LED 1,30

30 個の LED が赤色に光ります。

30 個の LED が青色に光ります。

#### ●LED をいろいろな色で 30 個光らせる

LED が 30 個光っても、全部同じ色ではおもしろくありません。 行番号を付けたプログラムで、いろいろな色で光らせてみましょう。



入力できたら、**RUN** (またはF5キ ー)で、プログラムを実行してみまし ょう。 30 個の LED がいろいろな色で光り

ます。



FOR 命令、NEXT 命令は、くりかえしの命令です。

FOR	I = 0	то	87	STEP	3
	変数の		変数の		変数の
	初期値		終値		増分

変数の初期値	ループ変数の最初の値
変数の終値	ループ変数の最後の値
変数の増分	ループ変数をどのくらい変化させるかの値
	STEP 以下を省略すると1ずつ増やす

# **NEXT** ここまでくりかえす

今回は、ループ変数 I が 0,3,6,9…87 まで 3 ずつ増えて変化します。 その結果、20 行の LET 命令で、配列変数[0]~[89]までランダムな値が設定されます。 RND 関数は、乱数(何が出てくるかわからない数)を出す関数です。

#### RND( 50) 乱数の範囲

乱数の範囲	0~(この数-1)までの乱数が出る
	例えば RND(10)なら、0~9 のうちのどれかの数が出る

RND 関数の乱数の範囲(かっこの中の数)を変えると、LED の色が変わります。 いろいろ変えて試してみましょう。

●30 個の LED をいろいろな色でくりかえし光らせる

1回光って終わってしまうだけではおもしろくありません。
 30個のLEDを、いろいろな色でくり返し光らせてみましょう。
 プログラムを2行追加します。



入力できたら、**RUN**(または F5 キー)で、プログラムを実行してみましょう。 30 個の LED がいろいろな色で光って、ずっとくりかえします。 プログラムを止める時は、キーボード左上の ESC キーを押しましょう。

WAIT (ウェイト)命令は、時間待ちをする命令です。

## WAIT 30

待ち時間

 (待ち時間 60 分の1 秒単位で指定する。「60」で1 秒。

 WAIT の待ち時間を変えると、LED のくりかえし速度が変わります。

GOTO (ゴートゥー) 命令は、ほかの行へジャンプする命令です。

 GOTO
 10
 指定した行番号へジャンプします

●30 個の LED をアニメーションさせる

さっきはランダムに光りましたが、順番にアニメーションして光らせてみましょう。 アニメーションするプログラムを追加します。(60~100行)



入力できたら、**RUN** (または F5 キー)で、プログラムを実行してみましょう。 30 個の LED がいろいろな色で光って、テープの先の方へアニメーションします。

70 行で、配列変数の値を3 個分(LED1 個分)ずつ後ろヘコピーしています。 それを60 行・80 行の FOR~NEXT で、LED テープの先から元の方へくり返します。 最後に90 行で、1 個目の LED 用の配列変数[0],[1],[2]に新しい値を設定します。



50 行の WAIT の値を変えると、アニメーションのスピードが変わります。 試してみましょう。



プログラムを実行すると、先端の30個目のLEDが赤く光ります(壁) 緑色のLED(ボール)が壁に向かって飛んでいき、反射してもどってきます。 ボールが1個目のLEDに来た時に、IchigoJamのボタンを押して打ち返してください。 壁はだんだん手前へ移動します。打ち返せなかった場合や、ボタンを押すのが早すぎた場 合はゲームオーバーです。ボタンを押すとリプレイします。