

講師用

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムⅡ①

(第1回/第2回テキスト)

必ず、生徒に授業日と自分の名前を記入させるようご指導をお願いいたします。

だい かい じゅ ぎょう び
第1回授業日 2024年 月 日

だい かい じゅ ぎょう び
第2回授業日 2024年 月 日

な まえ
名前



ロボット博士養成講座
ロボティクスプロフェッサーコース

2024年4月授業分

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムⅡ①

第1回

電子回路とプログラミング (LED)

講師用

目 次

0. 電子回路とプログラミング (LED)

- 0.0. 「電子回路とプログラミング (LED)」でやること
- 0.1. 必要なもの
- 0.2. 回路とマイコンボード
- 0.3. 301 ブレッドボードの仕組み

1. 電子回路のつくり方

- 1.0. 各パーツの準備
- 1.1. LED を点灯させる
- 1.2. 抵抗

2. LED のプログラム

- 2.0. マイコンの接続とプログラムの実行
- 2.1. プログラムの確認
- 2.2. LED を制御する
- 2.3. LED を 4 個つなげよう
- 2.4. for 文を活用する

3. まとめ

○ 授業開始にあたって

授業のはじめは、着席させ、大きな声であいさつしてから始めます。

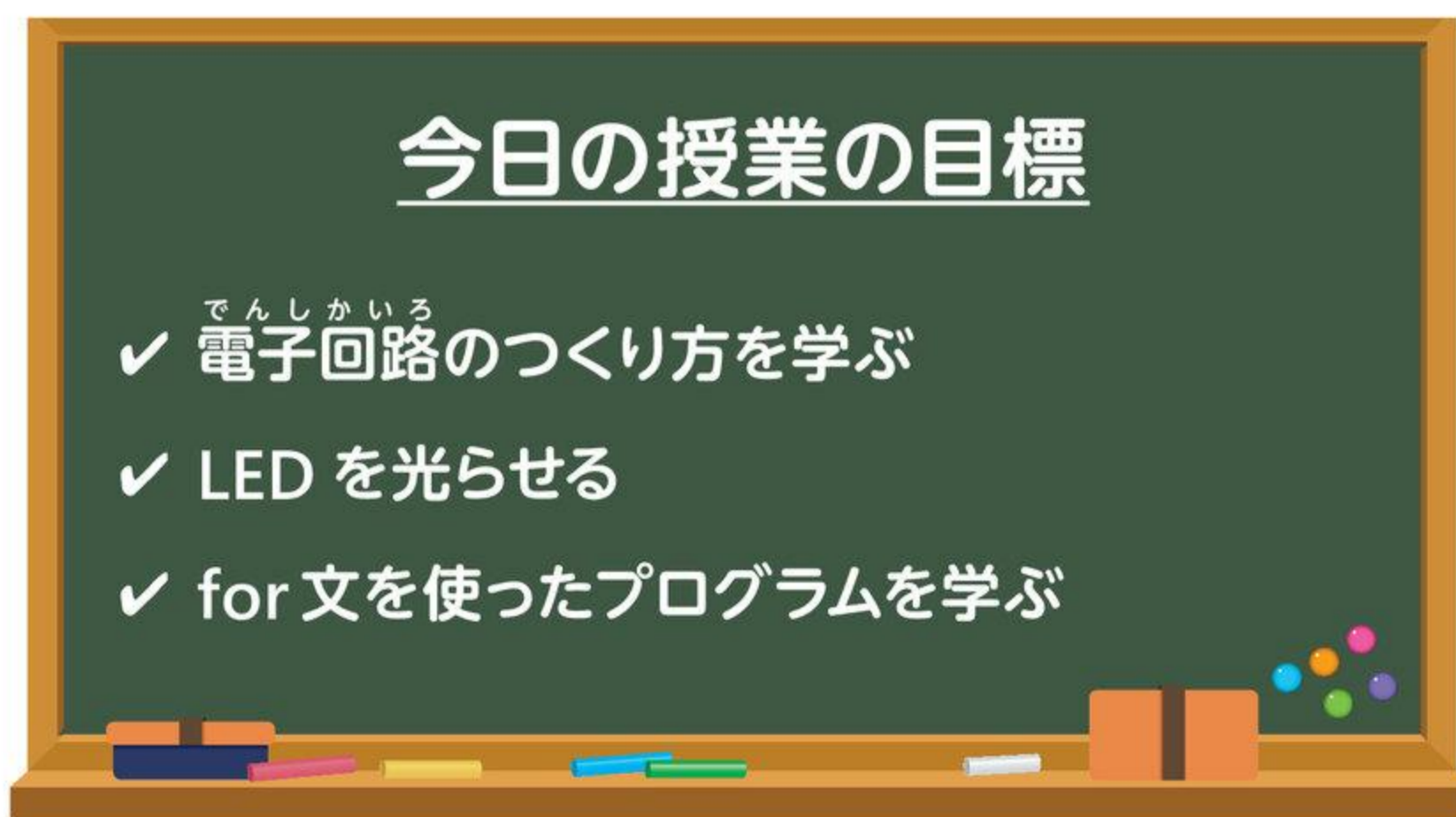
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. 電子回路とプログラミング (LED) (目安 15分)

0.0. 「電子回路とプログラミング (LED)」でやること



「不思議アイテムⅡ」では本格的なロボット製作には欠かせない電子回路の基礎を学んでいきます。私たちの身の回りの電化製品の中にも、さまざまな電子部品を使った回路が入っています。

あまり機械を分解することはないと思いますが、便利な電化製品にはかならずといっていいほどマイコンが使われていて、目に見えない電気信号でモーターやセンサーを動かしています。今回から、全6回の授業では、実際に電子回路をつくりながら、電気の流れなどを学んでいきましょう。

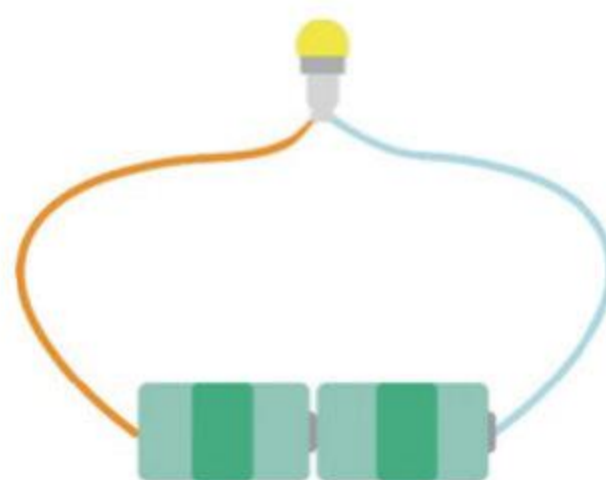
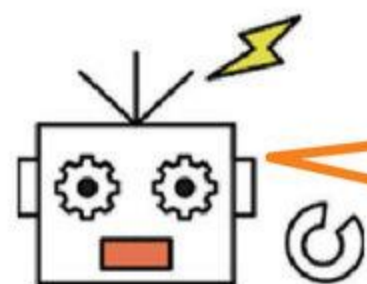


図 0-0 豆電球



ガッコウの理科実験で豆電球の回路をつくったことはないカナ？

0.1. 必要なもの

以下のパーツを準備してください。




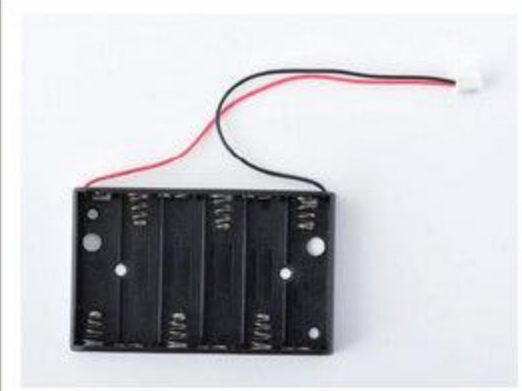
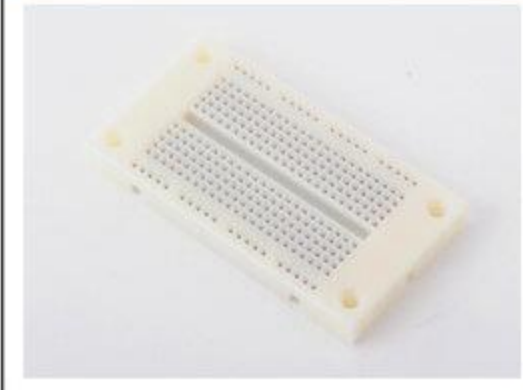



ラジオペンチ 1	USB ケーブル 1	マイコンボード 1	電池ボックス 1
			
301 ブレッドボード 1	ジャンパー線 65	抵抗 (100 Ω /1k Ω /10k Ω) 各10	緑色 LED 10
			

図 0-1 必要なもの

0.2. ^{かいろ}回路とマイコンボード

小学校や中学校の理科でも学びますが、^{かんてんち}乾電池やコンセントなどの電源から電気を得るとき、基本的には一周して戻ってくるような電気の通り道、つまり「^{かいろ}回路」をつくる必要があります。もし一周していなければ、電子部品をつないでも動作しません。

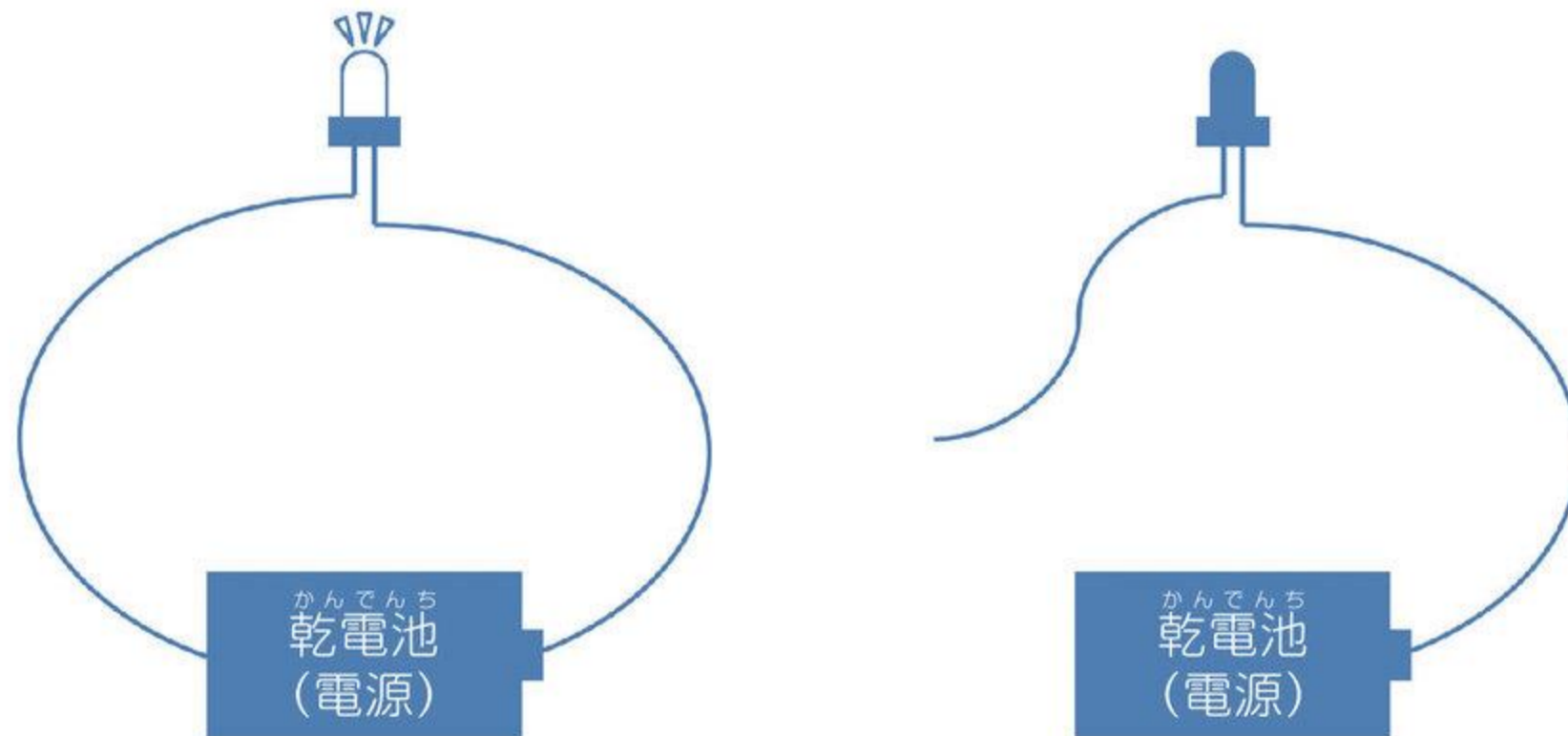


図0-2 ^{かいろ}回路

たとえば、電子部品を^{かんてんち}乾電池のプラス極とマイナス極につなぎ、^{かいろ}回路をつくれれば動作します。ロボプロで使用しているマイコンボードも同じように、電子部品と^{どうせん}導線でつなぐことで^{かいろ}回路をつくり、電気を供給することができます。

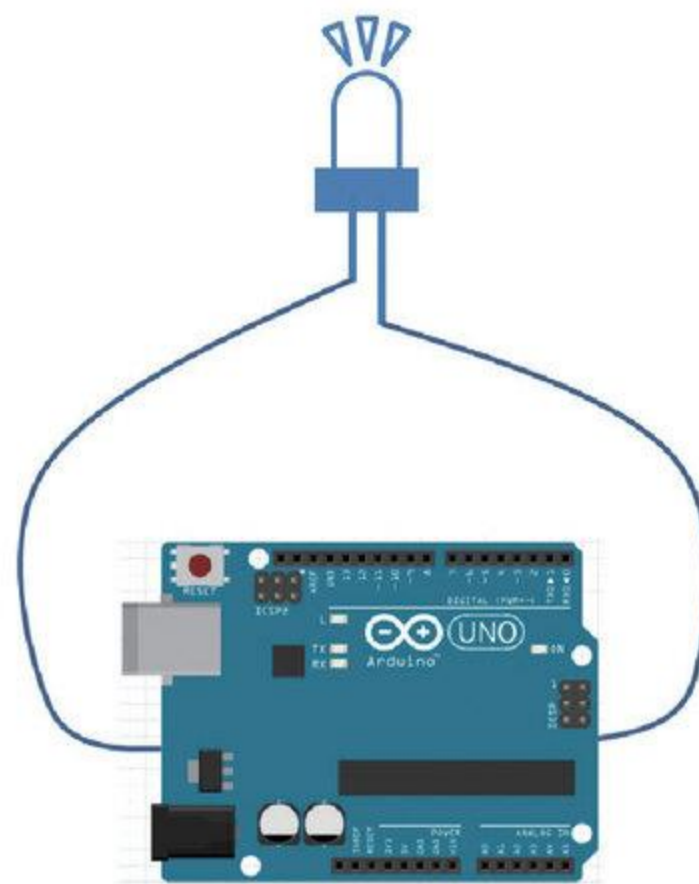


図0-3 マイコンボードを用いた^{かいろ}回路

たとえばギアドモーターを使うロボットでは、マイコンボードにロボプロシールドを取り付け、ギアドモーターのコネクターをさし^こ込んでいましたね。これで、マイコンボードからギアドモーターを通り、またマイコンボードに戻ってくるという^{かいろ}回路をつくっていたのです。

コネクターを使わなくても、^{かいろ}回路ができてさえいれば電子部品を動作させることができます。今回はマイコンボードに直接^{どうせん}導線をさして、「^{かいろ}回路」のつくりをしっかりイメージできるようにしてみます。

また、接続した電子部品に直接命令を出すようなプログラムに触れ、これまでのロボット製作で登場したプログラムが、各パーツにどのような指示を出していたのかを振り返ってみましょう！

0.3. 301ブレッドボードの仕組み

301ブレッドボードは電子部品をさし込むだけで電子回路をつくることのできる、とても便利な道具です。電子部品やジャンパー線を抜きさしして、何回も再利用できるので、さまざまな組み合わせで気軽に回路をつくることができます。

301ブレッドボードを上から見ると、穴がいっぱい空いています。図0-4の赤線を引いたところに電気の通り道（導電体）があって、そこに沿ってプラス側からマイナス側へ電気が流れるように電子部品をさし込んでいきます。

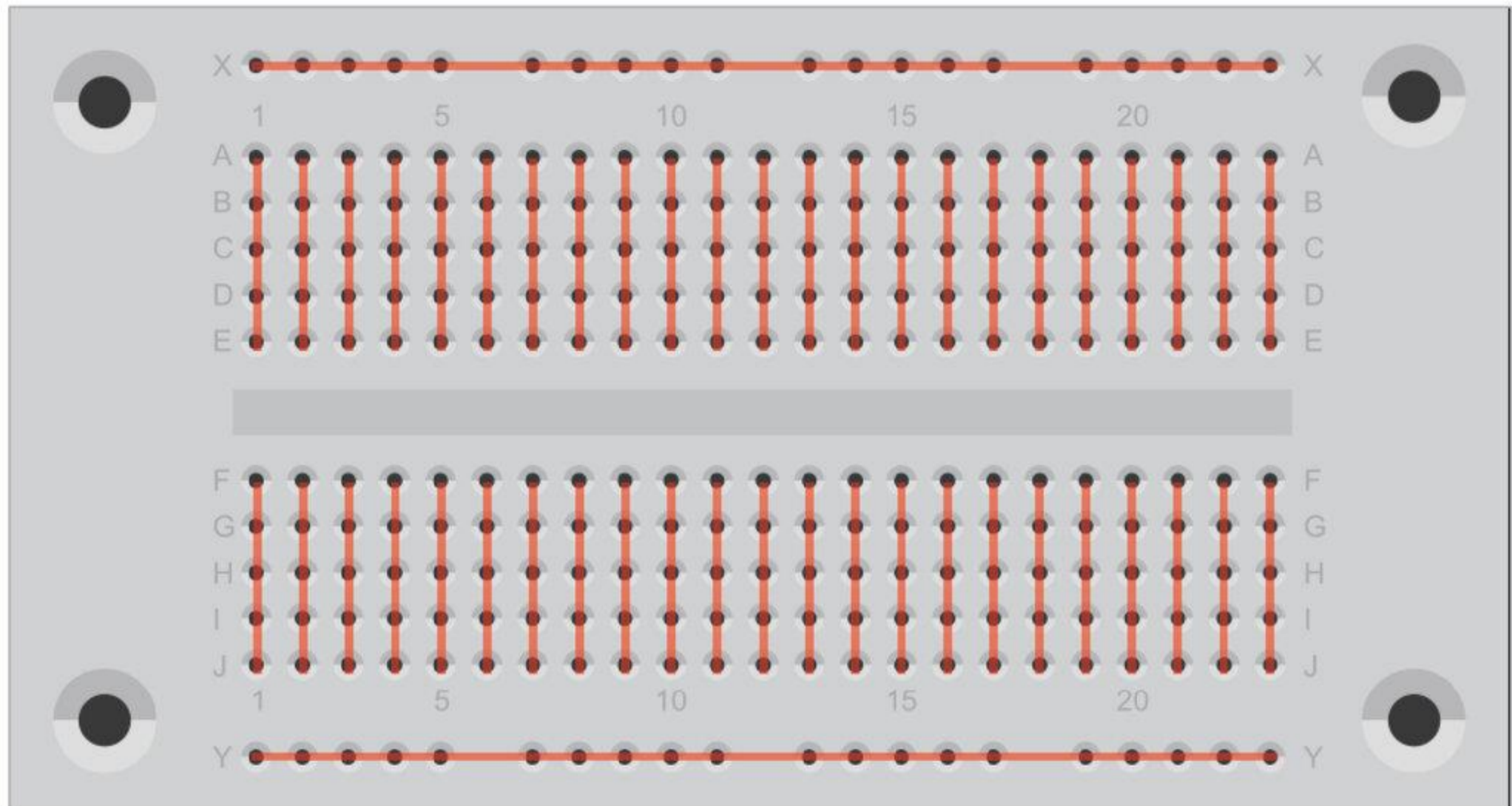


図0-4 301ブレッドボードの内部接続図

講

電子回路は英語では、「エレクトロニック サーキット Electronic circuit」といいます。電流が流れるサーキットと捉えると、理解しやすいと思います。電子回路は、レースのサーキットのように、ループ状にしなければなりません。途中で切れていると、電流が流れないからです。つまり、電池を使った場合、電池のプラス側から導電体を介して、素子（電子部品）をマイナス側に繋げる必要があります。マイコンボードから電気を流す場合、プラス側（電源、出力ソケット）から始まり、マイナス側（グラウンド GND）に繋げる必要があります。

1. 電子回路の作り方 (目安 25 分)

1.0. 各パーツの準備

さて、ここでは実際に 301 ブレッドボードにパーツを取り付けて LED を光らせる実験をします。

1) 緑色 LED の使い方

LED は一般的に足の長さで極性が表されています。長い方の線が「アノード (+)」で、短い方の線が「カソード (-)」です。図面では、足の線が曲がっている方がアノードになります。なお、アノードとカソードを反対向きに取り付けても点灯せず、LED やマイコンボードが壊れてしまう可能性があるため注意しましょう。

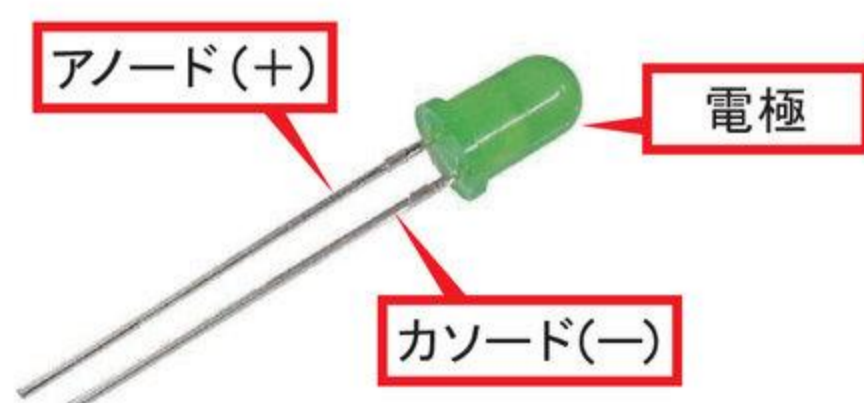


図 1-0 LED の極性 (実物の見分け方)

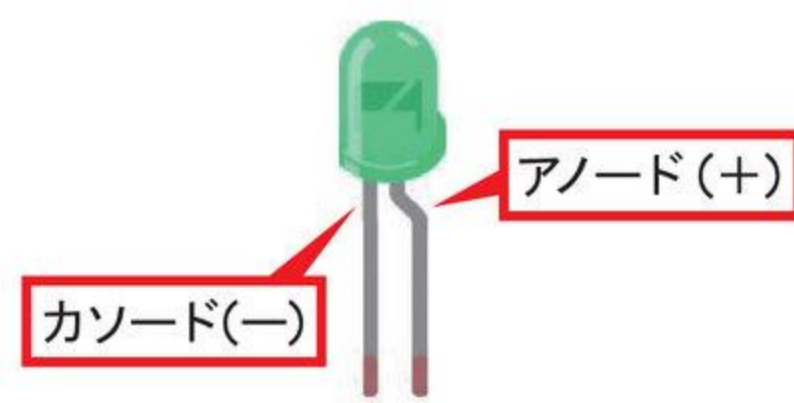


図 1-1 LED の極性 (図面の見分け方)

2) 抵抗の使い方

乾電池やマイコンボードから電子機器に直接電流を流すと、機器に大きな負荷がかかり壊れてしまうことがあります。これを防ぐため、回路をつくるときには「電流をジャマする部品」を取り付けておき、少し流れを抑えるようにしておくことが多いです。

電流をジャマするには「抵抗 (抵抗器)」という部品を使います。抵抗は電流をジャマする作用の強いものや弱いものがあり、部品に書かれた線の色を読み取ることで種類を見分けられます。

電流をジャマする作用の強さは「 Ω (オーム)」という単位をつけて数字で表せます。g (グラム) や m (メートル) と同じく、1000 Ω をまとめて 1k Ω (キロオーム) とすることもできます。

今回は、「茶・黒・茶・金」の線が書かれている抵抗をメインで使います。この色の抵抗は「100 Ω 」であることを表します。ロボプロでは他にも「1k Ω 」と「10k Ω 」の抵抗が登場します。

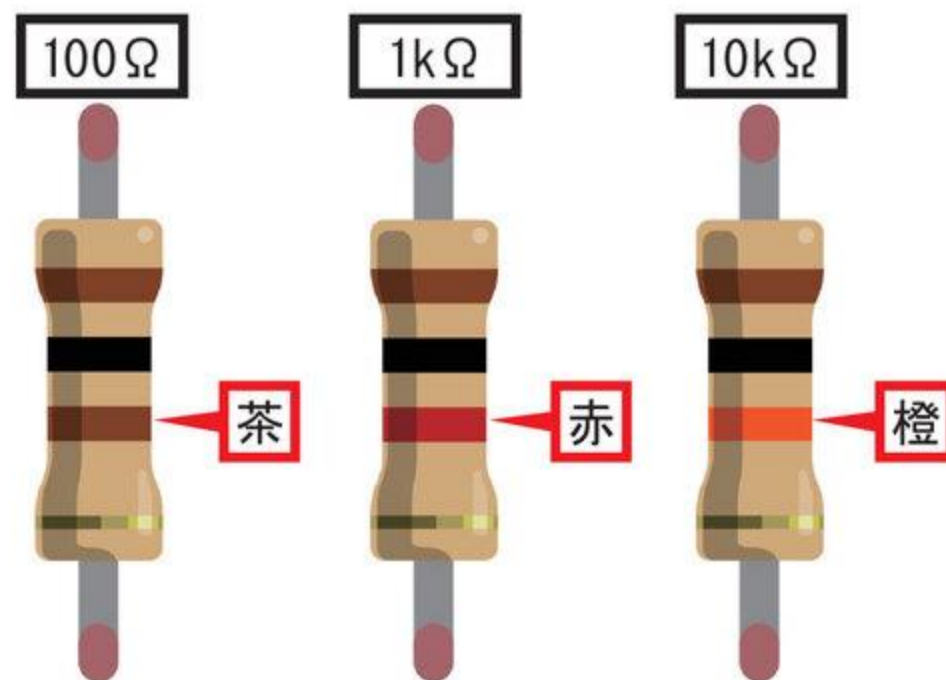


図 1-2 抵抗の種類

図 1-3 のように抵抗の足を曲げてから、301 ブレッドボードにさし込みます。なお、この抵抗には向きはありません。

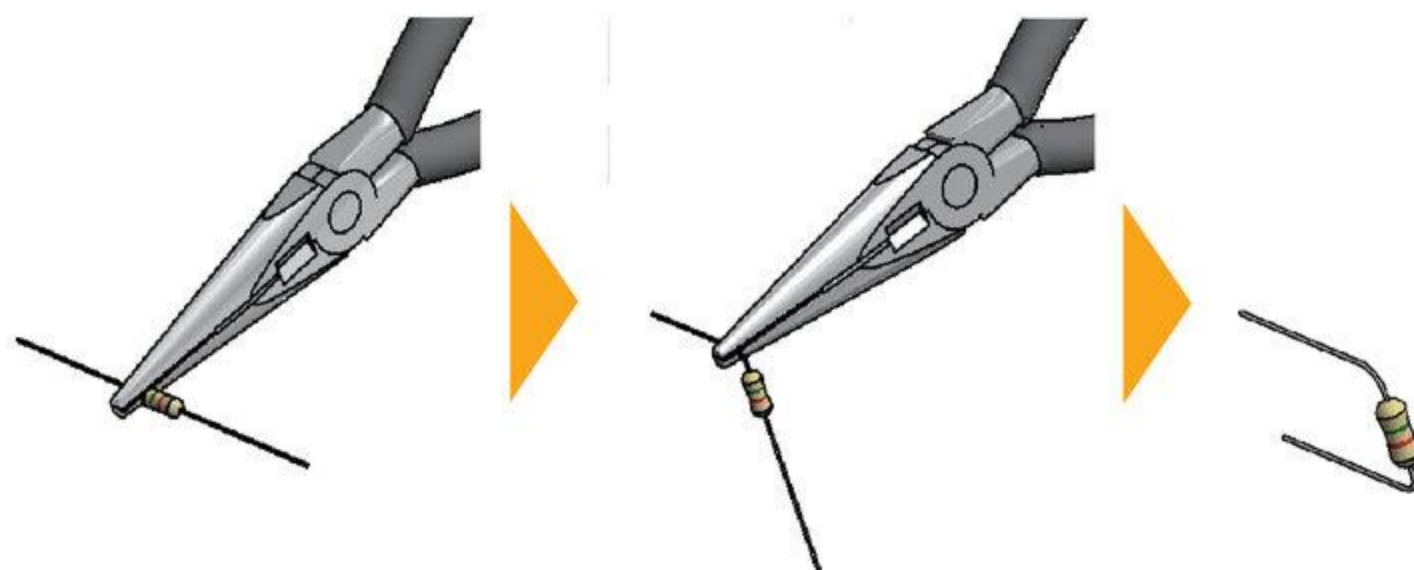


図 1-3 抵抗の曲げ方

3) 電池ボックスの使い方

実験によっては、電池ボックスから直接電気を流して各LEDを点灯させたりします。その場合は、**図1-4**のように、電池ボックスのコネクタに直接ジャンパー線をさし込みます。電池ボックスのリード線は、赤がプラス極、黒がマイナス極と決まっていますが、ジャンパー線の色はそれらとちがっていてもかまいません。

ただし、線のプラス極とマイナス極は絶対に接触させないようにしましょう。電池が発熱したり、液漏れや破裂をしたりする原因になります。301ブレッドボードと電源コネクタの接続を片方ずつ完了させたり、長いジャンパー線を使って物理的に接触しにくいような工夫をしたりしましょう。

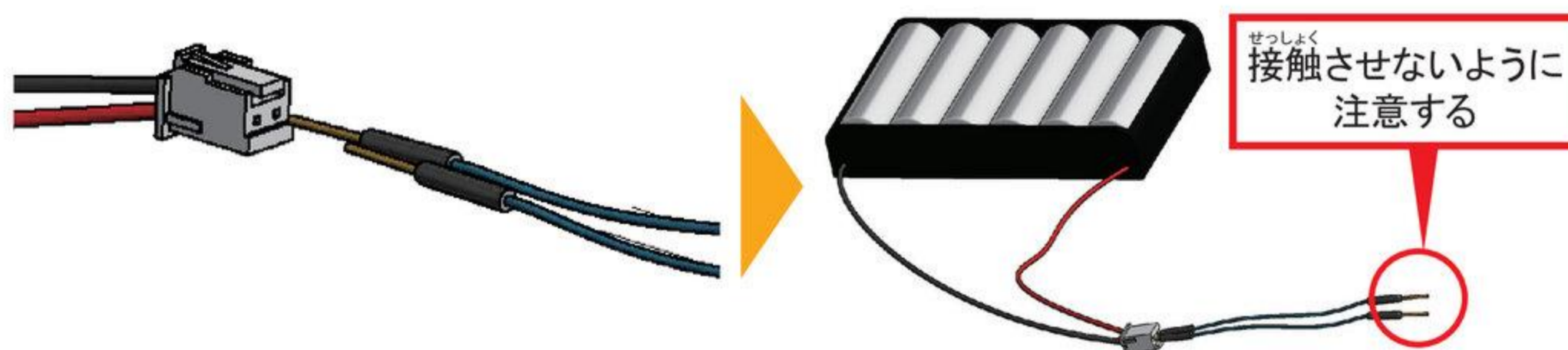


図1-4 電池ボックスの使い方

⚠ 注意！

今後回路をつくる際、つなぎ方を間違えて電源のプラス極とマイナス極を直結させてしまうと、やがて回路から変なにおいがしたり、電池や部品が異常に発熱したりすることがあります。その場合はあわてず、ジャンパー線を引き抜くなどして電気が流れないようにしましょう。ただし、乾電池やLEDの脚などの金属部分は非常に熱くなっている場合があるので、絶対に触れないようにしてください。

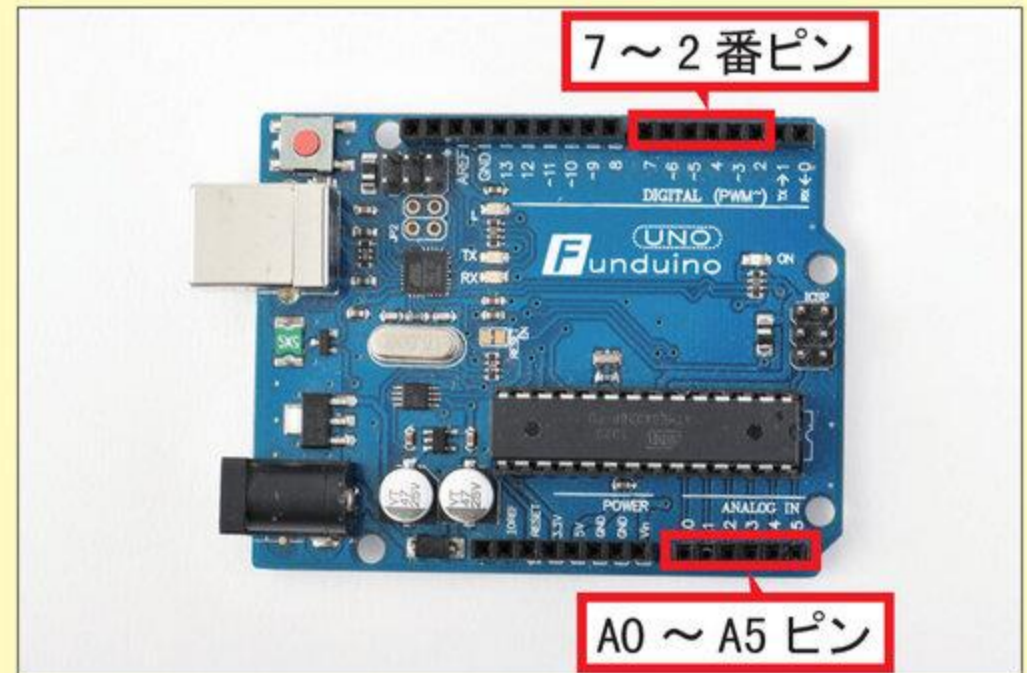
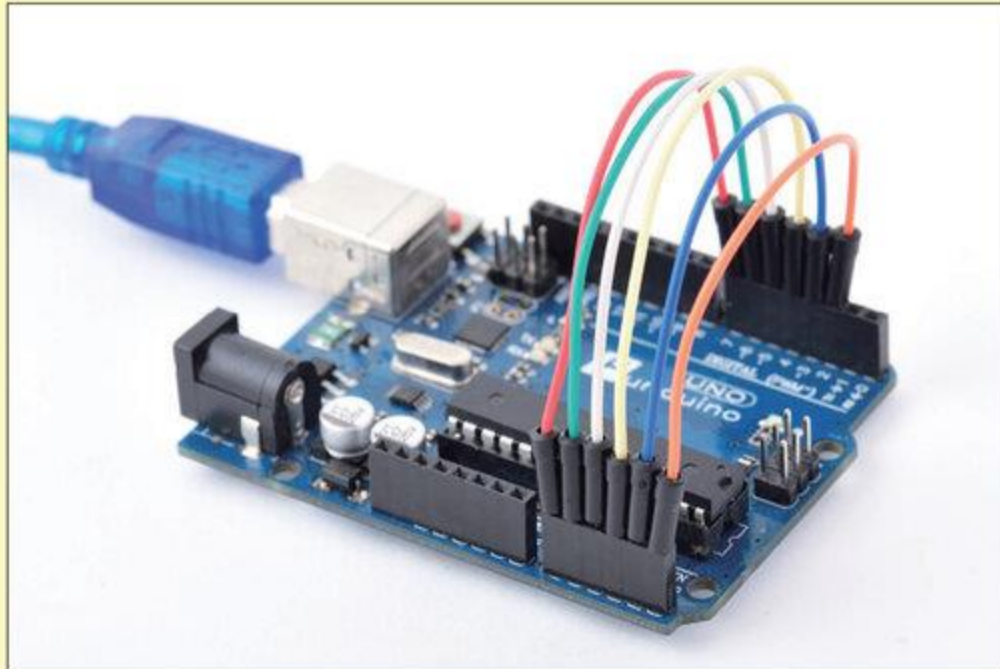


POINT

●ジャンパー線の導通チェック

うまく動作しない場合はジャンパー線が断線している可能性があります。その場合は、チェックしましょう。

まず、以下の図のようにマイコンボードの「7～2番ピン」を「A0～A5ピン」にそれぞれ対応するように接続します。



次に以下のプログラムを実行します。

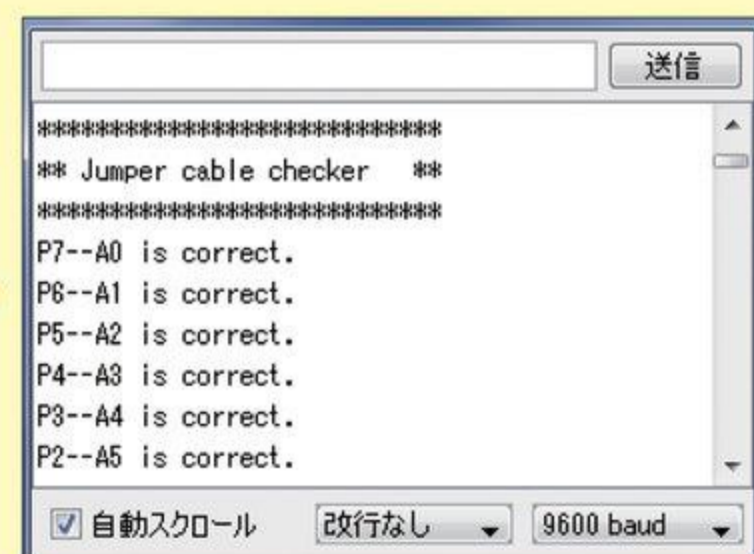
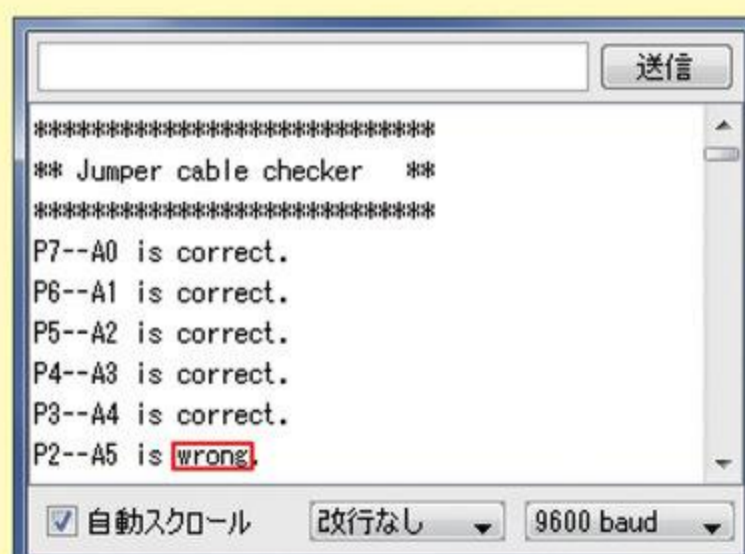
∞ プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA1 > JumperCheck

開発環境ソフトの画面右上の端の虫眼鏡マークをクリックし、シリアルモニターを開きます。「9600baud」に設定しましょう。



シリアルモニターに以下のように各ピンの導通状況が表示されます。つながっていれば「correct」、断線の場合は「wrong」と表示されます。「wrong」と表示されているものは入れかえて「correct」と表示されるものを使用するようにしましょう。



1.1 LEDを点灯させる

各パーツの使い方が確認できたので、実際にLEDを点灯させてみましょう。まずは、 1-5の回路をつくります。電気の流れを意識して部品をさし込みましょう。

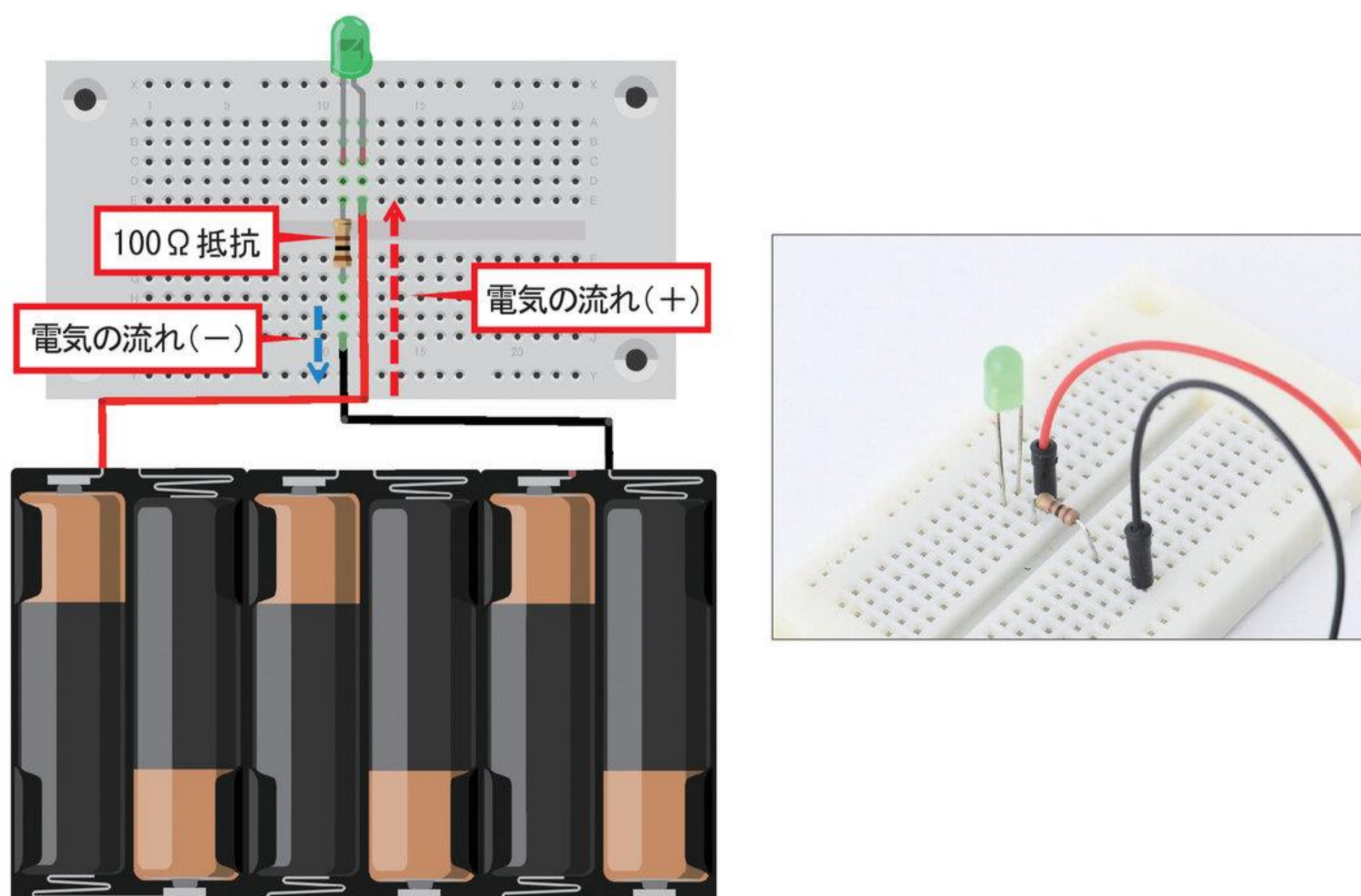


図 1-5 電子回路（回路のイラストと完成写真）

やってみよう！

抵抗を $100\ \Omega$ から、 $1\text{k}\ \Omega$ 、 $10\text{k}\ \Omega$ の順に交換して LED の光の変化を観察したらどうなるかな？ ○で囲もう。

- 抵抗を $1\text{k}\ \Omega$ （赤のシマがある抵抗）に交換した場合、 $100\ \Omega$ のときとくらべて、（少し明るくなる ・ **少し暗くなる**）。
- 抵抗を $10\text{k}\ \Omega$ （橙のシマがある抵抗）に交換した場合、 $100\ \Omega$ のときとくらべて、（かなり明るくなる ・ **かなり暗くなる**）。

3種類の抵抗を使ってみて、3パターンの明るさを確認しましたが、違いはわかりましたか？

抵抗値によって、明るさが変化したかと思います。抵抗の数字が大きくなると回路の電流がよりジャマされます。その結果、LEDが暗くなってしまいましたね。



コラム 電圧・抵抗・電流とオームの法則

●電圧

お風呂のお湯を、栓を引き抜いて排水する様子を思い浮かべてください。栓を引き抜いた直後は勢いよく水が流れていくのに、だんだん水の勢いが弱くなっていきますね。

電気にも、それと同様に「勢い」のようなものがあります。「電気を流そうとする力の大きさ」とも言い換えられます。その大きさのことを「電圧」と呼び、「V (ボルト)」という単位で表します。たとえば、皆さんの家にあるコンセントは、プラグをつなぐと100Vで電気を流そうとしてきます。一般的なアルカリ乾電池は、1つにつき1.5Vです（ロボプロでは6本を直列つなぎにしているので、 $1.5 \times 6 = 9V$ になっています）。

●抵抗と電流

お風呂の場合、同じ水の量で栓を引き抜いても、排水口が大きいお風呂と小さいお風呂では、流れていく水の量は違うはずですよ。電気も同じように、「同じ電圧をかけても、電気がたくさん流れるときとそうでないときがある」のです。ここで「抵抗」の考え方が必要になります。「やってみよう！」で、100Ωの抵抗を1kΩ（1000Ω）に交換したときのことを考えてみましょう。電池の本数は変わっていないので、電圧も変わっていないはずですが、しかし、交換することでLEDは暗くなったので、流れる電気が小さくなったことがわかります。抵抗とは「電気の流れを妨げるもの」なので、この値が大きくなると、より電気が流れにくくなるようジャマされてしまうというわけです。「実際に回路を流れた電流の大きさ」には「A (アンペア)」という単位をつけて表します。たとえば、「100Ωの抵抗を持つ回路に、100Vの電圧をかけたら、1Aの電流が流れた」などと表現します。

●オームの法則

電圧 [V]、抵抗 [Ω]、電流 [A] はそれぞれ、変数にするときは「E・R・I」の3つのアルファベットを使うのが一般的です。この3つの値の間には、必ず次のような式が成り立ち、これを「オームの法則」と呼びます。

- ・E=Electromotive force(電圧)
- ・R=Resistance(抵抗)
- ・I=Intensity of electric current(電流)

オームの法則

$$E \text{ (電圧V)} = I \text{ (電流A)} \times R \text{ (抵抗値}\Omega\text{)}$$



「E・R・I」のうち、2つが決まればあと1つは計算で求められることが分かりますね。「新たに電気回路をつくりたいけど、何Aくらいの電流が流れるか、あらかじめ予想したい」、「4Aで動く機械をつくりたいけど、何Ωの抵抗を使えばいいんだろう」などが、実際に回路をつくったり、電気を流したりすることなく予想ができるようになるのです。

1.2. 抵抗

授業では、3種類の抵抗を使いますが、世の中にはもっとたくさんの種類の抵抗が存在します。

何種類もある抵抗値を記憶するのは難しいので、抵抗はシマの色とパターンで見分けます。抵抗を読むときは、金色のシマを一番右にして読み取っていきます。シマは左から「第一数字」、「第二数字」、「第三数字」、「許容差（誤差）」を表します。

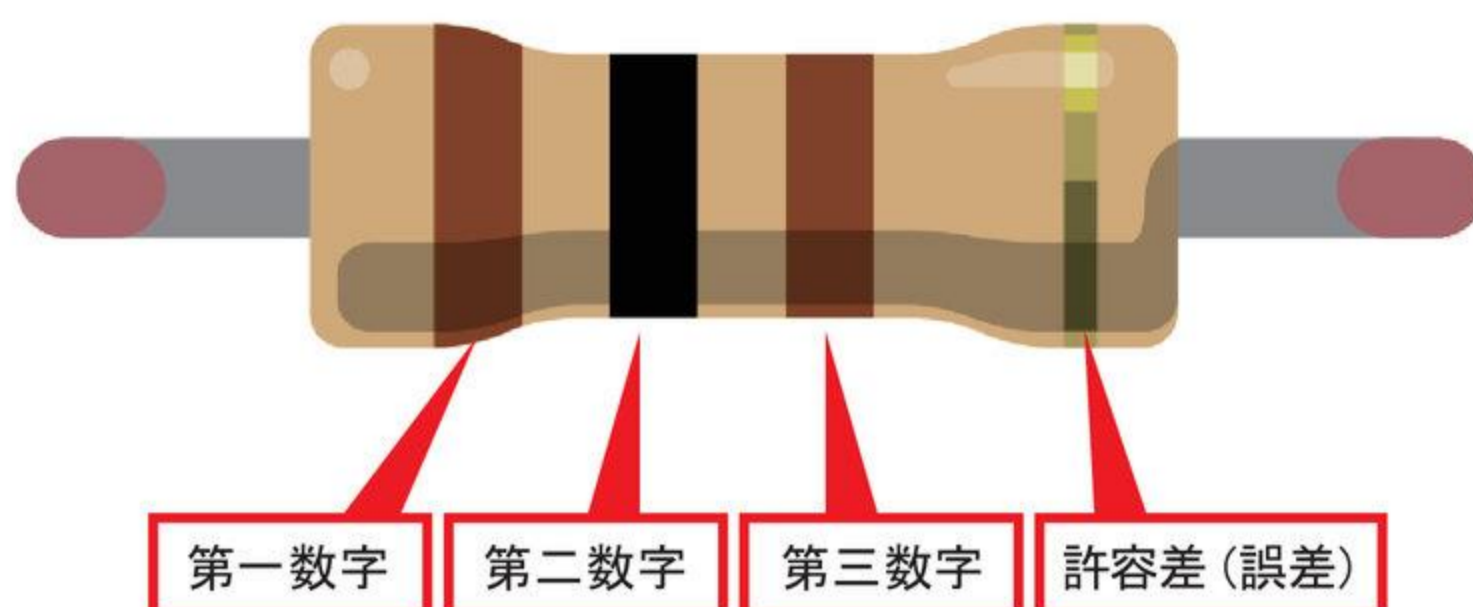


図 1-6 抵抗値の読み方

表 1-0 が各項目の値です。各項目の数字を読み取り、第一数字を10の位、第二数字を1の位だと考え、それを第三数字でかけると抵抗値がわかります。

例えば図 1-6 の場合は、左から「茶 = 1、黒 = 0、茶 = 10」ですね。そのため「10 × 10 で 100 Ω」だとわかります。

「茶・黒・赤・金」の場合は、「茶 = 1、黒 = 0、赤 = 100」のため、「10 × 100 で 1000 Ω (=1kΩ)」だとわかります。

表 1-0 抵抗値の読み方

色	第一数字	第二数字	第三数字	許容差 (誤差)
黒	0	0	$10^0=1$	
茶	1	1	$10^1=10$	± 1%
赤	2	2	$10^2=100$	± 2%
橙	3	3	$10^3=1000$	
黄	4	4	$10^4=10000$	
緑	5	5	$10^5=100000$	
青	6	6	$10^6=1000000$	
紫	7	7	$10^7=10000000$	
灰	8	8	$10^8=100000000$	
白	9	9	$10^9=1000000000$	
金			$10^{-1}=0.1$	± 5%
銀			$10^{-2}=0.01$	± 10%

ここまでの説明で電子回路の基本はわかってきましたか？

ここからは、301ブレッドボードの回路とマイコンボードを組み合わせ、プログラムを使ってLEDの光をコントロールします。

2. LED のプログラム (目安 60 分)

2.0. マイコンの接続とプログラムの実行

さて、電気回路や電子部品の性質を学んだところで、いよいよマイコンボードを使った回路づくりにチャレンジしてみましょう！

とはいえ、先ほど電池ボックスでつくった回路と形はほぼ同じです。電源が電池ボックスからマイコンボードに置きかわるだけ、とイメージするとわかりやすいですね。

電池ボックスのコネクタにさし込んでいたジャンパー線を引き抜き、プラス極側にささっていたジャンパー線はロボプロシールドの [8] と書いてある穴 (ピン) に、マイナス極側にささっていたジャンパー線は [GND] と書いてあるピンに、それぞれ接続するのが主な変更点です。

[GND] ピンは全部で3か所ありますが、今回は8番ピンと同じ列にあるものを使いましょう。

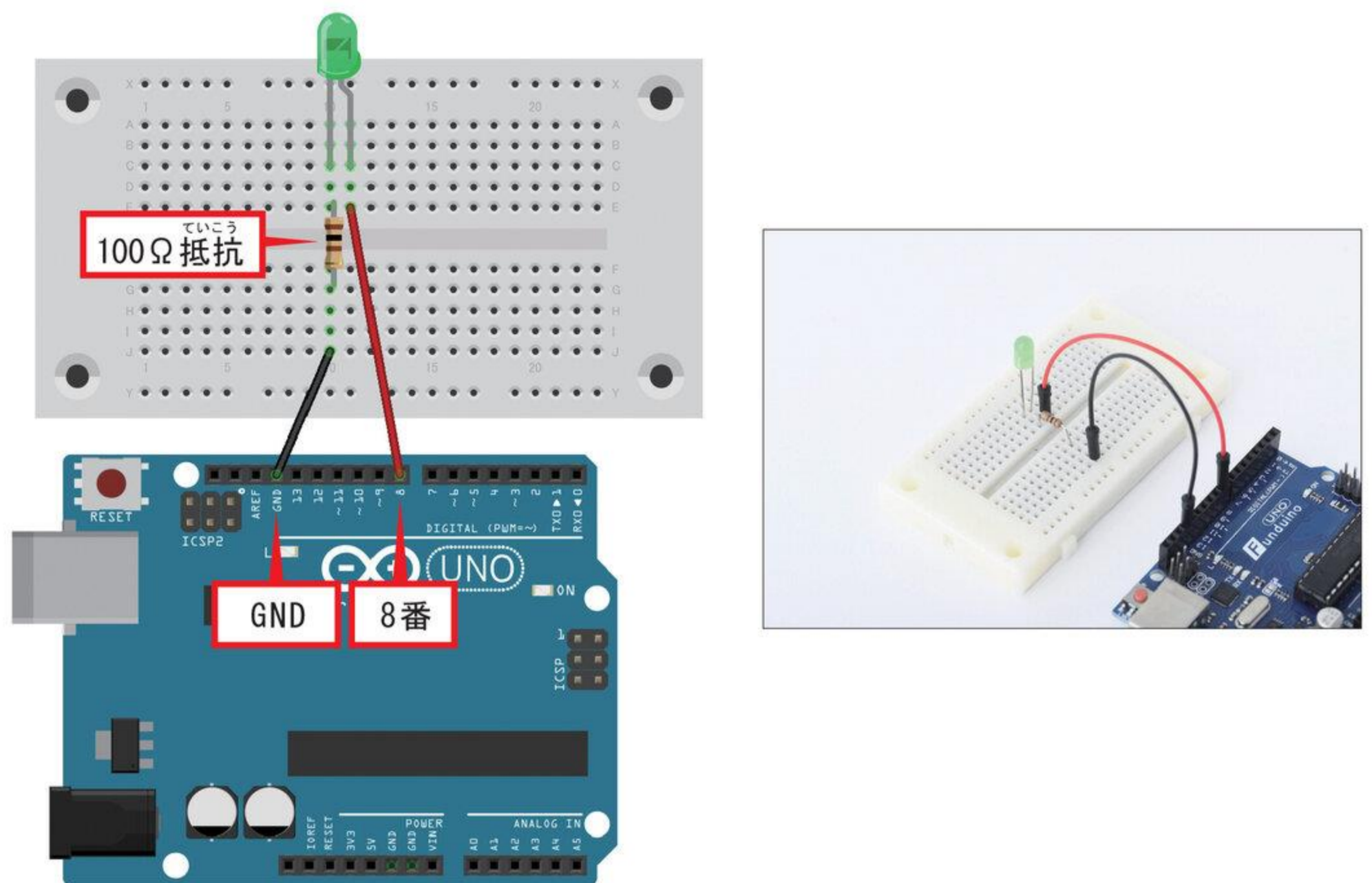


図 2-0 マイコンボードへの配線

これで回路は完成です。とてもシンプルですね！

電池ボックスとちがうのは、マイコンボードはただ導線をつないだだけでは電流を流さないという点です。これまでのロボットでもそうしてきたように、まずプログラムで命令する必要があります。

ために、「8番ピンから電流を流す」というプログラムを実行してみましょう。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA1 > Blink8

さて、どのように動作しましたか？ 緑色LEDが1秒間隔で点滅したら成功です！

2.1. プログラムの確認

1) プログラム全体の確認

では、プログラム「Blink8」をのぞいてみましょう。非常にシンプルなプログラムです。

□ プログラム「Blink8」より抜粋

```
int led = 8; // ledという名の変数を用意する(led=8)

void setup(){
  pinMode(led, OUTPUT); // 8番のピンを出力に設定する(led=8)
}

void loop(){
  digitalWrite(led, HIGH); // LEDをスイッチオン
  delay(1000); // 1秒間お休み
  digitalWrite(led, LOW); // LEDをスイッチオフ
  delay(1000); // 1秒間お休み
}
```

一番はじめの `int led = 8;` はこれまでに何度も登場しましたね。 `led` という変数をつくり、その値を8としています。

今回、特に重要なのは以下の2つの命令です。

命令 [pinMode]

動作内容：指定の番号のピンを、指定のモードで使う

使い方：pinMode(8, OUTPUT);

// 8番ピンは、出力（つないだ部品に電流を流す）モードで使う

命令 [digitalWrite]

動作内容：出力に指定したピンの電圧を5Vにする（または、0Vにする）

使い方：digitalWrite(8, HIGH); // 8番ピンに5Vの電圧をかける

片方ずつ見ていきましょう。

講

プログラムの基本構成

- ①宣言などは、利用する別のプログラムファイルの定義や、プログラムで使う変数や関数を宣言するのに利用します。
- ②「setup()」関数は、プログラムを実行した後に、一度だけ実行される「初期設定」です。使われる電子部品や利用されるライブラリの設定などを行います。
- ③「loop()」関数は、繰り返し処理を行う関数です。プログラムの最後まで到達すると、loop()の初めに戻り再度実行を行います。

2) ピンのモード指定

マイコンボードのピンには、ジャンパー線などを使えば様々な電子部品をつなぐことができます。今回はLEDをつなぎ、マイコンボード側から電流を流して点灯させましたが、センサーをつなぎ「マイコンボードのほうが、センサー側から流れてきた電流を受け取る」となるような場面もあるはずですよ。

そのため、ピンを使う際には、そのピンが「電流を外に流すピン（出力ピン）」なのか、「流れてきた電流を受け取るピン（入力ピン）」なのかを指定しておく必要があるのです。

今回のプログラムでは `pinMode(led, OUTPUT);` とあります。変数 `led` は `8` と等しいので、実質 `pinMode(8, OUTPUT);` と書いてあるのと同じですね。 `OUTPUT` は出力（外へ向けて信号を発すること）という意味です。

つまり、この命令が「8番ピンは出力モードで使うよ！」という宣言せんげんになっているのです。この宣言せんげんははじめに1回するだけで済むので、`void setup()` 内に書かれています。

3) 出力ピンから電流を流す

8番ピンを出力モードに設定したので、あとは実際に電流を流すよう命令するだけです。このプログラムでは `digitalWrite` という命令で、ピンの電圧を5Vにしたり0Vにしたりして、電流のオン・オフを制御せいぎよしています。 `()` の中に `HIGH` と書けばオン、`LOW` と書けばオフになるというわけです。オン・オフの時間は、おなじみの `delay` 命令で制御せいぎよしていますね。

2.2. LEDを制御する

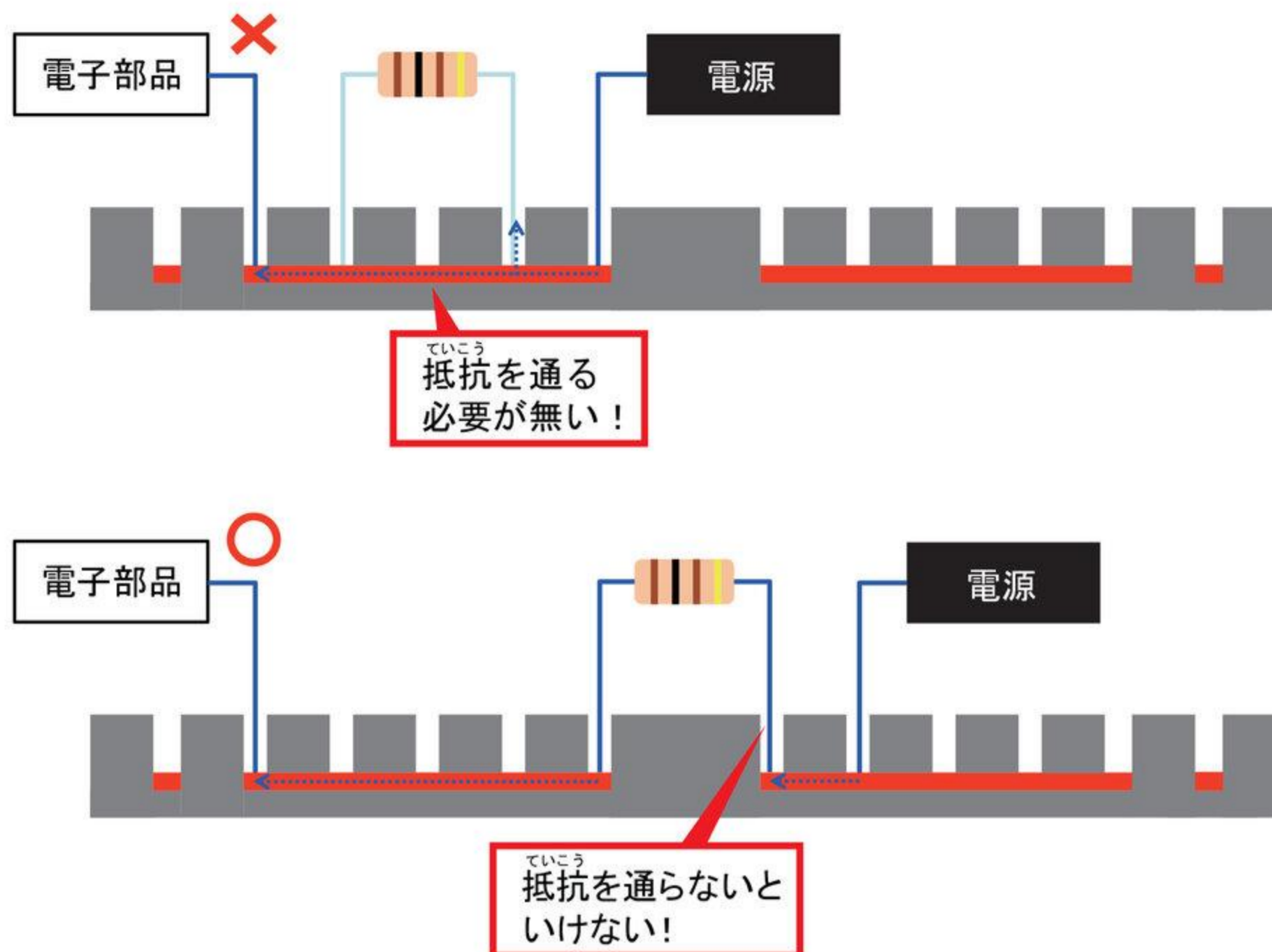
これまで使用してきたギアドモーターやマトリクスLEDなどは、狙い通りに動作させるのに電圧のかけかたを細かく調整したり、複数のピンに電流を流したりしなければなりません。そのため `digitalWrite` だけでは正確に動作させることができません。しかし、LEDであれば1つのピンに電圧をかければとにかく点灯するので、今回のようなシンプルな命令で制御できるのです。

やってみよう！

- ① プログラム「Blink8」と回路を変更し、2番ピンにつないだLEDが点滅するようにしてみよう！
- ② さらにプログラムと回路を変更し、8番ピンと9番ピンにLEDをつなぎ、2つのLEDが同時に点滅するようにしてみよう！

⚠️ 注意！

回路をつくるときは、「LEDのアノード・カソードの向き」に注意することはもちろんですが、それ以外にも「回路の中に、抵抗を通らずに一周できてしまうルートはないか」という点にも気をつけましょう。



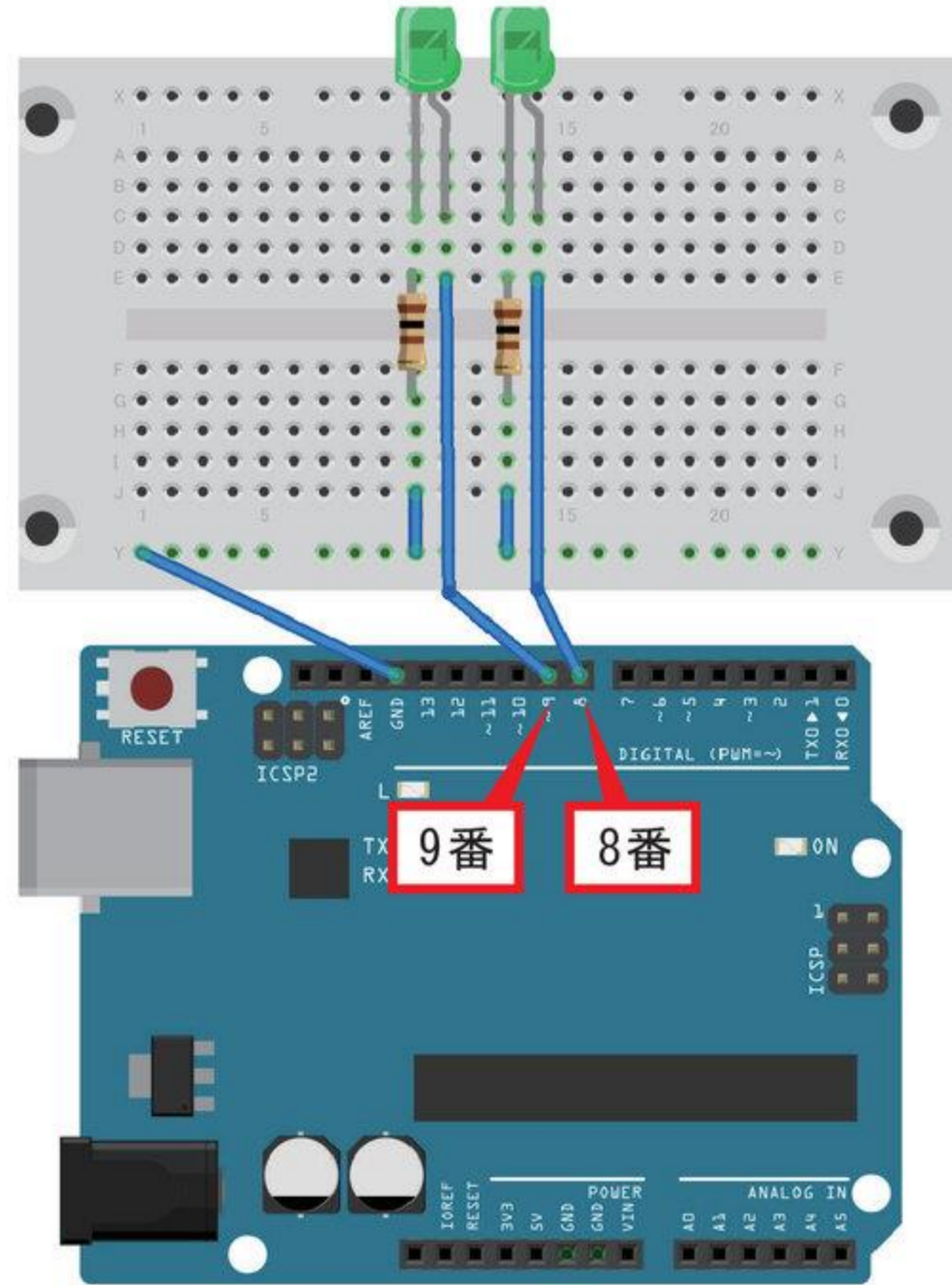
抵抗をつないでいても、他に通れる道があれば電流はそちらに流れ込みます。「抵抗がなければ、回路がつながらない」という形になるように抵抗の場所を工夫しないと、パーツが壊れるので注意しましょう。

- ①: 8番ピンのジャンパー線を2番ピンにさしかえ、`int led = 8;` を `int led = 2;` に変更します。`int` を用いることで、書きかえる場所が1か所で済んでいます。
- ②: 解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA1 > Blink89synchro

回路は以下の解答例をご参照ください。`GND` ピンを2つのLEDで共有しなければならないので、若干の工夫が必要です。抵抗の向きや、ジャンパー線の色は問いませんが、LEDの向きには注意が必要です。

ちなみに、プログラム「Blink89synchro_inverse」は2つのLEDが交互に点滅するよう変更したプログラムです。



2.4. for 文を活用する

LEDの数も増えてきたので、イルミネーションのように光らせてみましょう。今回は、以下の4つのステップをくり返すプログラムをつくります。



POINT

- ① 6番ピンのLEDだけ点灯。
- ② 7番ピンのLEDだけ点灯。
- ③ 8番ピンのLEDだけ点灯。
- ④ 9番ピンのLEDだけ点灯。

4つのLEDに命令を出すとなると、digitalWriteも同時に4行ずつかく必要があり、プログラムが一気に長くなってしまいますね。かくのが面倒なだけでなく、命令の数が多いとマイコンも処理が大変になりますし、何よりもかいている途中でミスが混じってしまう危険が高まります。

少しでもプログラムを短くまとめられるように、次の命令を活用しましょう。これまでも登場してきた命令ですが、念のためおさらいです。

命 令 [for]

動作内容：指定した条件から外れるまで、動作をくり返す

使 用 方：

```
for(i = 2; i < 7; i++){  
    digitalWrite(i, HIGH); // i番ピンをスイッチオン (iは2~6)  
}
```

上の例の場合、まず `i = 2` からスタートなので、`digitalWrite(2, HIGH);` の命令を実行します。次に `i++` なので `i = 3` になり、`digitalWrite(3, HIGH);` を実行します。このまま `i` を1ずつ増やしていき、`i < 7`、つまり `i` が7未満である間くり返し続けます。(つまり最終的に2~6番ピンがオンになります)。

ただし、この場合、for文の中で使う変数 `i` は、事前に宣言しておく必要があります。

for文を活用して、シンプルなLED制御プログラムをつくってみましょう！

ステップアップ

プログラム「Blink8」を書きかえ、for文を活用し4つのLEDを順番に点灯させるプログラムにしてみよう！

講

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA1 > Nightrider

上の例のようにすると点けたLEDが消えずに点きっぱなしになるため、for文の中に消灯命令も併せて入れる必要があります。

チャレンジ課題

順番に点灯させることができたなら、さらに動作を改造してみよう！
たとえば、for 文でも LED の点灯が左右に往復するようになれるかな？
成功したら、さらに自分なりに点灯の順番を変えてみよう！

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA1 > Nightrider2

講

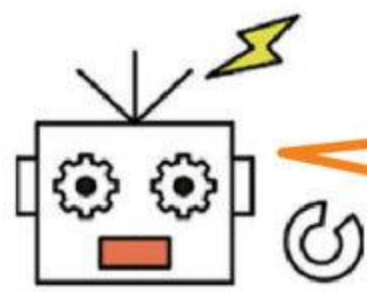
他にも点滅の方向を左右逆転させる、1つ飛ばしで点滅させる、同時に2個ずつ点灯させるなど、さまざまなアレンジが考えられます。自由に取り組みさせてください。街中の電飾看板やネオンサインなどの動きをマネするのもよいでしょう。なお回路上の LED の数を増やしても構いませんが、次回のテーマで LED を8個に増やした課題に取り組みますのでご注意ください。

3. まとめ (目安5分)

今回は LED を使って電子回路やプログラムのルールを学びました。これは、ロボットにおける「動く」部分の一つの例です。

次回は、タクトスイッチというパーツを使って、回路を組み立てます。「感じる」の部分と「考える」の部分を、徹底的に勉強しましょう。これをくり返していくことで、さらにカシコイロボットをつくれるようになります。

では、また次回！



次回はスイッチを使うよ！

《次回必要なもの》

ラジオペンチ	1	USB ケーブル	1	マイコンボード	1	電池ボックス	1
							
スピーカー	1	301 ブレッドボード	1	ジャンパー線	65	抵抗 (100 Ω / 1k Ω / 10k Ω)	各10
							
タクトスイッチ	10	緑色 LED	10	フルカラー LED	1		
							

図 3-0 次回必要なもの

講

- 以下の授業の目標を再確認します。
 - ・電子回路の作り方を学ぶ
 - ・LED を光らせる
 - ・for 文を使ったプログラムを学ぶ
- 今回の授業で学んだ感想や面白かったことなどを、生徒から聞いてみましょう。
- 次回テーマは「電子部品とプログラミング (スイッチ)」であることを告知します。