

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Ⅱ②

(第3回/第4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3回授業日 2024年 月 日

だい かい じゅ ぎょう び
第4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5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Ⅱ②

第3回

でんしかいろ

電子回路とプログラミング
(アナログセンサー)

講師用

目 次

0. 電子回路とプログラミング（アナログセンサー）

0.0. 「電子回路とプログラミング（アナログセンサー）」でやること

0.1. 必要なもの

1. LED の点灯と調光

1.0. LED の点灯

1.1. LED の調光

2. アナログセンサーを応用する

2.0. 可変抵抗ボリュームの接続と実験

2.1. 複数の LED を制御する

2.2. 照度計

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。

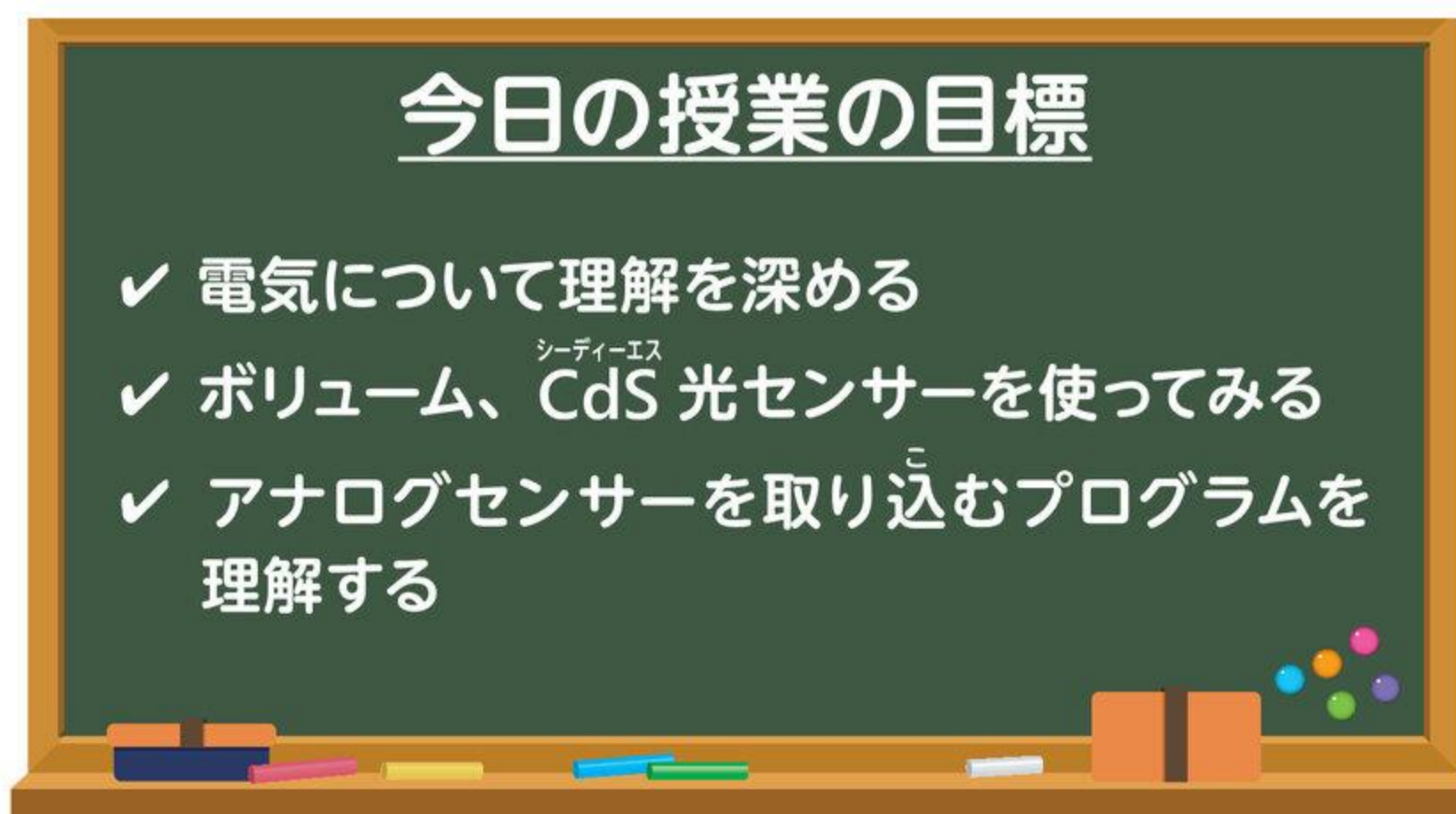
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

（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. ^{でんしかいろ}電子回路とプログラミング（アナログセンサー） （目安10分）

0.0. ^{でんしかいろ}「電子回路とプログラミング（アナログセンサー）」でやること



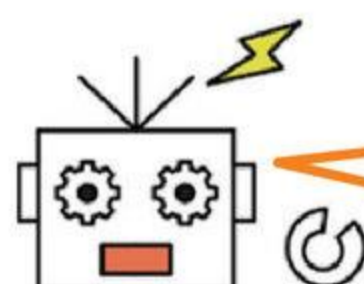
突然ですが、前回の授業で扱ったタクトスイッチあつかを覚えていますか。押したらLEDが付き、離したらLEDが消え…となったあのスイッチです。

みなさんは、タクトスイッチを「オン」と「オフ」の中間にすることはできるでしょうか。おそらくできないと思います。少しずつ力を加えていったとしても、ある瞬間でいきなり「オン」に切り替わってしまいますね。

これは、タクトスイッチが「オン」「オフ」という限られた種類のみを判別する「デジタルセンサー」であるためです。「LEDを点けるか、消すか」だけを判定すればいいときなら便利なのですが、「LEDの明るさを制御する」といった場面ではイマイチ使いづらいですね。

今回の授業では、オン・オフの判定だけでなく「どれくらいの値か」まで見わけることができる「アナログセンサー」を何種類か使います。アナログセンサーなら、LEDの明るさを徐々に変化させたり、1つのセンサーで複数のLEDを別々に点灯させたりといった高度な制御せいぎょができるようになります。

ロボットには欠かせない「センサー」の使い方を、ここでしっかりマスターしましょう！



あいまいな情報も読み取れるゾ。

0.1. 必要なもの

以下のパーツを準備しておきましょう。

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
			
可変抵抗ボリューム 2	タクトスイッチ 10	CdS 光センサー 1	緑色 LED 10
			

図 0-0 必要なもの

1. LEDの点灯と調光（目安25分）

1.0. LEDの点灯

まずは、前回は扱った「デジタルセンサー」であるタクトスイッチのおさらいをしましょう。電源には電池ボックスを使用します。

やってみよう！

電池ボックスとブレッドボード、ジャンパー線、100Ω抵抗を使い、タクトスイッチを押したら緑色LEDが点灯する回路をつくってみよう！

⚠️ 注意！

回路をつくる際には、以下の点に注意しましょう。

- ・電池ボックスのプラス極とマイナス極が直接接触し合うような配線になっていないか？
- ・LEDのアノードとカソードの向きは正しいか？（足の長いほうをプラス極に接続）
- ・回路の中に「100Ω抵抗を通らないと進めない部分」はつくってあるか？

これらは回路のショート（抵抗なしで一気に電流が流れること）などの原因になり、異常な発熱や発煙、各製品の故障につながります。

気付いた時には急いでジャンパー線を引き抜くようにしましょう（乾電池や電子部品は熱くなっている場合があるので触れないように注意しましょう）。

できましたか？

次のような回路をつくれれば、タクトスイッチでLEDのオン・オフができますね！

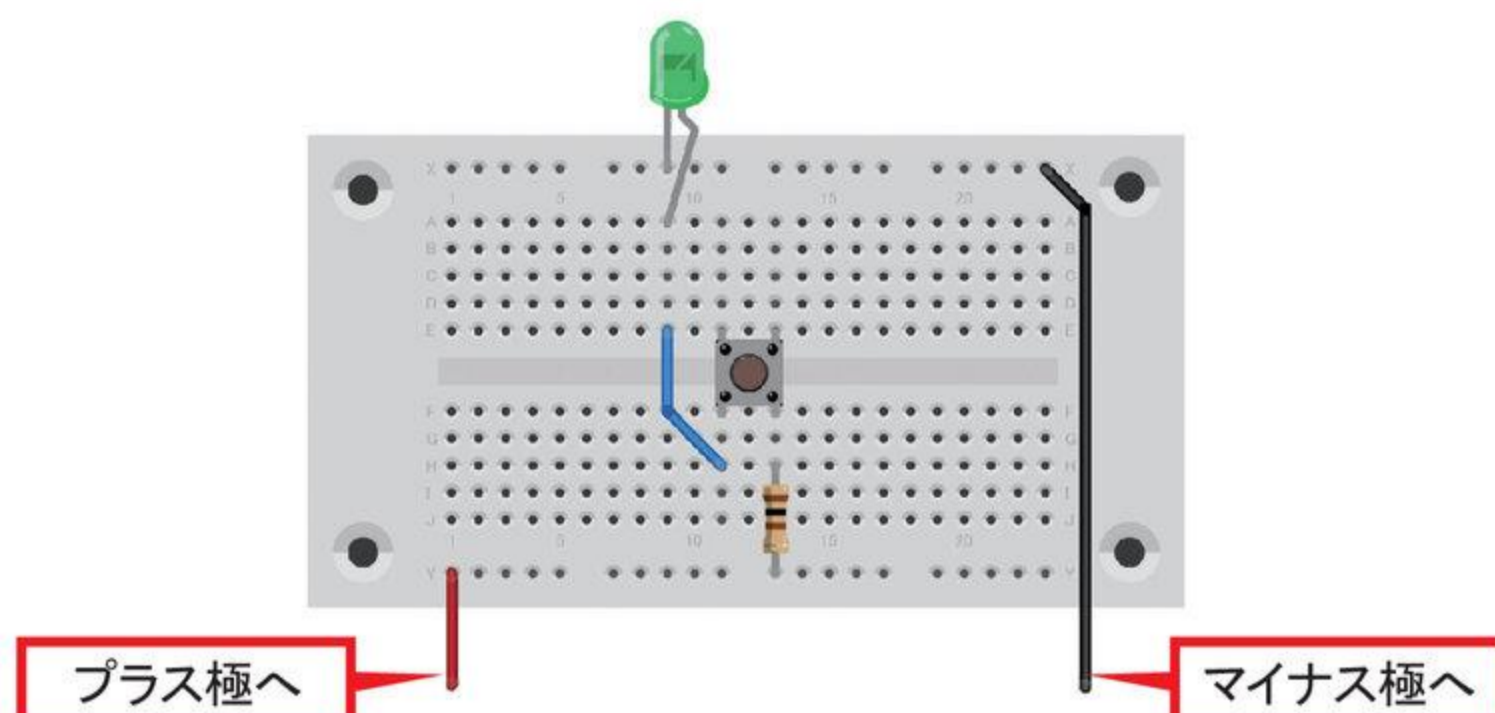


図1-0 タクトスイッチによるLED制御

タクトスイッチはデジタルセンサーなので、電流を「流す」か「流さない」か、つまりLEDを「点灯」か「消灯」かのどちらかにしかできません。

これをアナログセンサーに交換して、どのような動作になるか確認してみましょう。

1.1. LED の調光

1) 可変抵抗ボリューム

先ほどの回路からタクトスイッチを取り外し、かわりに可変抵抗ボリュームを取り付けましょう。

可変抵抗ボリュームは左右に金属の「つまみ」がついているので、これがブレッドボード中心の溝に入るようにさし込んでください。

3つある端子のうち、真ん中の端子を $100\ \Omega$ 抵抗に、向かって左側の端子を緑色 LED に接続するようにします。

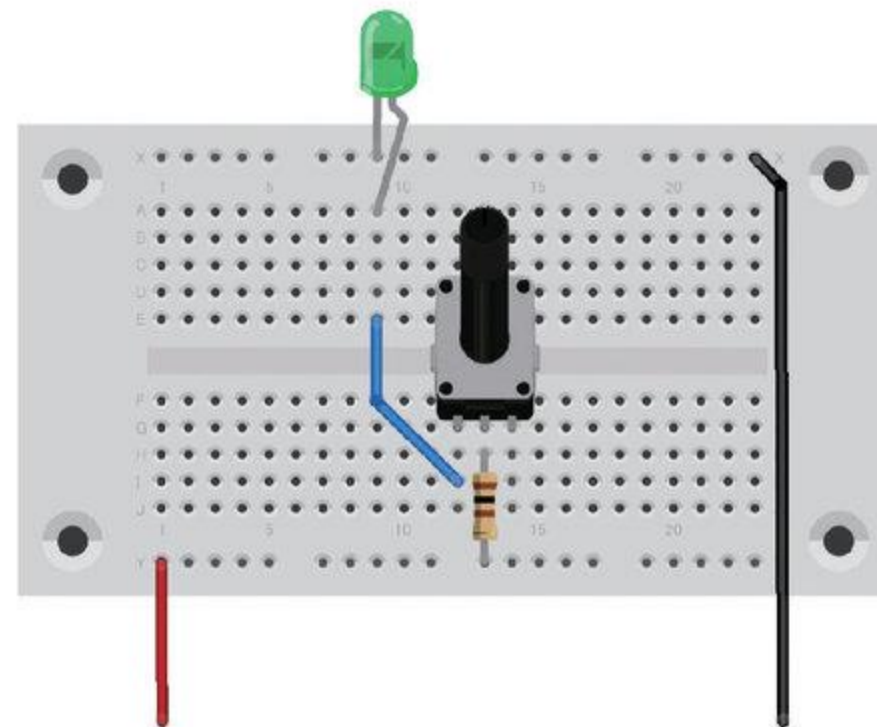


図 1-1 可変抵抗ボリュームによる LED 制御

やってみよう!

可変抵抗ボリュームのつまみを回したとき、LED がどのようになるか確かめてみよう!

つまみを時計回りに回すと、緑色 LED の光が (明るく ・ 暗く) なる。

つまり、抵抗の値が (大きく ・ 小さく) になっていくことがわかる。

つまみを反時計回りに回すと、緑色 LED の光が (明るく ・ 暗く) なる。

つまり、抵抗の値が (大きく ・ 小さく) になっていくことがわかる。

「可変」とは文字通り、変化させることが可能という意味です。つまり可変抵抗ボリュームはこれまで使っていた抵抗と違い、抵抗の値を自由に変化させることができます。抵抗の値が変化すれば回路を流れる電流も変化するので、LED の明るさも変化したというわけです。

前回は LED の明るさを変化させるのに `analogWrite` という命令を使いましたが、可変抵抗ボリュームを使うことでプログラム側でなく回路側からも LED の制御ができるようになったのです。

2) CdS 光センサー

先ほどは可変抵抗ボリュームをアナログセンサーとして使いましたが、「状況に応じて少しずつ抵抗の値が変化するという性質を持っているならば他の電子部品でもアナログセンサーになります。

たとえば、可変抵抗ボリュームを取り外し、「CdS 光センサー」を接続してみましょう。

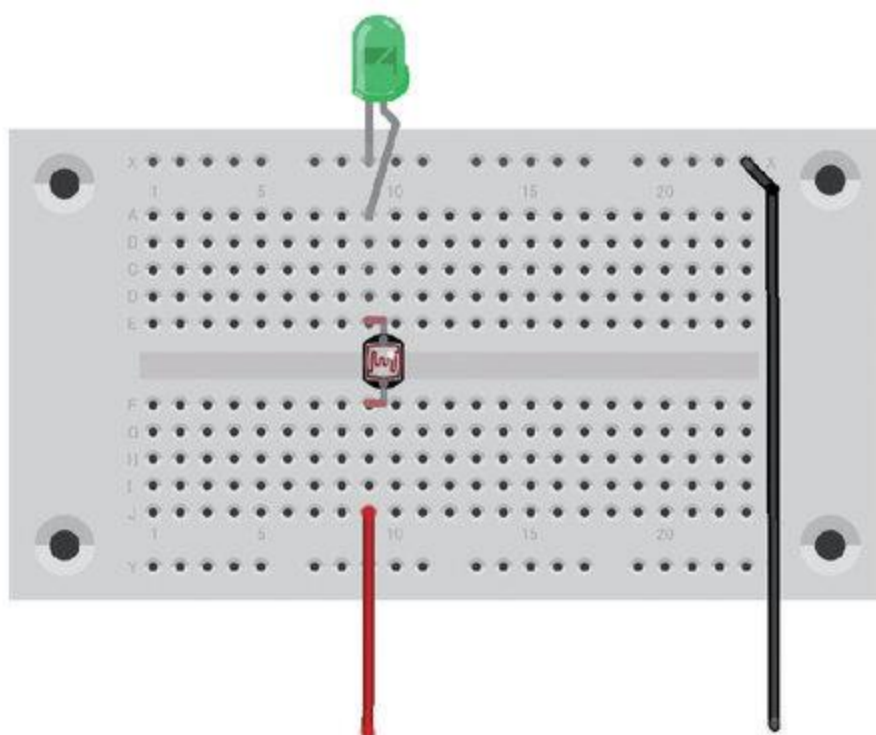


図 1-2 CdS 光センサーによる LED 制御

CdS 光センサーは「硫化カドミウム」という物質で作られていて、「光の量に応じて抵抗の値が変化するという性質があります。ためしにセンサー部分を指などでおおってみると、LED が暗くなりますね。

つまり、光が弱くなるほど抵抗の値が大きくなるのがわかります。

光の強さしだいで少しずつ抵抗の値を変化させられるので、これも立派なアナログセンサーです。

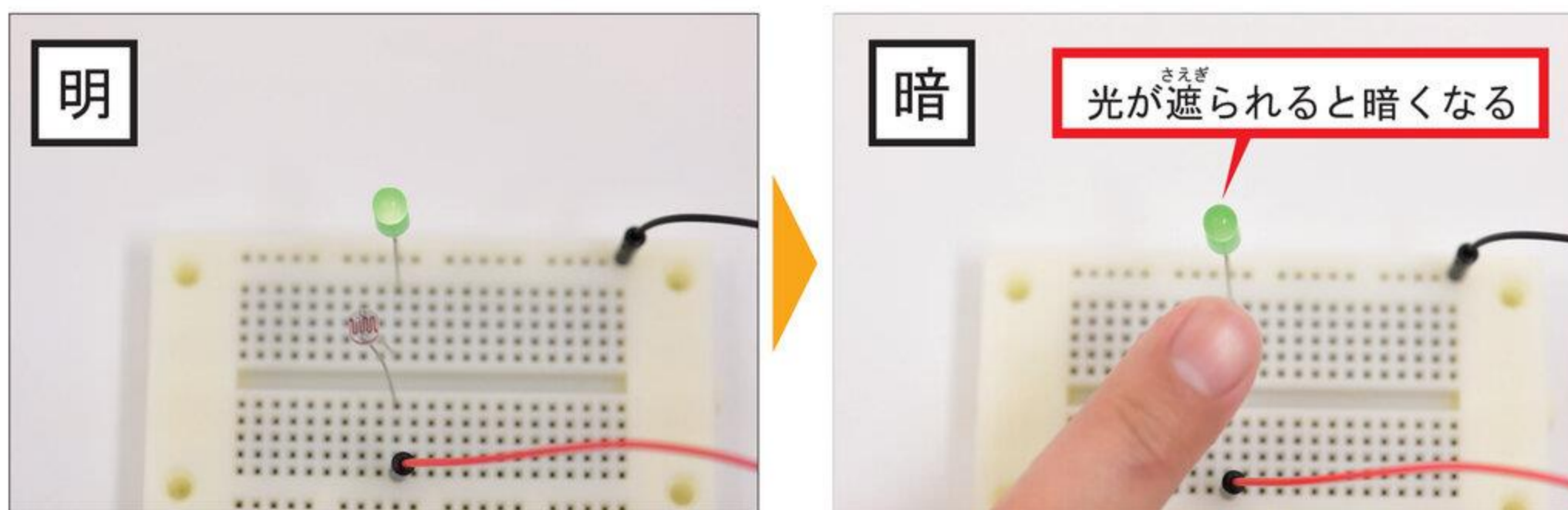


図 1-3 CdS 光センサーの明暗

もちろん人の手で制御するならば、可変抵抗ボリュームの方が操作しやすいですね。ただ CdS 光センサーは「人の手で動かす必要がない」という利点があります。みなさんの住んでいる地域には、日が暮れてくると自動的に点灯する街灯はありますか。この街灯にも、CdS 光センサーを使っているものが多く存在します。

2. アナログセンサーを応用する (目安 65 分)

2.0. かへんていこう 可変抵抗ボリュームの接続と実験

1) 実験 1

では、電源をマイコンボードに交換しましょう。

アナログセンサーを使ったプログラミングの前に、まずは前回使ったデジタルセンサー、つまりタクトスイッチのおさらいをします。

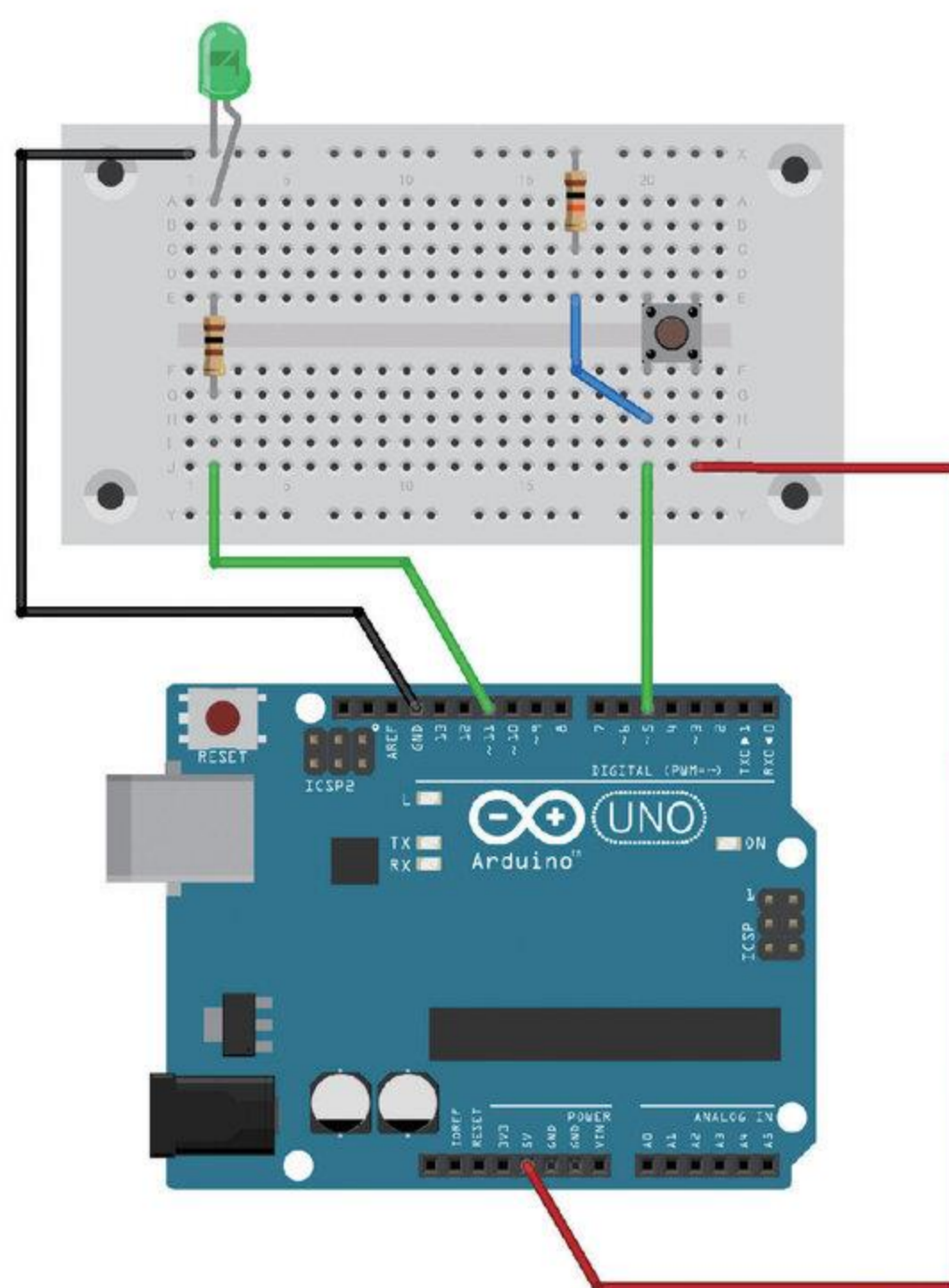


図 2-0 タクトスイッチを使ったマイコンボード回路かいろ

図 2-0 の回路かいろであれば、タクトスイッチを押しているときだけ、マイコンボードの 5 番ピンに電流が流れます。また、11 番ピンから電流を流せば、緑色 LED が点灯しますね。よって「5 番ピンに電流が流れている (HIGH である) ときだけ、11 番ピンから電流を流す」という条件分岐ぶんきでプログラムをかけば、タクトスイッチで LED を制御するプログラムせいぎよになります。

ステップアップ

前のページにかかれた通りの動作になるよう、プログラムをかいてみよう！



ヒント

`void setup()` に 5 番ピンと 11 番ピンの入力・出力モード^{せんげん}宣言を、
`void loop()` に if 文をかけばいいね！ それ以外の部分は不要だよ！

かけましたか？

次のようなプログラムをかけば、思い通りの動きになりますね。

```
void setup(){
  pinMode(5, INPUT);
  pinMode(11, OUTPUT);
}

void loop(){
  if(digitalRead(5) == HIGH){
    digitalWrite(11, HIGH);
  }
  else{
    digitalWrite(11, LOW);
  }
}
```

タクトスイッチはデジタルセンサーなので、`HIGH` と `LOW` という 2 通りの値しかとりません。よって、`digitalRead(5);` という命令で、5 番ピンのタクトスイッチが `HIGH` なのか `LOW` なのかを確認しています。

`HIGH` のとき、つまりタクトスイッチが押されているとき、11 番ピンの緑色 LED に対して `digitalWrite(11, HIGH);` と命令しています。11 番ピンに電流を流す命令でしたね。`digitalWrite` も `HIGH` か `LOW` かの 2 通りでしか命令できないので、点灯か消灯のどちらかしか選べませんでした。

LEDの明るさを調整したければ、以下のようにかきかえればよいですね。

```
if(digitalRead(5) == HIGH){  
    digitalWrite(11, HIGH);  
}
```



```
if(digitalRead(5) == HIGH){  
    analogWrite(11, 127);  
}
```

`analogWrite(11, 127);` としたことで、明るさを0～255の256段階で調整できるようになりました。明るさ127であれば、最大値のほぼ半分です。

では、タクトスイッチを可変抵抗ボリュームにつけかえましょう。次の図の通りに配線してください。

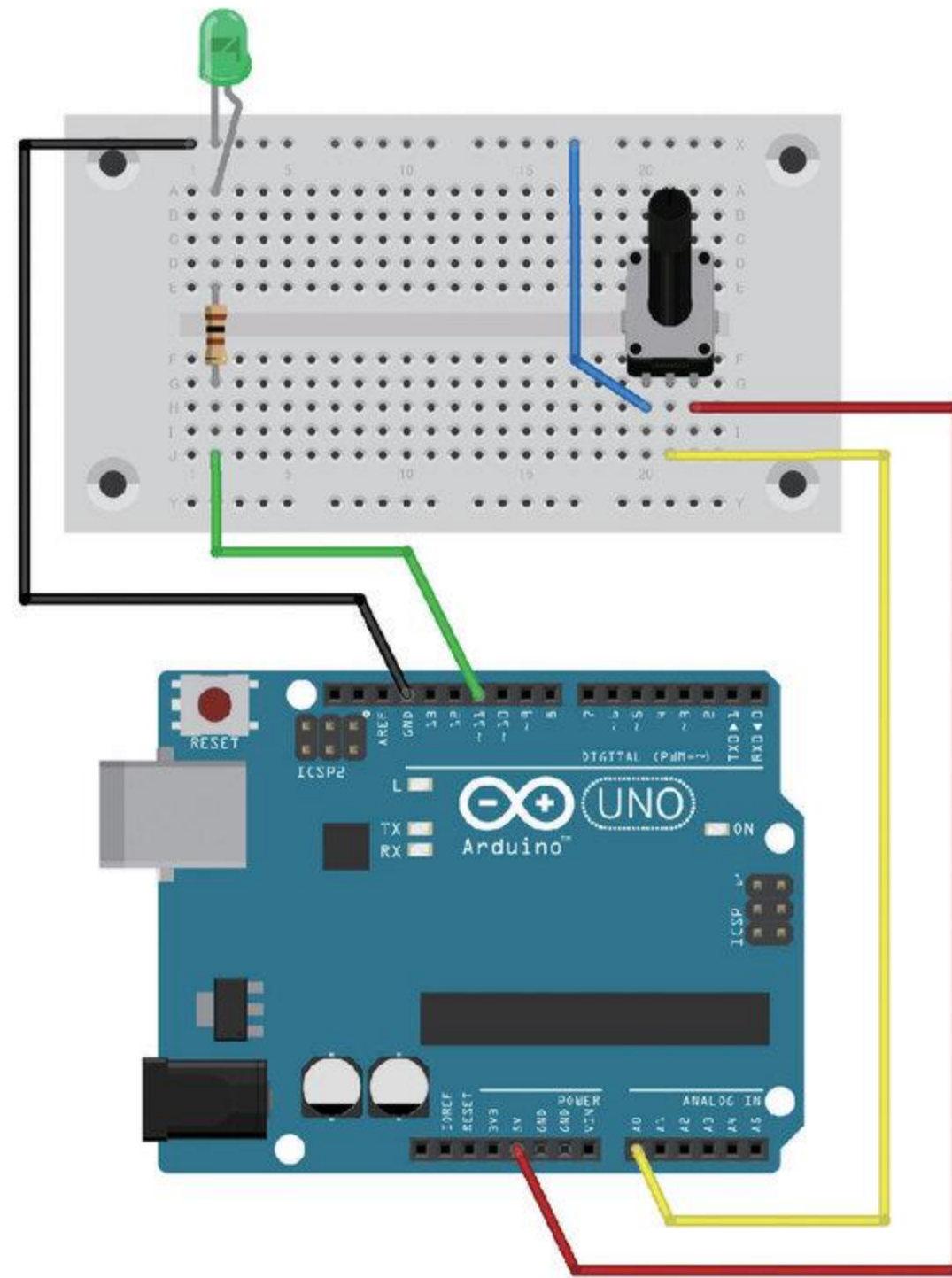


図 2-1 可変抵抗ボリュームを使ったマイコンボード回路

今回はセンサーからの電流が5番ピンではなく、[A0]ピンに流れ込みますね。プログラムは以下のものを使います。

プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA3 > LED_Vol1

実行結果：緑色LEDが点滅する。可変抵抗ボリュームのつまみを回すと、点滅速度が変化する。

点灯・消灯の時間が変化しているということは、[delay]の数値にヒントが隠れていそうです。

プログラムを見てみましょう。

プログラム「LED_Vol1」より抜粋

```
digitalWrite(11, HIGH); // 11番のLEDを点灯
delay(val);
digitalWrite(11, LOW); // 11番のLEDを消灯
delay(val);
```


点滅時間に `val` という変数が使われています。変数 `val` の値はどのように決められているのでしょうか。

少し上の行にその答えがあります。

プログラム「LED_Vol1」より抜粋

```
val = analogRead(A0); // A0番に接続したボリュームを測定
```

命令「analogRead」

実行内容：指定のピンの電圧を 0 ～ 1023 の 1024 段階でチェックする

使い方：`val = analogRead(A0);`

// A0 ピンの電圧を 0 ～ 1023 の範囲で数値化し、変数 `val` に入れる

アナログセンサーからマイコンボードに電流が流れるとき、その電圧はセンサーの状況に応じ、0V ～ 5V の範囲内で少しずつ変化します。`analogRead` を使えば、これを 1024 段階にわけて検知することができます。ちょうど中間(つまみに「カチッ」という感触があったとき)だと、0 ～ 1023 の中間である 511 程度の値になります。

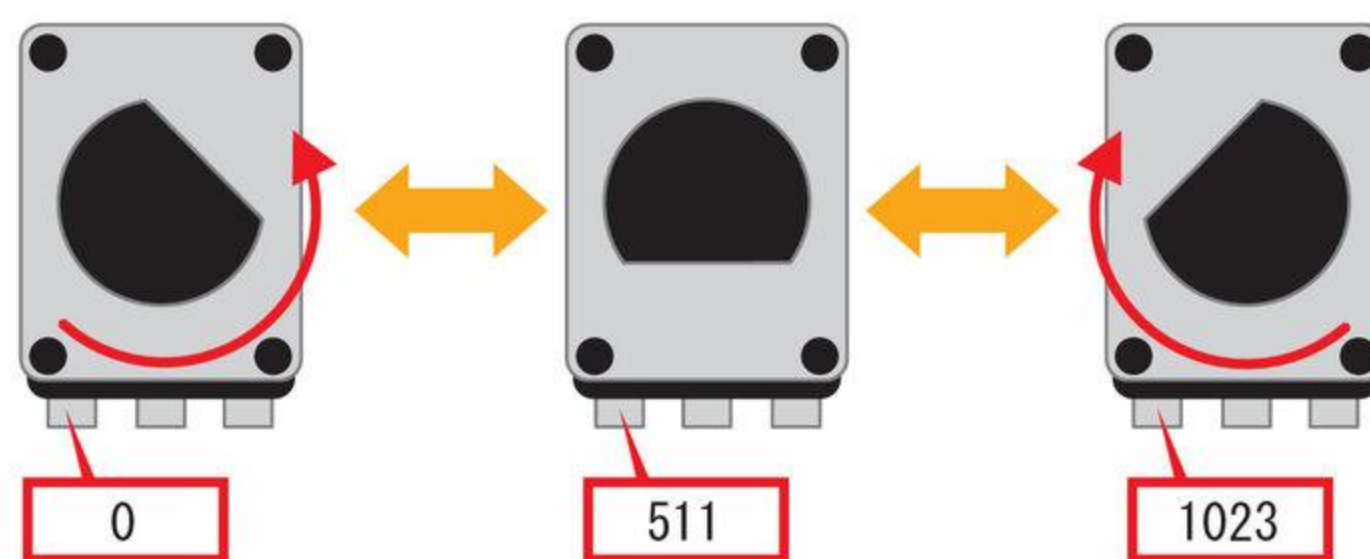


図 2-2 可変抵抗ボリュームの値の範囲

前回使った `digitalRead` は「電圧がかかっているか、いないか」しか判別できません。アナログセンサーを使う場合、`analogRead` のほうが向いていますね。

やってみよう！

プログラム「LED_Vol1」をかきかえ、緑色 LED が常に点灯していて、ボリュームでその明るさを制御できるようにしてみよう！

ヒント

可変抵抗ボリュームの値は最大 1023 だけど、`analogWrite` で指定できる LED の明るさは最大 255 だったね！

`val` の値をそのまま明るさに使ってしまうとすぐ明るさが最大になってしまうから、少し工夫をしてみよう！

ステップアップ

さらにプログラムをかきかえ、先ほどとは逆に、反時計回りにつまみを回したときLEDが明るくなるようにしてみよう！

「やってみよう」では、`vol`の値を4でわることで、0～1023の範囲を0～255に縮めています。

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA3 > LED_Vol2

講

「ステップアップ」では、変数`val`が0のとき明るさが255、255のとき明るさが0になればよいので、 $val \div 4$ の値を255から引くことで条件を満たしています。

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA3 > LED_Vol2_challenge

2) 実験 2

変数 `val` をうまく使うことで、LED 以外にも様々な電子部品を制御できます。
先ほどの回路に、スピーカーを追加してみましょう。

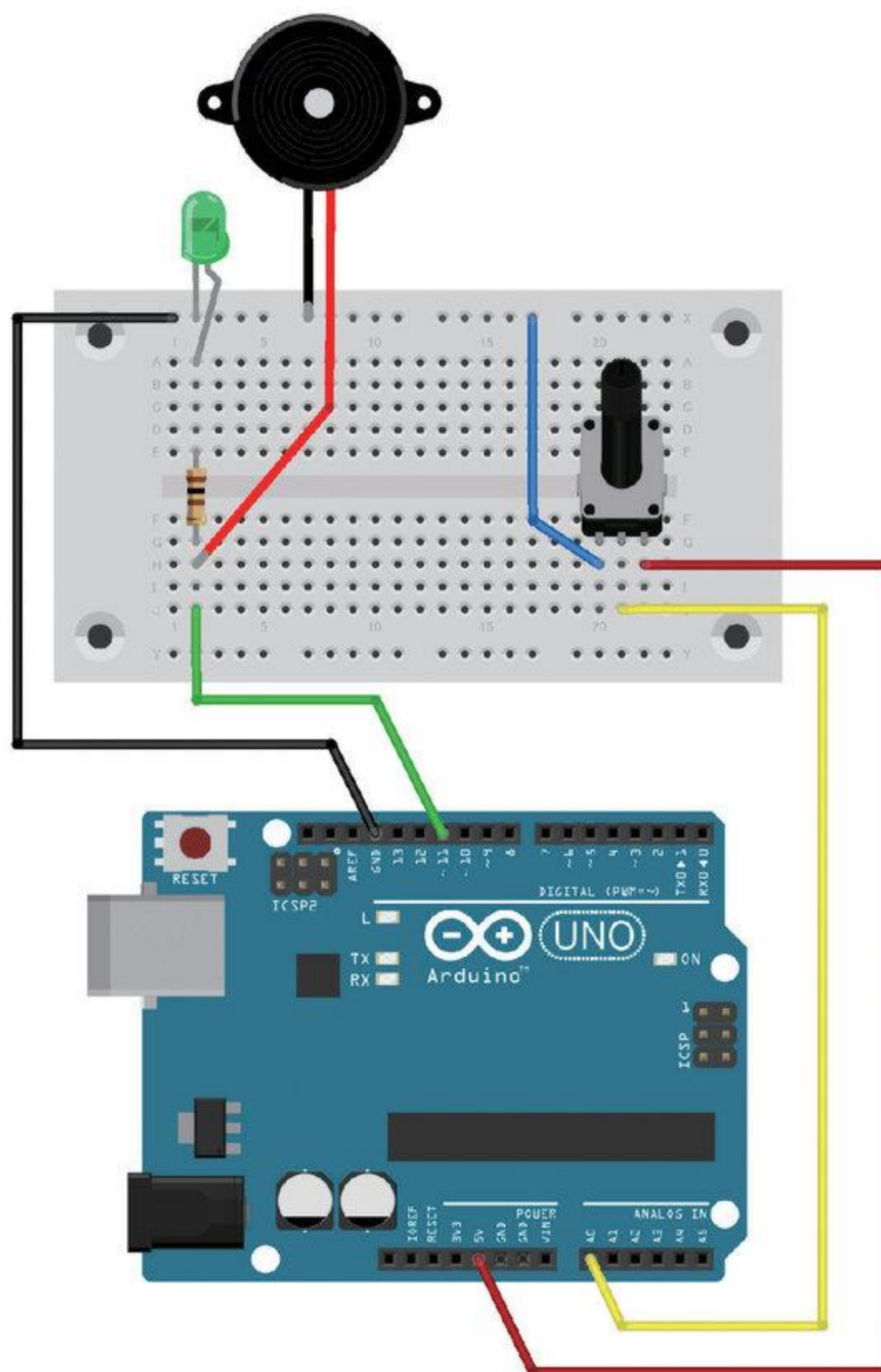


図 2-3 スピーカーを追加した回路

やってみよう!

プログラムをさらにかきかえ、LED ではなくスピーカーの音の高さを調整できるようにしてみよう!

💡 ヒント

スピーカーから音を出す命令は `tone(ピン番号, 周波数);` だったね!

講

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA3 > LED_Vol_Spk

周波数を `val * 2` などに変更すると、更に値が大きくなり音が高くなります。

2.1. 複数のLEDを制御する

回路を図2-4のように作りかえ、緑色LEDを5個に増やしてみましょう。

アナログセンサーは `analogRead` を使えば1024通りもの結果が得られるので、複数のLEDを別々に操作することもできます。先ほどのプログラム「LED_Vol1」はLED1つの点滅速度を可変抵抗ボリュームで調整していましたが、これをLED5つで行えるようにしてみます。

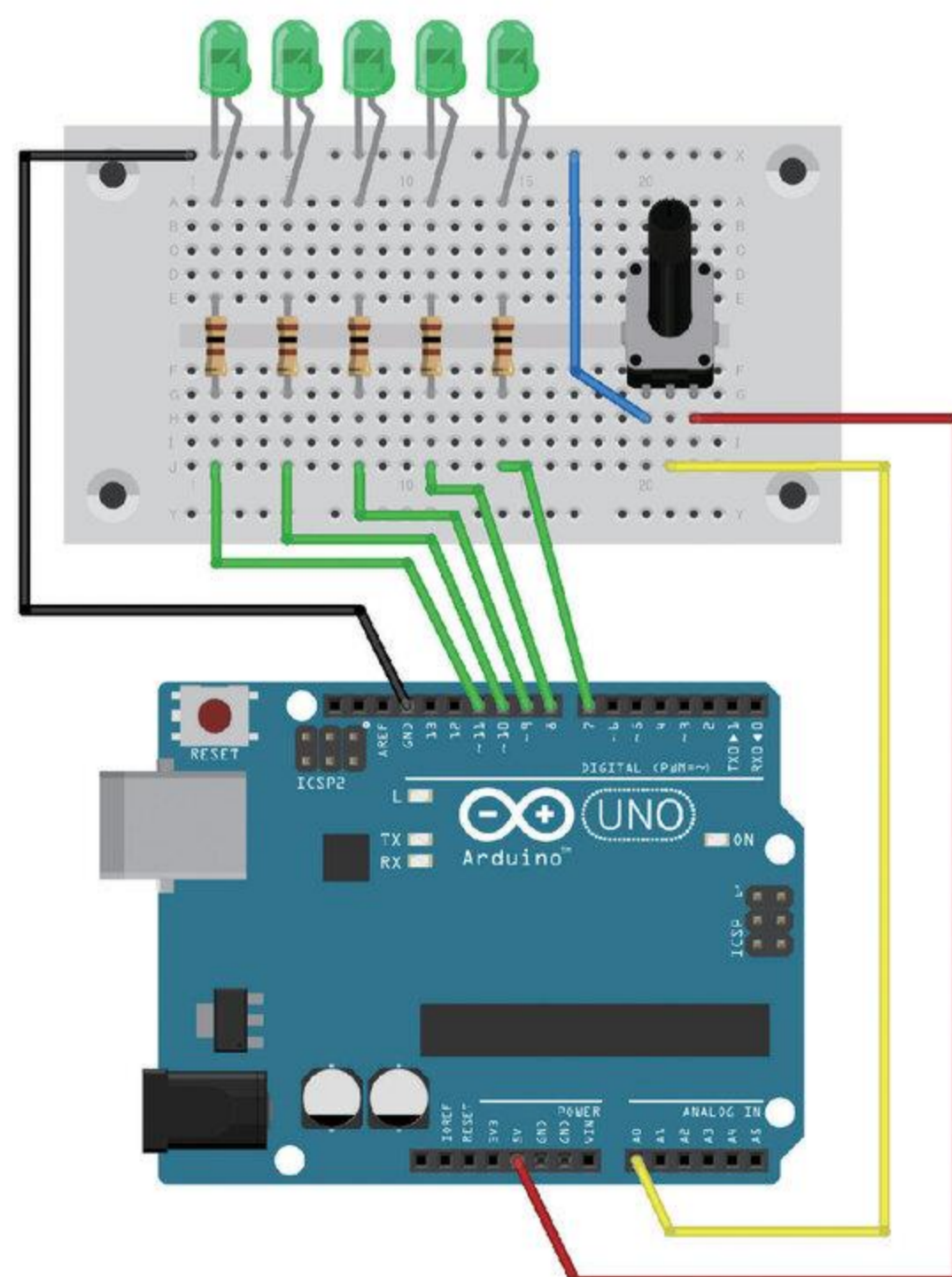


図2-4 緑色LED5個の制御

やってみよう！

プログラム「LED_Vol1」をかきかえ、5つのLEDが右から順に点滅していくようなプログラムにしてみよう！

点滅速度は可変抵抗ボリュームで調節できるようにしてね。

講

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA3 > LED5_Vol1

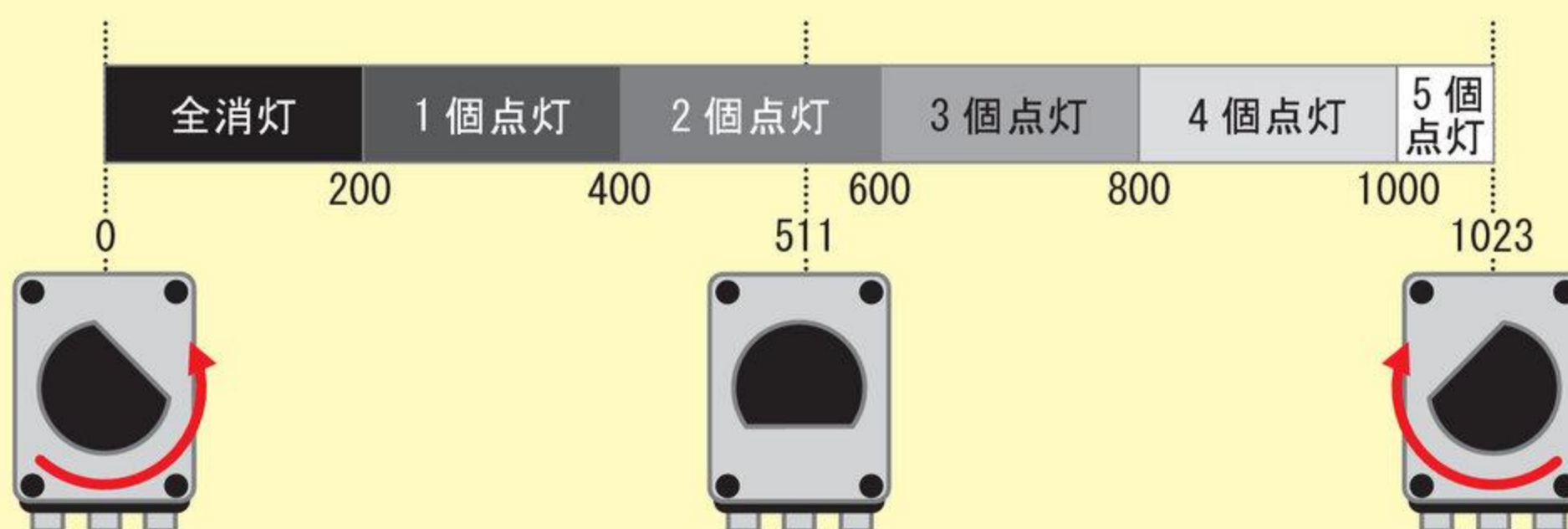
ピン番号が変化していくだけで、命令内容は同じものが連続します。

`digitalWrite` のピン番号を変数にして for 文でまとめてしまうことで、点灯・消灯命令を1行ずつかくだけで済むようになります。

続いて、^{かへんていこう}可変抵抗ボリュームの値を「可視化」(目に見える形にすること)してみましょう。マトリクスLEDなどに現在の値をそのまま表示させるのが一番正確でわかりやすいですが、今回は5つの緑色LEDをうまく活用してみます。

**POINT**

- ・反時計回りに目いっぱい回したとき (値が0のとき) は、LEDは一つも点灯しない
- ・値が200増えるごとに、LEDが左から1つずつ点灯していく。
- ・時計回りに目いっぱい回したとき (値が1023のとき) は、LEDがすべて点灯している。



フローチャートをかいてみると、少しわかりやすいですね。

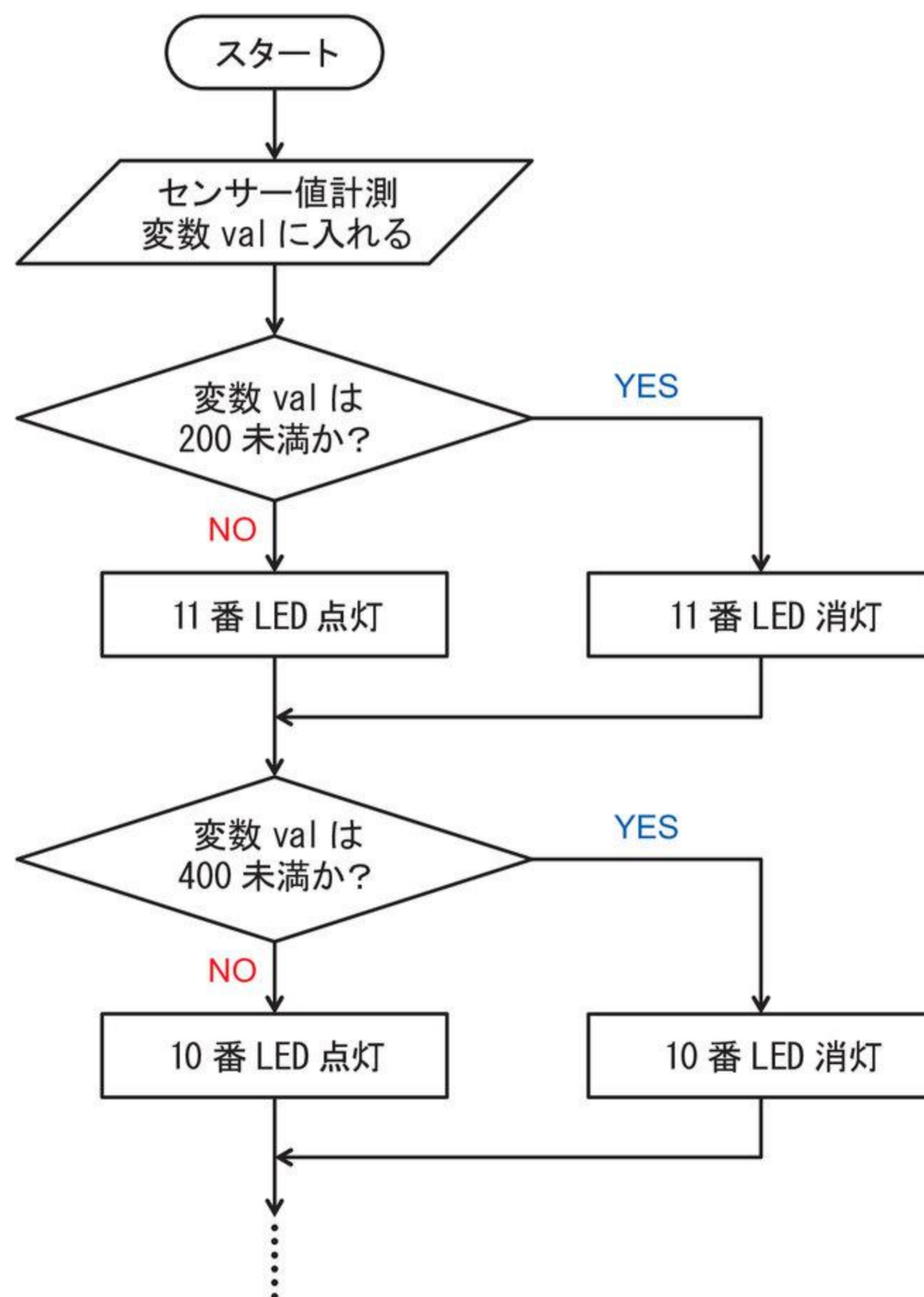


図 2-5 緑色 LED5 個の制御^{せいぎよ}フローチャート

ステップアップ

プログラム「LED_Vol1」をかきかえ、**図 2-5**のような動作にしてみよう！

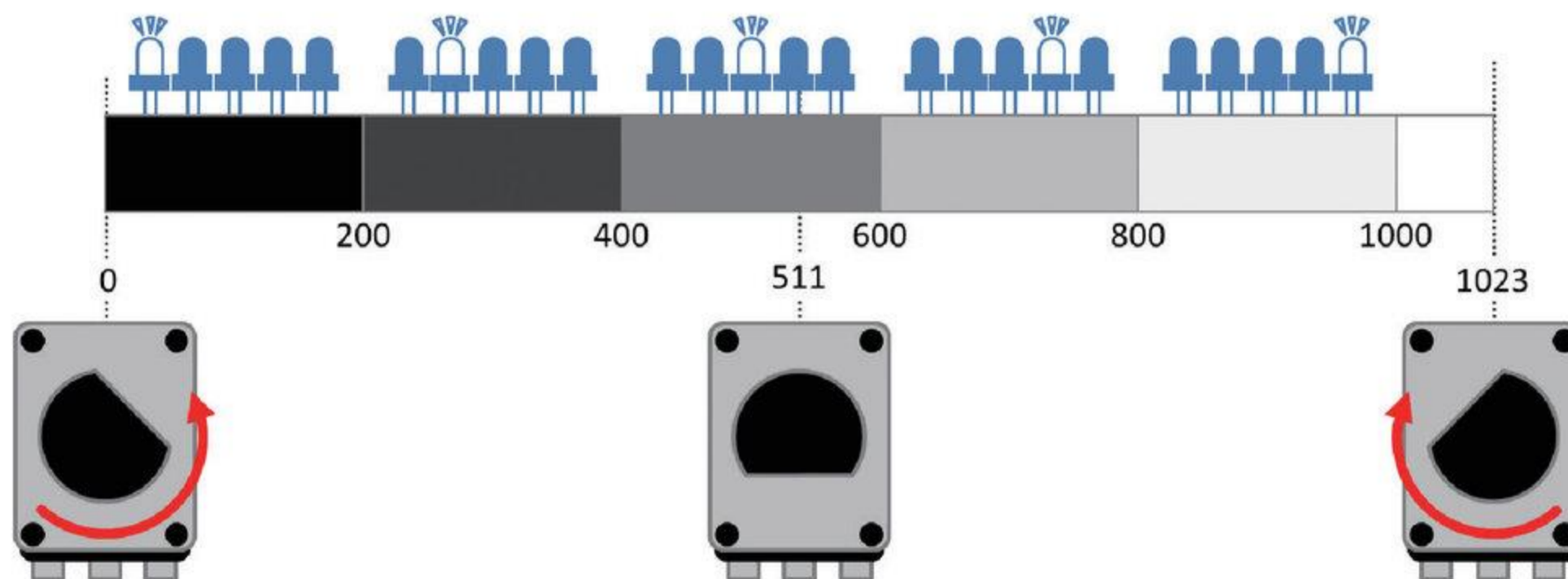
講

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA3 > LED5_Vol1_bar_answer

チャレンジ課題

さらにプログラムをかきかえ、以下の図のような動作にしてみよう！

 ヒント

ここから一気に分岐が複雑になるよ。迷ったらまずはフローチャートを描いて頭を整理しよう！

ちなみに「もし変数 `val` が 100 より大きくて、かつ 200 より小さければ」のように、複数の条件で `if` をかきたければ、`if(val > 100 && val < 200)` と 2 つの条件式を `&&` で結ぼう！

講

解答例は以下のプログラムです。

[RoboticsProfessorCourse2 > MagicItemA3 > LED5_Vol1_move_answer](#)

2.2. 照度計

プログラムはそのまま、アナログセンサーを可変抵抗ボリュームから CdS 光センサーにつけかえてみましょう。

光の強さによって点灯する LED の数が増えるため、明るさ計測器（照度計）として使うことができます。

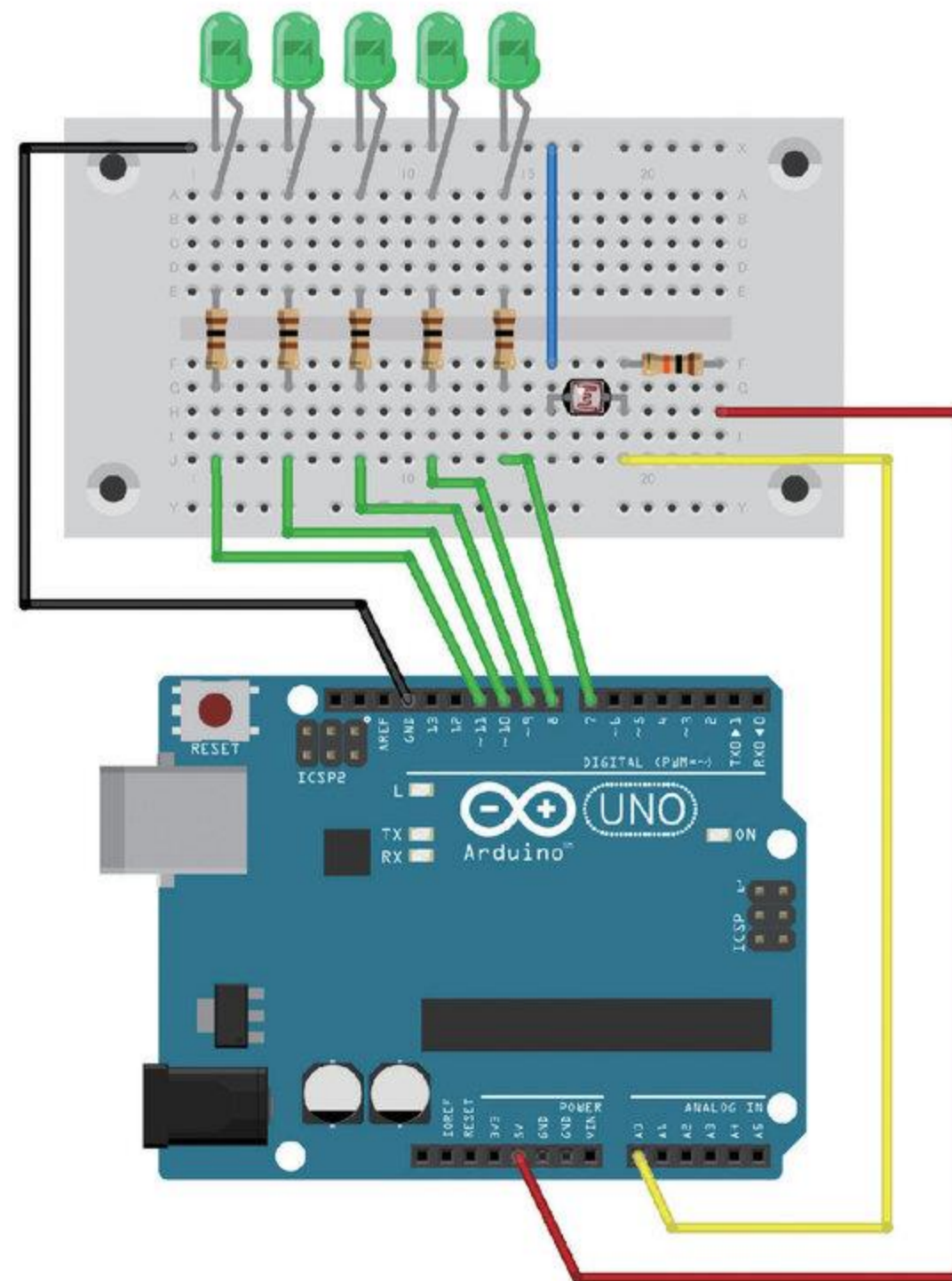


図 2-6 照度計

他にも、超音波距離センサーも物体との距離で値が変化するアナログセンサーです。障害物に近いときは全速力で下がり、遠いときはゆっくり下がる…といった、まるで生きているかのような動きをするロボットをつくることもできそうですね。

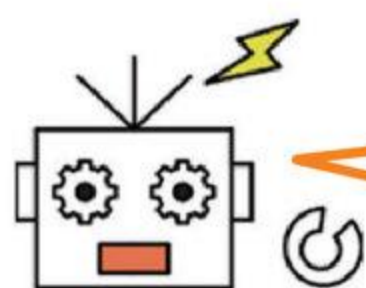
同じ `analogRead` を使っているのに、センサーの種類や回路の組みかた、ロボットの形を工夫することで色々な活用ができそうですね！

これがプログラミングの、そしてロボットづくりの面白いところです。みなさんも生活の中で「ちょっと不便だな」「もっと便利にできそうだな」という場面があったら「こういうセンサーでこういう分岐をつくれれば改善できそうなのに…」というアイデアを考えてみるクセをつけておくと、将来役立つかもしれませんね！

3. まとめ (目安5分)

プログラムはくり返し使うことによって覚えていきます。

次回は、7セグメントLEDという表示機ひょうじきを使います。今回まではLEDは1個ずつしか点灯しませんでした。LEDをたくさん配置してあるパーツを使って、センサーの値を表示したりしてみましょ。今回わかりにくかったアナログセンサーも、もっと理解が深まるかと思ひます。ではまた！



センサーのあいまいな値をロボットの動きにイカセそうだね。

講

- 以下の授業の目標を再確認します。
 - ・電気について理解を深める
 - ・ボリューム、CdS光センサーを使ってみる
 - ・アナログセンサーを取り込むプログラムを理解する
- 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。
- 次回テーマは「電子回路とプログラミング (7セグメントLED)」であることを告知します。

《次回必要なもの》

次回は、以下のパーツを持ってきてください。



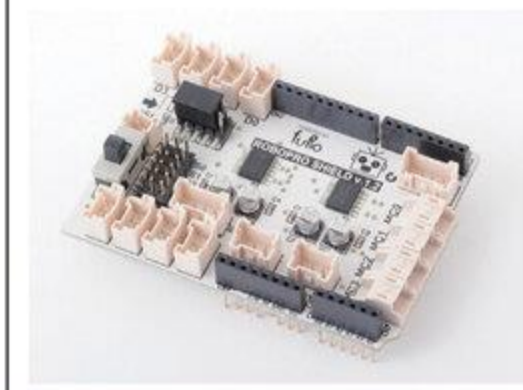
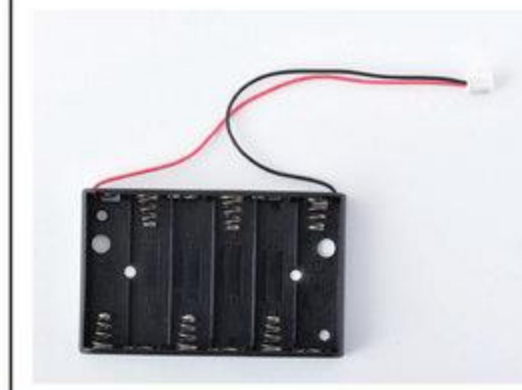





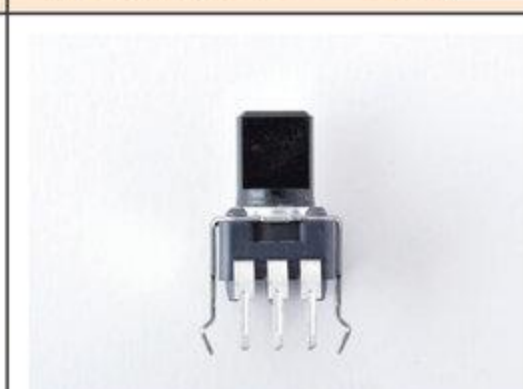
USB ケーブル 1	マイコンボード 1	ロボプロシールド 1	電池ボックス 1
			
マトリクスLEDシールド 1	超音波距離センサー 1	センサーケーブル 1	301ブレッドボード 1
			
ジャンパー線 65	7セグメントLED 1	可変抵抗ボリューム 2	
			

図 3-0 次回必要なもの