

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I-1①

第2回

タッチセンサーとスピーカーで遊ぶ

講師用



# 目 次

## 0. タッチセンサーとスピーカーで遊ぶ

0.0. 「タッチセンサーとスピーカーで遊ぶ」でやること

0.1. 必要なもの

## 1. タッチセンサーを使ってみよう

1.0. タッチセンサーの取り付け

1.1. タッチセンサーの動作確認

1.2. LED カウントを操作してみよう

1.3. 変数とは

1.4. 変数と数式の理解を深めよう

## 2. スピーカーを使ってみよう

2.0. スピーカーの取り付け

2.1. if文を使ってみよう

## 3. タッチセンサーとスピーカーで遊ぼう

3.0. タッチセンサーを使ってキャラクターを動かす

3.1. スピーカーで遊ぶ

## 4. 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)

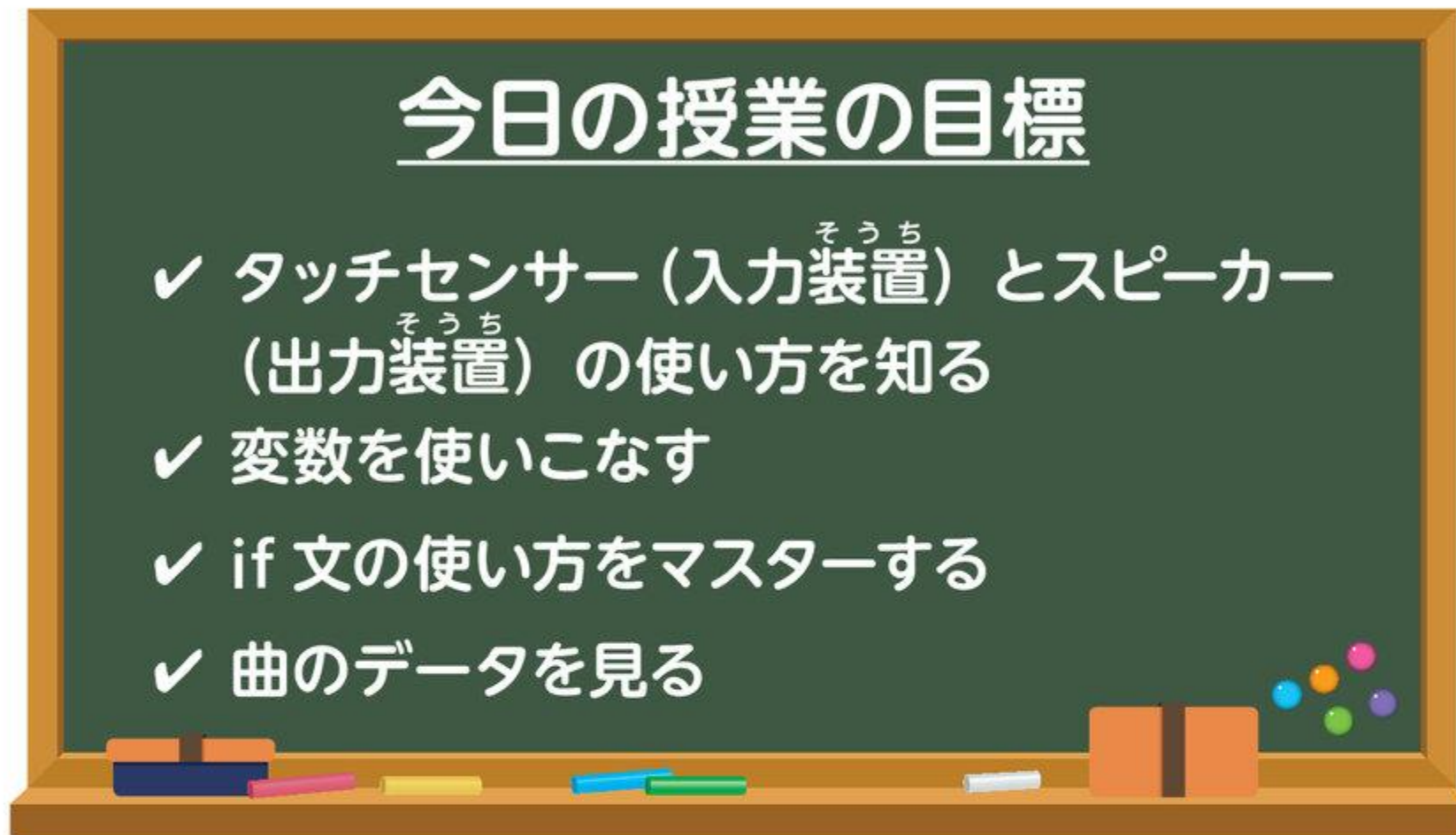
目安時間は授業時間 120 分のうち、休憩 10 分程度を取ることを想定しています。  
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。





## 0. タッチセンサーとスピーカーで遊ぶ (目安5分)

### 0.0. 「タッチセンサーとスピーカーで遊ぶ」でやること



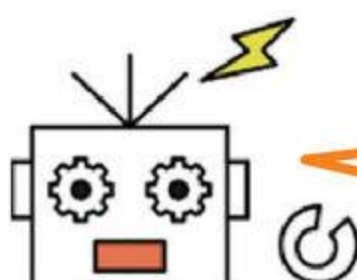
前回は、マトリクス LED という表示板を使って、プログラムの基本をマスターしました。プログラムを操作することで、思うように LED を点滅させたり移動させたり、好きな絵を表示したりできるようになりましたね。

マトリクス LED は、コンピューターで計算した結果を目に見える形で出力する装置です。今回は、コンピューターの出力装置としてもう一つ、スピーカーを用意します。スピーカーは、コンピューターで計算した結果を、耳に聞こえる形、つまり音で出力します。授業で使うものは、ブープ音だけしか鳴らせないスピーカーですが、これがすごいものになると、映画館やコンサート会場のスピーカーになります。今回行うのは、その初歩ということです。さらに今回は、コンピューターへの入力装置としてタッチセンサーを使います。タッチセンサーを使うことで、マトリクス LED 上の数字を変えたり、音楽をつくることも可能です。また、これから高度なプログラムをつくるうえで欠かせない「変数」の考え方、どんなプログラムでも必要になる命令文「if」に関してマスターしていきます。

今回はタッチセンサーとスピーカーを使って、好きな絵を表示したり、好きな音楽を演奏したりしてみましょ。ロボットが登場するときに、カッコいい音楽がついていたらいいと思いませんか？ シビれる音楽で、見ている人たちをビックリさせましょ！

講

ブープ音とは、単一の波形で構成される音で、単純な電子回路で合成できることから、洗濯機や電子レンジなど、さまざまな機器に組み込まれています。



音楽とは音を楽しむという意味なのだよ、諸君！



## 0.1. 必要なもの

以下のパーツを準備しておきましょう。








|  |   |   |   |  |   |   |   |
|--|---|---|---|--|---|---|---|
| USB ケーブル   | 1 | マイコンボード   | 1 | ロボプロシールド   | 1 | タッチセンサー   | 2 |
|   |   |   |   |   |   |  |   |
| マトリクス LED シールド   | 1 | マトリクス LED   | 1 | スピーカー  | 1 |   |   |
|  |   |  |   |  |   |   |   |

図 0-0 必要なもの

なお、前回使用したマイコンボード、ロボプロシールド、マトリクス LED シールド、マトリクス LED の順番に積み重ねていったユニットを今回も使います。

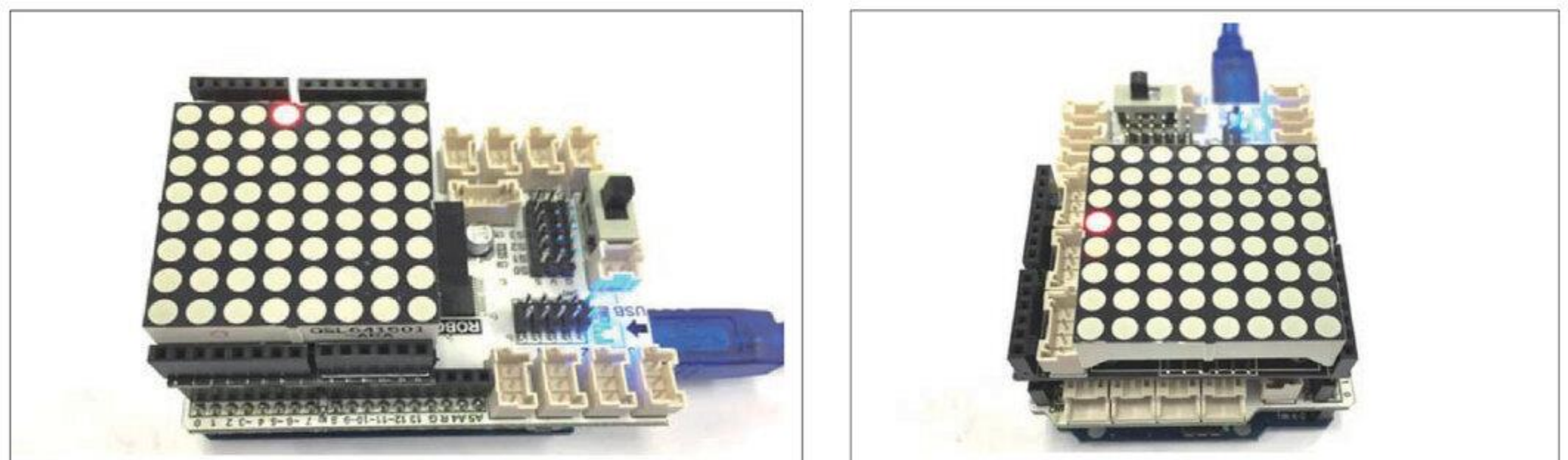


図 0-1 前回組み立てたもの

講

タッチセンサーは、タッチセンサー用ケーブルとスイッチが別々になっています。ケーブルのコネクター部分をスイッチに接続します。

スピーカーは無理に引っ張ったり、負荷がかかると断線しやすいパーツです。取扱いには必ずコネクター部分を持ってロボプロシールドへぬき差しするように指導してください。



# 1. タッチセンサーを使ってみよう（目安 45 分）

## 1.0. タッチセンサーの取り付け

ロボプロシールドにタッチセンサー2個を接続し、プログラムを書き込んで動作の確認をします。図1-0のように、タッチセンサーのケーブルをロボプロシールドの[D0]と[D1]のコネクターにしっかりとさし込みます。

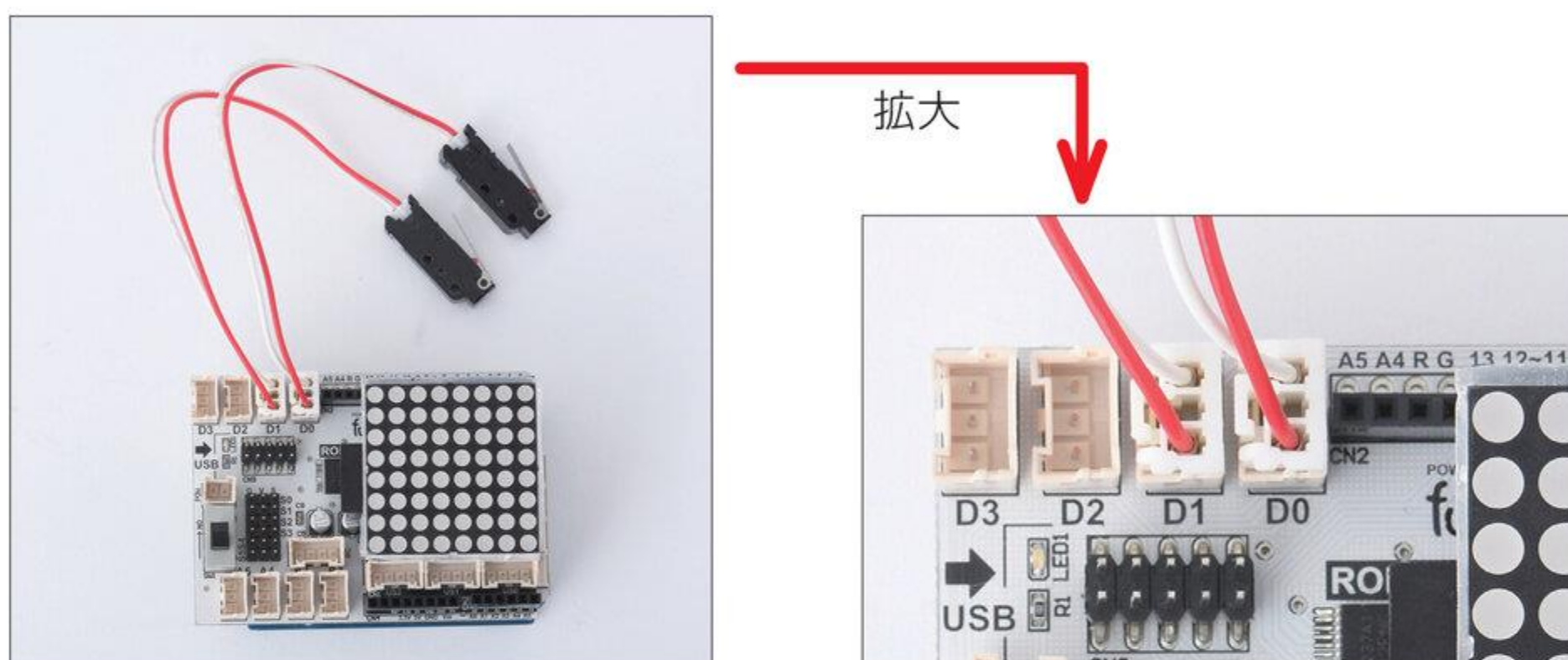


図 1-0 マトリクスLEDシールドとタッチセンサーの接続完成図（全体図と拡大図）

ケーブルをぬく場合は、ケーブルの爪<sup>つめ</sup>の部分をおさえながら引きぬきましょう。無理に引っ張るのは、ケーブルを痛めるのでやめましょう。

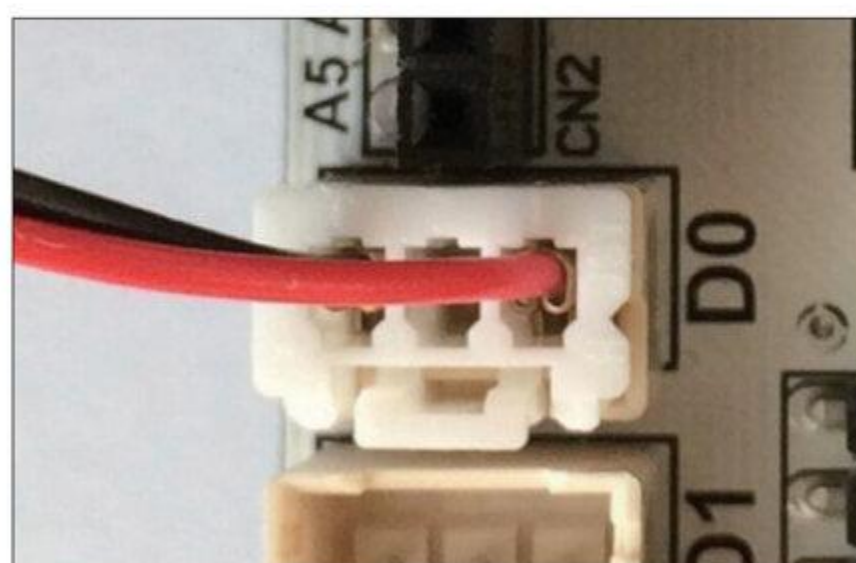


図 1-1 ケーブルの爪<sup>つめ</sup>の部分



## 1.1. タッチセンサーの動作確認

タッチセンサーの接続ができたなら、動作確認をしましょう。  
以下のプログラムを実行してください。

**講** パソコンに接続したときは毎回ポート番号 (COM) を必ず確認させましょう。

### プログラムの書き込み

RoboticsProfessorCourse1 > PreCourse > Switch0

実行結果：[D0] につながったタッチセンサーのスイッチ部の金属バーを押すと、「カチッ」と音が鳴ってスイッチがオンになり、マイコンボードの 13 番 LED が点灯します。

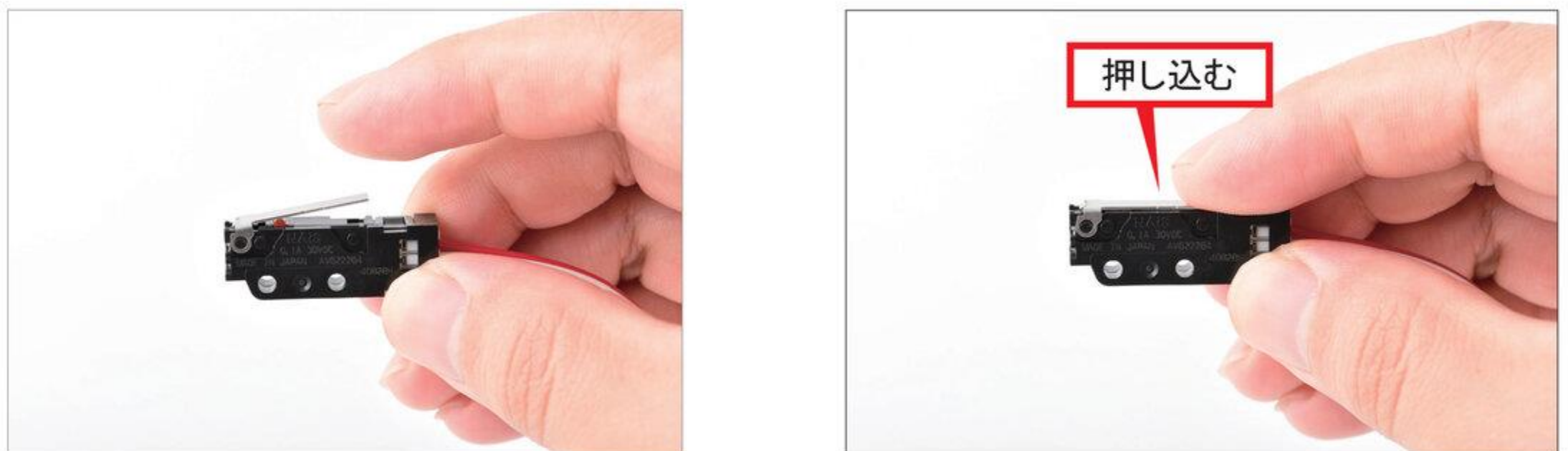


図 1-2 スイッチオフの状態 (左図) とオンの状態 (右図)

次に、[D1] につながったスイッチの動作の確認をします。以下のプログラムを実行してください。

### プログラムの書き込み

RoboticsProfessorCourse1 > PreCourse > Switch1

実行結果：[D1] につながったタッチセンサーのスイッチ部の金属バーを押すと、「カチッ」と音が鳴ってスイッチがオンになり、マイコンボードの 13 番 LED が点灯します。



## 1.2. LED カウントを操作してみよう

次にタッチセンサーの入力値をマトリクス LED に表示してみましよう。


### やってみよう！

3つのプログラムを見比べて、実行結果のちがいを確認してみよう。  
それぞれのプログラムの実行結果を書いてみてね。

### ∞プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > LedCount1

[D0] と [D1] のタッチセンサーのスイッチを押したときに、マトリクス LED はどのように変化しますか？


タッチセンサー [D0] を押したときに、 数字が1つずつ増えていく。


タッチセンサー [D1] を押したときに、 何も起こらない。

次に、以下のプログラムを実行して、同じように変化を見ましよう。

### ∞プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > LedCount2

タッチセンサー [D0] を押したときに、 数字が1つずつ増えていく。


タッチセンサー [D1] を押したときに、 数字が1つずつ減っていく。

最後に、以下のプログラムを実行して、同じように変化を見ましよう。

### ∞プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > LedCount3

タッチセンサー [D0] を押したときに、 何も起こらない。

タッチセンサー [D1] を押したときに、 何も起こらない。

プログラム「LedCount3」ではスイッチと関係なく、数字が増えていきます。  
明らかに動作がちがいましたね。この3つのプログラムのちがいをみてみましよう。

講

それぞれのプログラムの実行時のマトリクス LED の変化を記入させましよう。  
LedCount3 では、自動的に数字が1ずつ上がっていくので、解答は「スイッチでは何も変化しない」となります。



パソコンのプログラムの画面も一緒に見比べましょう。

### □ プログラム「LedCount1」より抜粋

```
if(SW1.onPress()){
    i++;
    myMatrix.putd2(0, 0, i);
}
```

// もしSW1が押されたら  
// 変数iの値に1を足す  
// マトリクスLEDに2けたの数字を表示

### □ プログラム「LedCount2」より抜粋

```
if(SW1.onPress()){
    i++;
    myMatrix.putd2(0, 0, i);
}
if(SW2.onPress()){
    i--;
    myMatrix.putd2(0, 0, i);
}
```

// もしSW1が押されたら  
// 変数iの値に1を足す  
// マトリクスLEDに2けたの数字を表示  
  
// もしSW2が押されたら  
// 変数iの値から1を引く  
// マトリクスLEDに2けたの数字を表示

### □ プログラム「LedCount3」より抜粋

```
for(i = 0; i < 10; i++){
    myMatrix.putd2(0, 0, i);
    delay(500);
}
```

// マトリクスLEDに2けたの数字を表示

つまり、プログラム「LedCount1」は、SW1（タッチセンサー **D0** のこと）が押されるたびに、「変数 i」が1ずつ増えていきます。プログラム「LedCount2」は、それに加えて、SW2（タッチセンサー **D1**）が押されるたびに、「変数 i」が1ずつ減っていきます。プログラム「LedCount3」は、スイッチに関係なく0.5秒ごとに1ずつ増えていきます。for 命令は前回の授業でも登場しましたね。

講

パソコンのプログラム画面とも照らし合わせて、ゆっくりと一つずつのプログラムを確認させてください。



### 1.3. 変数とは

さて、ここで出てきた「変数*i*」とは何のことでしょうか？ 変数とは、値が変化していく部分を文字で表したものです。前のページのプログラムは、はじめに「*i*」という名前の変数をつくっておいて、タッチセンサーや時間経過で「*i*」に入る値を変化させるという命令になっていたのです。マトリクスLEDは「*i*の値を表示する」とプログラムされていて、ループのたびに「いまの*i*の値はいくつだろう？」と確認します。「*i*」という箱をあけて、中に入っている数字を見るようなものです。

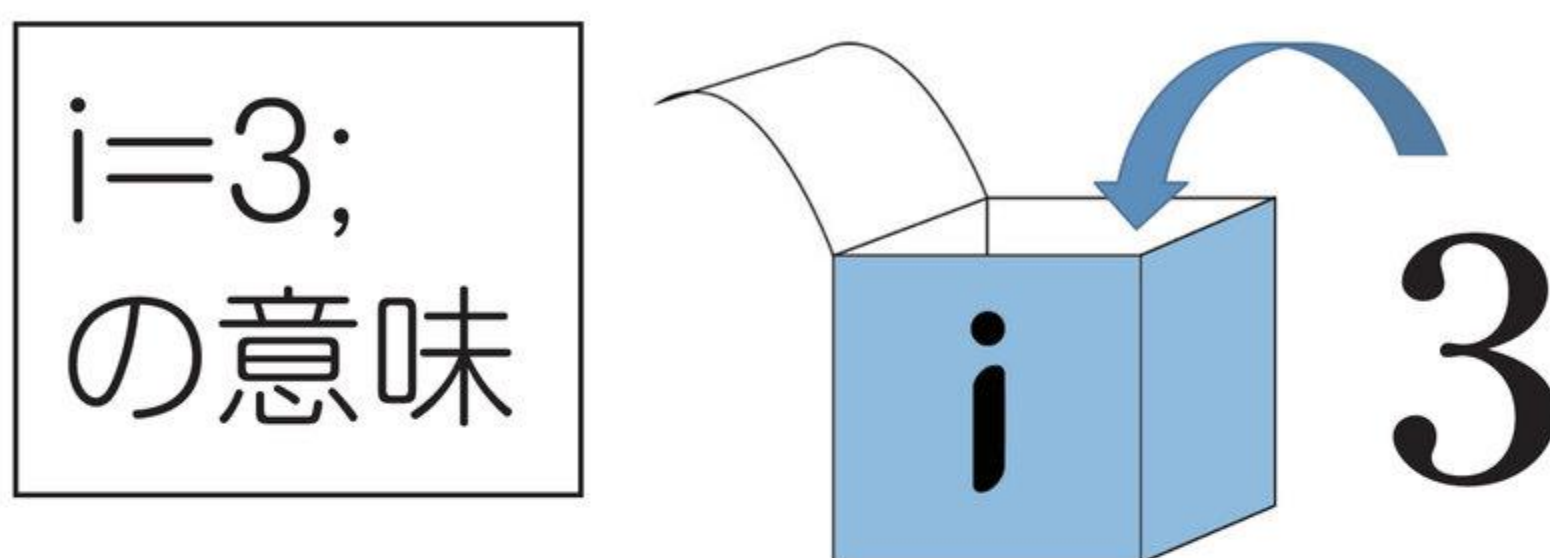


図 1-3 変数のイメージ図

#### □ プログラム例

```
int i; // 変数intをiという名前で使う
```

「int」は整数型という変数の種類です。このプログラムでは、マトリクスLEDの数字は1ずつ変化させるので、2.3や0.7といった小数は出てきませんよね。そのため、「中に整数しか入らない変数を使います」という宣言をしているのです（もし小数も使いたいときは、intではなく「float」などという種類の変数を使います）。

そのあとの「*i*」は、変数として使う文字の宣言です。「*i*」以外の文字を使ってももちろんかまいません。自分の名前の頭文字でもいいですし、1文字である必要はないので、犬が大好きな人は「Pochi」という変数にしてしまうこともできます。

#### 講

変数とはデータを一時的に保存する領域のことをいいます。プログラムの実行結果や計算結果、さまざまなセンサーの入力値を一時的に記憶して、プログラムの処理で利用することができます。



## 1.4. 変数と数式の理解を深めよう

では、プログラムを少し改造しながら理解を深めましょう。

### やってみよう！

プログラム「LedCount1」の黄色の部分<sup>へんこう</sup>を以下のように変更<sup>へんこう</sup>してみよう。  
プログラム変更<sup>へんこう</sup>後の実行結果を書いてみてね。

### □ プログラム「LedCount1」より抜粋<sup>ぼつすい</sup>


```
if(SW1.onPress()){ // もしSW1が押されたら
    i++; // 変数iの値に1を足す
    myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
}
```

```
if(SW1.onPress()){ // もしSW1が押されたら
    i = i + 2; // 変数iの値に2を足す
    myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
}
```

何が起きましたか？

 タッチセンサーを押すごとに、数字が 2、4、6、…と 2 ずつ増えるようになった。

同じように、 $i = i + 5;$  と変更<sup>へんこう</sup>して実行してみましよう。何が起きましたか？

 5、10、15 と 5 ずつ増える。

### やってみよう！

プログラム「LedCount2」を変更<sup>へんこう</sup>して、タッチセンサーを押したときに表示される数字が 10 ずつカウントアップするようにしてみよう。プログラムをどのように変更<sup>へんこう</sup>すればよいかをまずは書いてみてね。

  $i++$  を  $i = i + 10$  に変える。

同じように、数字のカウントダウンを 2 ずつにするようには、どのように変更<sup>へんこう</sup>すればよいでしょうか？

  $i--$  を  $i = i - 2$  に変える。

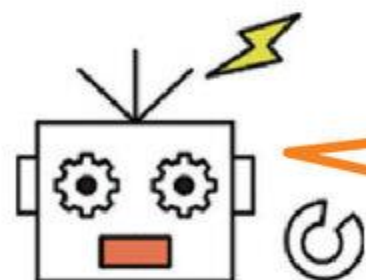


## チャレンジ課題

プログラム「LedCount3」のfor文を<sup>へんこう</sup>変更して、99、96、93、…、39、36、33と、99から3ずつ数字が減っていき、33で止まるようなプログラムにしてみよう。

講

for(i = 0; i < 10; i++) を for(i = 99; i >= 33; i = i - 3) に変えます。



変数の理解ができれば、ロボットがとってもカシコクなるよ！



## 2. スピーカーを使ってみよう (目安 35 分)

### 2.0. スピーカーの取り付け

スピーカーを図 2-0 のように [D2] に接続しましょう。

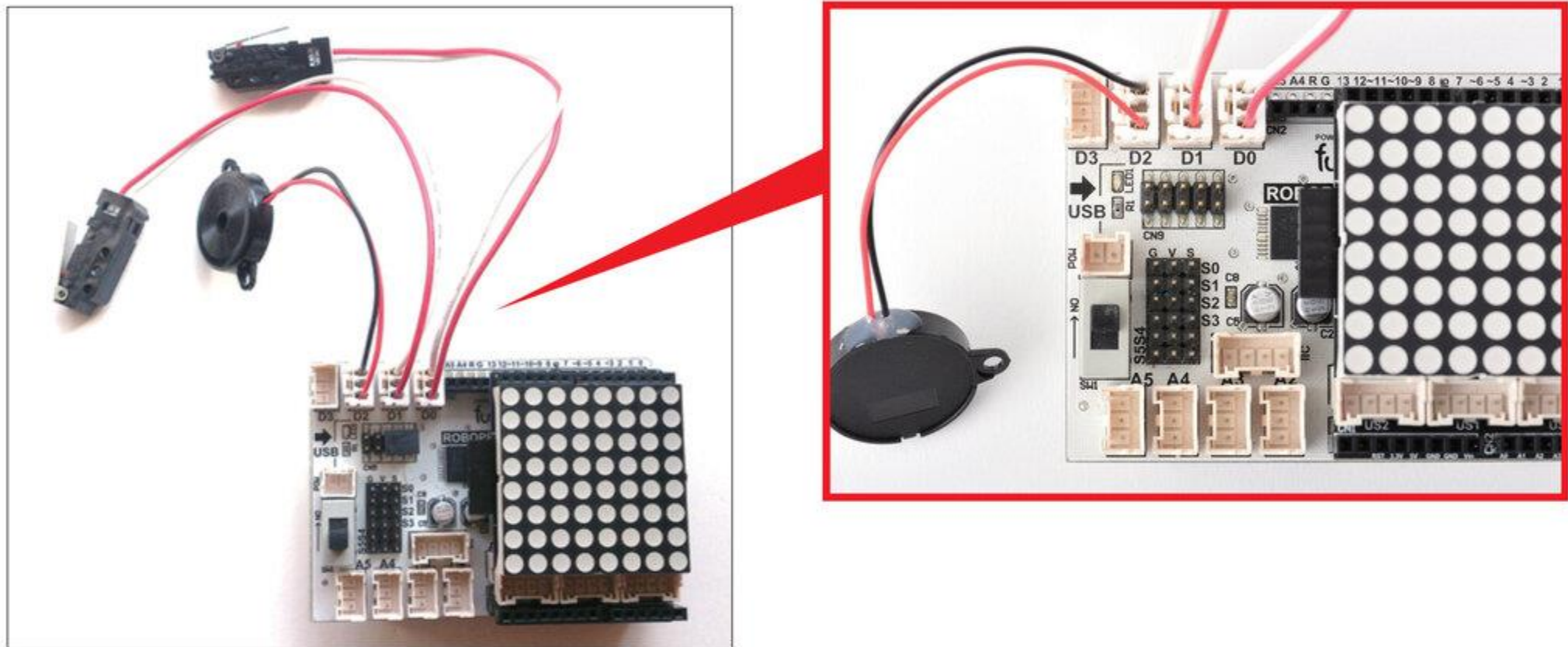


図 2-0 スピーカーの取り付け

### 2.1. if 文を使ってみよう

まずは以下のプログラムを実行してください。

#### ∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > LedCount4

プログラムの実行後、[D0] のタッチセンサーを押していくと何が起きるでしょうか？  
タッチセンサーを 10 回押してみましょう。どんな音が出ましたか？



では、プログラムの中を見てみましょう。

### □ プログラム「LedCount4」より<sup>ぼっすい</sup>抜粋

```
void loop(){

  SW1.listen();           // SW1の状態をマイコンに聞く
  SW2.listen();           // SW2の状態をマイコンに聞く

  if (SW1.onPress()){     // もしSW1が押されたら ①
    i++;                  // 変数iの値に1を足す
    myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
    tone1.play_rtttl(sound1); // sound1の曲を流す
    if (i == 10){        // ②
      tone1.play_rtttl(sound2); // sound2の曲を流す
    }
  }

  if (SW2.onPress()){     // もしSW2が押されたら ③
    i--;                  // 変数iの値から1を引く
    myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
  }

}
```

さて、いくつか新しい命令がありますが、ここでは `if` に注目します。

`if` 文は「もし、～だったら、～をする」という命令で、ロボットの人工知能の中枢<sup>ちゅうすう</sup>になる命令になります。

#### 命 令: `if`

実行結果：もし、「条件」をみたすならば「実行内容」を行う

使 い 方:

```
if([条件]){
  [実行内容]
}
```



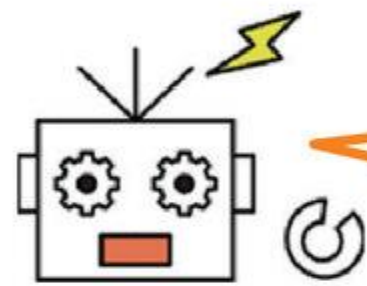
`SW1.onPress()` は、`D0` のタッチセンサーが押されているかチェックするための命令です。`SW2` なら、`D1` タッチセンサーをチェックします。

つまり、`void loop()` の中は、次のような処理になっていることがわかります。

**POINT**

- ①もし、`SW1` が押されたら、変数 `i` の値に 1 を足す、画面に表示する、「`sound1`」を出すをする
- ②もし、①を実行したときに、変数 `i` が 10 だったら、「`sound2`」を出すをする
- ③もし、`SW2` が押されたら、変数 `i` の値から 1 を引く、画面に表示するをする

なんとなく、プログラムの意味がわかってきそうですね！



超重要なポイントだよ！



## やってみよう！

`if (i == 10)` の条件を以下のように変更して、実行時にどのような動きになるか書いてみよう。

**1) `if(i >= 5)`**

`if(i >= 5)` は、変数  $i$  が 5 以上だったら、という意味になります。

 数字が 5 以上のときだけ、タッチセンサーを押すと `sound2` の音が鳴る。

**2) `if(i <= 5)`**

`if(i <= 5)` は、変数  $i$  が 5 以下だったら、という意味になります。

 数字が 5 以下のときだけ、タッチセンサーを押すと `sound2` の音が鳴る。

## ステップアップ

プログラム「LedCount4」を変更して、カウント 20、30、40、50 でも「`sound2`」を鳴らすようにしてみよう。

 ヒント

「もし変数  $i$  が 20 だったら」「もし変数  $i$  が 30 だったら」と、1つひとつ `if` 文が必要だよ！

講

解答例は巻末に記載します。



## チャレンジ課題

プログラム「LedCount4」を<sup>へんこう</sup>変更して、以下のように動作するようにしてみよう。

- ・ カウント 0 ~ 9 のときは、カウントアップしても音が出ない。
- ・ カウントが 10 以上のときは、カウントアップすると音 (sound1) が出る。
- ・ カウントが 20 になったら、sound2 が出る。

 ヒント

タッチセンサーを押したときの動作はいくつかあるよ。これらを、数字に関係なくいつも実行するものと、特定の数字のときだけ実行するものに分けて考えてみよう！

講 解答例は巻末に記載します。



## 豆知識

今回のプログラムでは、「i の値が 10 か」をチェックするとき `i == 10` という式を書いていました。等号 (イコール) が 2 つつながっています。

Arduino のプログラミングでは、`i == 10` など等号が 2 つ並んでいるときは「左側と右側が等しいかどうかチェックする」という意味に、`i = 10` など等号が 1 つだけのときは「左側の変数の値を右側の数字と同じにする」という意味になります。

つまり、`==` は変数の値を変えずにチェックだけをするのに対し、`=` は変数の値を変更することになります。





## コラム 文字を使った数式

数学の世界にも、プログラムの世界とちがった規則があります。今回は、特に文字が登場する式（文字式）のルールをまとめます。学校の数学にも必要なルールですので、しっかり学んでおきましょう。

### ●文字式の乗法（かけ算）と除法（わり算）のルール

乗法の場合は、かけ算の記号を省略する。

$$2 \times a = 2a \quad x \times y \times z = xyz$$

除法の場合は、わり算の記号を用いる代わりに、わる数を分母において分数の形に直す。

$$a \div 3 = \frac{a}{3} \quad x \times y \div z = \frac{xy}{z} \quad x \div y \div z = \frac{x}{yz}$$

数字と文字をかけるときは数字を先に、複数種類の文字をかけるときはアルファベット順に並べかえる。

$$n \times 5 = 5n \quad b \times c \times a = abc \quad i \times 3 \times h = 3hi$$

負の数（マイナスの数）についているかっこは省略する。

$$y \times (-2) = -2y \quad n \times (-8) \times m = -8mn$$

式についたかっこは省略せず、かっこそのものを1つの文字と考え、かけ算記号の省略や並べかえをする。

$$(a + b) \times 5 = 5(a + b) \quad (m + n) \times (x + z) = (m + n)(x + z)$$

分子や分母が1つのかっこのみでできているときは、かっこを省略する。

$$(a + b) \div 5 = \frac{a+b}{5} \quad (a + b) \times 2 \div (c + d) = \frac{2(a+b)}{c+d}$$

文字と1の積、または文字と-1の積になるときは、1は省略する。

$$n \times 1 = 1n = n \quad p \times (-1) \times c = -1cp = -cp$$

同じ文字やかっこを2回以上かけるときは、かける回数（指数）を右上に小さく書く。（『累乗』といいます）

$$a \times a = a^2 \quad m \times p \times m \times p \times m = m^3 p^2$$

$$(a + b) \times (a + b) \times (a + b) = (a + b)^3$$

### ●代入と式の値

文字式の中の文字をある値とおきかえるとき、「文字にその値を代入する」といいます。値を代入した式が計算できるようになったとき、計算によって求められた値をその式の値といいます。

文字式  $5a + 4$  において、 $a$  に  $8$  を代入したときの式の値は、

$$\begin{aligned} &5 \times a + 4 \\ &= 5 \times 8 + 4 \\ &= 44 \end{aligned}$$

$x = -5$  のとき、文字式  $10 - 3x$  の値は、

$$\begin{aligned} &10 - 3 \times x \\ &= 10 - 3 \times (-5) \quad \leftarrow \text{負の数を代入するときはかっこをつけます。} \\ &= 10 + 15 \quad \leftarrow \text{かけ算を先に計算するため、} -3 \times (-5) = +15 \text{ です。} \\ &= 25 \end{aligned}$$



### 3. タッチセンサーとスピーカーで遊ぼう (目安 20 分)

#### 3.0. タッチセンサーを使ってキャラクターを動かす

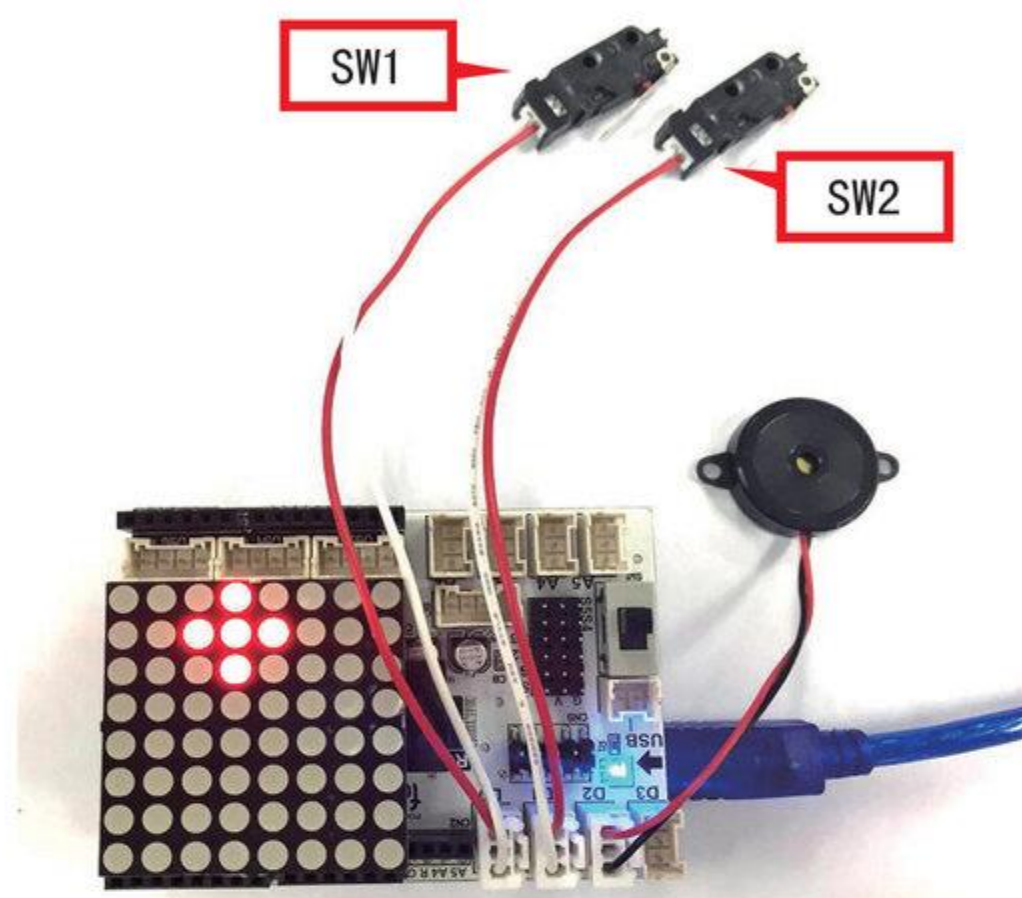


図 3-0 タッチセンサーの位置

ここまで学んだことを使って、キャラクターを思い通りに動かしてみましょ。以下のプログラムを実行してください。

#### ∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > MatrixSpriteMove1

実行結果：マトリクス LED に十字マークが表示され、SW1/SW2 を押すと左右に移動します。

次に以下のプログラムを実行してください。

#### ∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > MatrixSpriteMove2

実行結果：十字マークが左右に移動すると、同時にスピーカーから音が出ます。少しゲームらしくなってきました！

講

生徒が更なるチャレンジを希望する場合には、MatrixSpriteMove2 のキャラクターを自分の好きなものに変更させてもよいでしょう。



### 3.1. スピーカーで遊ぶ

ここまでいろいろと新しいことをしましたが、スピーカーの秘密に一度ふれておきましょう。わからなくても、あせらなくて大丈夫です。今回は体験を<sup>だいじょうぶ</sup>して、後からまたしっかりと勉強をしていきます。以下のプログラムを実行してください。

#### 🔗 プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA2 > RTTTLtest

実行結果：曲が流れます。

聞こえましたか？ 今回はいろいろな曲を聞いてみましょう。

プログラム「RTTTLtest」の黄色の部分<sup>へんこう</sup>を以下のように変更してください。

#### 📄 プログラム「RTTTLtest」より抜粋<sup>ぼつすい</sup>

```
//char *song = "M3A:d=4,o=4,b=135:16g,16c5,16p,……
//char *song = "M4A:d=4,o=4,b=170:16g,16c5,16p,……
char *song = "M5A:d=4,o=6,b=164:16c,16a5,16b5,……
//char *song = "M6A:d=4,o=6,b=173:8c,16c,16g5,8d,……
//char *song = "M7A:d=4,o=5,b=173:4g,4a,16b,16b,……
```

```
//char *song = "M3A:d=4,o=4,b=135:16g,16c5,16p,……
//char *song = "M4A:d=4,o=4,b=170:16g,16c5,16p,……
//char *song = "M5A:d=4,o=6,b=164:16c,16a5,16b5,……
char *song = "M6A:d=4,o=6,b=173:8c,16c,16g5,8d,……
//char *song = "M7A:d=4,o=5,b=173:4g,4a,16b,16b,……
```

<sup>へんこう</sup>変更前とは違う曲が流れましたね。

いま削除<sup>さくじょ</sup>したり新たに書きこんだりした「//」は「コメントアウト」という機能を使うための記号でした。今回のプログラムでも「//」がついている曲はプログラムとして認識されないため、「//」がついていない曲だけが「\*song」という変数に入り、流れるようになっています。

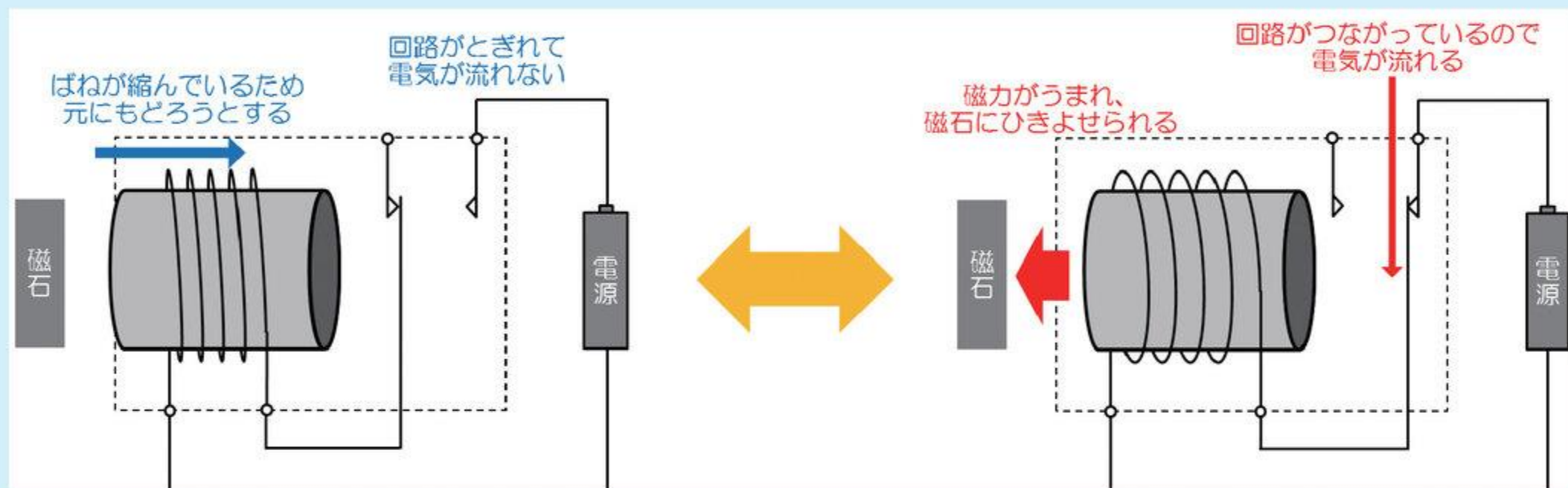




## コラム ブザーのしくみとは

スピーカーとは、電気を利用して音を鳴らす機械です。そのしくみにはたくさん種類がありますが、「電磁石」を使ったものが一般的です。

導線（電線）には、電気が流れているときだけ磁石の力（磁力）をうむという性質があります。みなさんの家にある電気コードも、電源がオンのときは磁力をもっているのです。とても弱い力なので磁石にくっつけることはできませんが、導線を何度もグルグル巻くなどして一か所に集中させると、磁石にくっつけることができるくらい磁力を強められます。導線をグルグル巻いたものを「コイル」といい、コイルを磁石として使う機械を「電磁石」といいます。



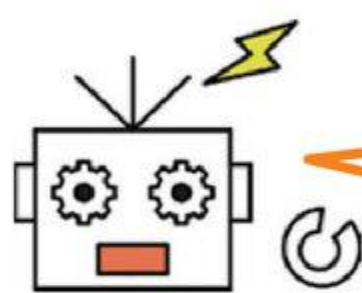
このように、電磁石に電気が流れているときと流れていないときを交互にくりかえすことで、振動がうまれます。1秒間に数百回～数千回というスピードで振動させ、それが空気に伝わることで音が鳴るというしくみです。



## 4. まとめ（目安5分）

今回はマトリクスLEDとタッチセンサー、そしてスピーカーを使ってif文を学びました！  
if文は「こういうときはこう動く」という、ロボットの考えを決めるととても大事な命令です！  
今後何度も登場しますので、ぜひマスターしましょう！

また、スピーカーの音の鳴らし方を少しだけ学びましたが、これは次回もっとくわしく学びます！  
どんな曲を流したいか、考えておいてください！



スピーカーを使った音の鳴らし方に関してはまた次回くわしく行います！では、また、次回お楽しみに！

### 《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 <span style="float: right;">1</span>                                       | マイコンボード <span style="float: right;">1</span>   | ロボプロシールド <span style="float: right;">1</span>   | リボンケーブル <span style="float: right;">1</span>  |
|  |  |  |  |
| コントローラー <span style="float: right;">1</span>  | 無線受信モジュール <span style="float: right;">1</span>                                       | タッチセンサー <span style="float: right;">2</span>  | マトリクスLEDシールド <span style="float: right;">1</span>                                     |
|  |  |  |  |
| マトリクスLED <span style="float: right;">1</span>                                       | スピーカー <span style="float: right;">1</span>   |   |   |
|  |  |   |   |

図 4-0 次回必要なもの



## 講

- 以下の授業の目標を再確認します。
  - ・タッチセンサー（入力装置）とスピーカー（出力装置）の使い方を知る
  - ・変数を使いこなす
  - ・if 文の使い方をマスターする
  - ・曲のデータを見る
- 次回テーマは「マイコンで演奏してみよう」であることを告知します。

## P.13 ステップアップ 解答例 (void loop 以下を抜粋)

```
void loop(){

    SW1.listen();           // SW1の状態をマイコンに聞く
    SW2.listen();           // SW2の状態をマイコンに聞く

    if (SW1.onPress()){     // もしSW1が押されたら
        i++;                // 変数iの値に1を足す
        myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
        tone1.play_rtttl(sound1); // sound1の曲を流す
        if (i == 10){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
        if (i == 20){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
        if (i == 30){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
        if (i == 40){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
        if (i == 50){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
    }

    if (SW2.onPress()){     // もしSW2が押されたら
        i--;                // 変数iの値から1を引く
        myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
    }

}
```



## P.14 チャレンジ課題 解答例 (void loop 以下を抜粋)

```
void loop(){

    SW1.listen();           // SW1の状態をマイコンに聞く
    SW2.listen();           // SW2の状態をマイコンに聞く

    if (SW1.onPress()){    // もしSW1が押されたら
        i++;                // 変数iの値に1を足す
        myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
        if (i >= 10){
            tone1.play_rtttl(sound1); // sound1の曲を流す
        }
        if (i == 20){
            tone1.play_rtttl(sound2); // sound2の曲を流す
        }
    }
    if (SW2.onPress()){    // もしSW2が押されたら
        i--;                // 変数iの値から1を引く
        myMatrix.putd2(0, 0, i); // マトリクスLEDに2けたの数字を表示
    }
}
```