

アイロンビーズを光らそう

虎の巻(メンター向け)



PCN 横浜

2018/10/14

(アイロンビーズに LED を挿入するアイデアは PCN 秋葉原が開発しました)

PCN 秋葉原 info@assemblage.tokyo)

めあて

ブレッドボードを使用した初歩の電子工作を行います。本実習によりブレッドボードの構造が理解できます。IchigoJam の I/O(アイオー)機能を利用して、発光ダイオード(LED)を光らせます。LED に極性のある事を理解します。応用として LED の数が増えると、光量が減ることも学習できるでしょう。

プログラミングの観点からは、OUT, WAIT, RND 命令について学習します。OUT 命令においては 1 ビットずつ出力する方法と、まとめて出力する方法が分かります。応用として、2 進数について解説することもできるでしょう。

原理

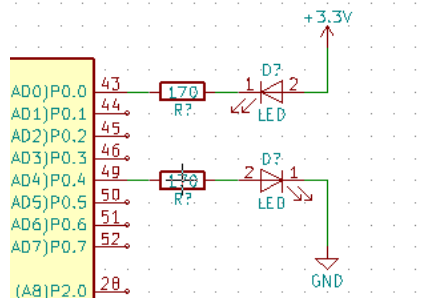
IchigoJam は I/O を持っています。I/O とは、電気的な信号を入力したり、出力したりする仕組みです。IchigoJam BASIC では、I/O の制御に IN, OUT 命令を使います。IN 命令は電気信号を入力するとき、OUT 命令は電気信号を出力するときに使用します。

LED を光らせるためには適度な電流を流す必要があります。LED には V_f という「順方向電圧」と、 I_f という「順方向電流」の二つのパラメータがあり、 V_f は色やメーカーによって異なります。白色、青色の LED では、2.8 から 3.6V が一般的です。それ以外の色では、1.5 から 2.0V 程度となります。 I_f は数 mA(ミリアンペア)から 20mA 程度流します。数 mA を下回ると光っていないのが分からないほどぼんやり点灯します。20mA を超えると LED が球切れしてしまいます。LED を IchigoJam に接続するときには、この V_f と I_f を適切な値にする必要があります。

では、どのようにすれば V_f と I_f が適切になるのでしょうか。IchigoJam の I/O は、3.3V の電源電圧が上限となっており、信号は OUT x, 0 では 0V、OUT x, 1 では 3.3V が出力されます。青色と白色 LED では、 V_f が 3.3V を超えてしまうと、原理的に電流が流れないので、光らなくなります。したがって、 V_f が 3.3V 未満であるとわかっていない限り使うことが出来ません。それ以外の色の場合は 3.3V から 1.6V 程度に電圧を落とす必要があります。電圧を落とすには、抵抗を用います。例えば、 V_f が 1.6V の LED があった場合、 $3.3-1.6=1.7V$ つまり 1.7V 電圧を落とせば良いのです。1.7V 電圧を落とすと同時に電流値も決定します。電流は 10mA 流すとします。1.7V の電圧降下で 10mA 電流を流すためには、オームの法則を用いて次のように計算します。計算結果ぴったりの値が売られていることは稀です。実際には多少の誤差は目をつむって、近い値の抵抗値を選択します。

$$R = \frac{V}{I} = \frac{1.7}{0.01} = 170 \approx 180 \Omega$$

IchigoJam に LED を接続する方法は二通りあります。直列に接続した抵抗と LED を、Vcc(3.3V)および I/O 端子の間に接続する方法と、I/O 端子および GND 端子の間に接続する方法があります。前者は OUT x,0(low)の時に LED が点灯します。後者は OUT x,1(high)の時に点灯します。



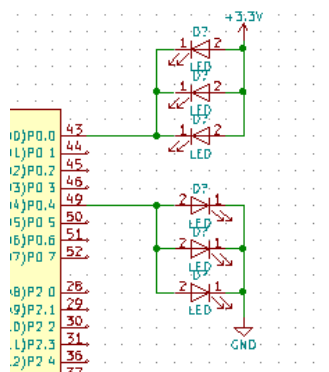
LED の知識として、アノードとカソードについて知る必要があるでしょう。LED は有極性部品です。逆方向に電流を流すことはできません。電流はアノードからカソードに流れます。LED の部品はその形状でアノードとカソードを示しています(回路図では"2"がアノード、"1"がカソードになります)。



足の長い方がアノード、短い方がカソードです。ほか、LED の消費する電圧と電流の関係について勉強すると理解が深まります(データシート参照)。

ワンポイント

アイロンビーズ工作では、一つの I/O 端子に LED を複数組み合わせることもあるでしょう。その場合は LED を(おおむね 3 つ以上)並列にすることで抵抗器を省くことが出来ます。これは IchigoJam で使用しているマイクロプロセッサの I/O 端子の特性によるものであって、すべての場合にかなう理屈ではありません。つまり、IchigoJam で使用している NXP 社の LPC1114 という IC のデータシートに、ショートサーキットカレント(短絡回路電流)max.45mA という規定のあることによります。規定電圧(Voh, Vol)では数 mA(Ivh または Ivl)しか流せないはずですが、回路をショートすると最大で 45mA もの電流が流れることを意味します。すると、I/O 端子の内部抵抗による電圧降下を織り込めば、ひとつの I/O 端子でたくさんの LED を点灯することが出来ます。これは電子回路設計としては一般的ではありませんが、工作というジャンルでは十分実用になる発見です。仮に LED が 5 個の場合と 10 個の場合で比較します。LED が 5 個の場合は I/O に 20mA 流れたとして、 $20/5=4\text{mA}$ 。つまり、LED 一つあたり 4mA 流れます。LED が 10 個の場合は同様の一つあたり 2mA となります。つまり LED の数が多くなるほど、光る光量が等分で減ってゆきます。3 つ以上抵抗無しで接続する場合は、この点に注意します。多少高価になりますが、高輝度 LED を使用すると、数 mA でも明るく点灯するでしょう。



また、青色、白色以外の LED であっても、その色によって Vf が若干異なります。したがって同じ I/O に異色の LED を混在することはできません。Vf の低いものしか光らなくなります(あるいは光がぼんやりする)。

用意するもの

アイロンビーズまたはくみ上げたもの 1つ

IchigoJam 本体 1個

ミニブレッドボード(秋月電子: BB-601(White)) 1枚

ジャンパコード黒(オス-メス)(秋月電子: DG01032-0024-BK-015) (LED の数)本

ジャンパコード赤(オス-メス)(秋月電子: DG01032-0024-RD-015) (LED の数)本

ジャンパコード赤(オス-オス)(秋月電子: 116-719-01-015) (LED グループの数)本

ジャンパコード黒(オス-オス)(秋月電子: 116-719-05-015) (LED グループの数)本

LED(秋月電子: OSDR5113A(赤) OSDG5113A(緑) OSDY5113A(黄)) 任意の数

LED を固定する場合は、グリュウガン(先から溶けた樹脂の出るコテ)

例) OUT1, 2, 3, 4 を使用して、各々LED を3つずつ光らす場合。

OUT1 を1グループ、OUT2 を2グループの順で、OUT4 を4グループとします。

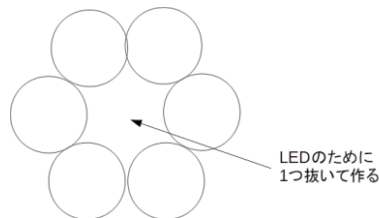
LED グループの数=4つ

LED の合計数=4×3=12個

LED の色はグループごとに共通とします。(赤6個、緑3個、黄色3個など)

作り方

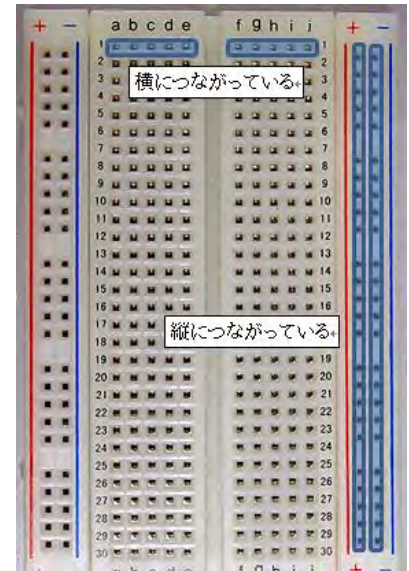
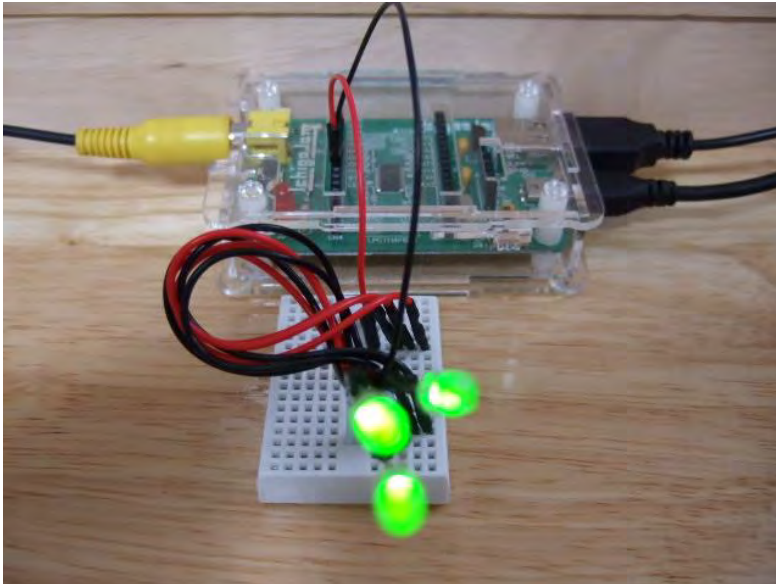
アイロンビーズをくみ上げます。その際、ビーズに LED を入れる隙間をいくつか作っておきます。



ビーズに LED を通します。LED の脚にはジャンパコードを挿し込みます。LED に使うジャンパコードは、片方がメス、もう片方がオスになっているものを使用します。LED の脚は強く挿し込む必要があります。固定する場合はあらかじめ LED の足を短くカットして(アノードとカソードが分かるように長さを変えて)、ジャンパコードと LED をグリュウガンで固定すると良いでしょう。



グループごとに LED が並列になるようにブレッドボードに組んでゆきます。最後に IchigoJam とブレッドボードをグループごとに配線します。(下図は1グループのみ配線した実例)



今回は「ブレッドボード」を使って作ります。
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。
(出典:上田市マルチメディア情報センター)

★完成写真挿入

プログラム例(OUT x,1 で光る場合))

```

10  ' BEADS LED
20  OUT 1,1
30  WAIT 60
40  OUT 2,1
50  WAIT 60
60  OUT 3,1
70  WAIT 60
80  OUT 4,1
90  WAIT 60
100 OUT 0
110 WAIT 60
120 GOTO 20

```

グループが4つの時のコード。

グループを増やすときは OUT5,1
OUT6,1 と増してゆく。

グループを減らすときは OUT4,1 か
ら行を削除してゆく。

ランダムに光らせる

```
10  ' BEADS LED RND
20  A=RND(16)
30  OUT A
40  WAIT 60
50  GOTO 20
```

グループが4つの時のコード。

グループを増減するときは
RND(16)の中の数字を変える。

グループ数=n のとき、

RND(2^n)