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Ⅰ－１①

（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。

第１回授業日　2024年　　月　　日

第２回授業日　2024年　　月　　日

名　前

Human ヒューマンアカデミー ジュニア STE"AMスクール

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7月授業分

※私的使用以外の複製・配布・転売・譲渡等を行うことを禁止いたします。

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－10①

第1回

マトリクスLEDで遊ぶ

講師用

目　　次

○　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。

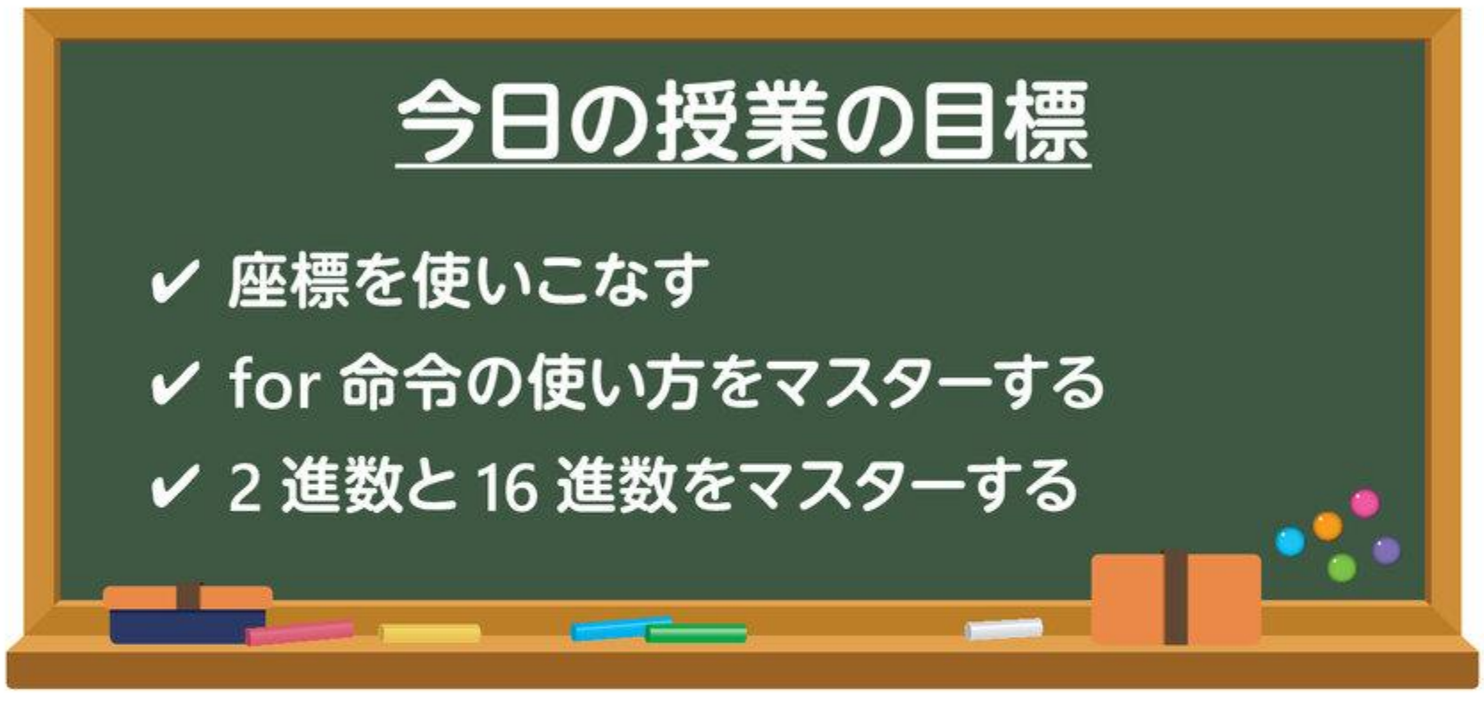
○　今回の目標をパネルで用意するか、黒板に予め書いておきます。

　（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．マトリクス LED で遊ぶ（目安 5 分）

0.0. 「マトリクス LED で遊ぶ」でやること

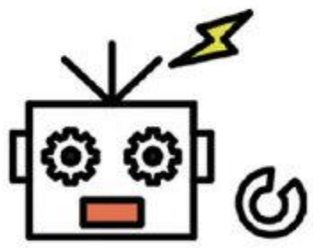


今回は、「マトリクス LED」という電子部品を使って、プログラムの基本をマスターしていきます。

LED は知っていますね？ 最近は省エネを行う目的で、家庭にもたくさん入ってきています。小さくても、とても明るく光る素子のことです。その LED が複数集まって形を表示できるようになったものが、マトリクス LED と呼ばれる電子部品になります。この授業で使うものは 8 × 8 と少し小さいですが（LED が全部で 64 個集まっています）、つなげていけば、電車やバスで使われているような電光掲示板になるものです。原理的には TV やパソコン、携帯電話の液晶画面と同じです。マトリクス LED を活用すれば、文字や図形、そしてロボットの顔も表示できますね。

また、プログラムをしていると欠かせない巧みの技、2 進数と 16 進数についても学びます。日常で使っているものは 10 進数です。2 進数、16 進数とはいったいどんなものでしょうか？ さらに、さまざまなプログラムでも必要になる、くり返し命令「for」に関してマスターしていきます。

今回は、マトリクス LED とプログラムを使って、好きな絵や文字を表示したり、LED のドットを動かしたりしていきましょう！



マトリクス LED で LED を自由にあやつろう！

0.1. 必要なもの

以下のパーツを準備しておきましょう。
なお、マトリクスLEDのピンは折り曲げないように注意しましょう。

USBケーブル 1	マイコンボード 1	ロボプロシールド 1	マトリクスLEDシールド 1
			

マトリクスLED 1


図 0-0　必要なもの

講　今回の授業では、マイコンボードとパソコンをUSBケーブルで接続したままでプログラムの実行を行います（パソコンからの電力で動作させます）。

1．マトリクス LED シールドの組み立てと動作確認（目安 15 分）

1.0.　マトリクス LED シールドの組み立て

マトリクス LED シールドやマイコンボードなどを組み立てます。**図 1-0** が完成図です。下から、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 の順番になります。手順に沿って組み立てていきましょう。

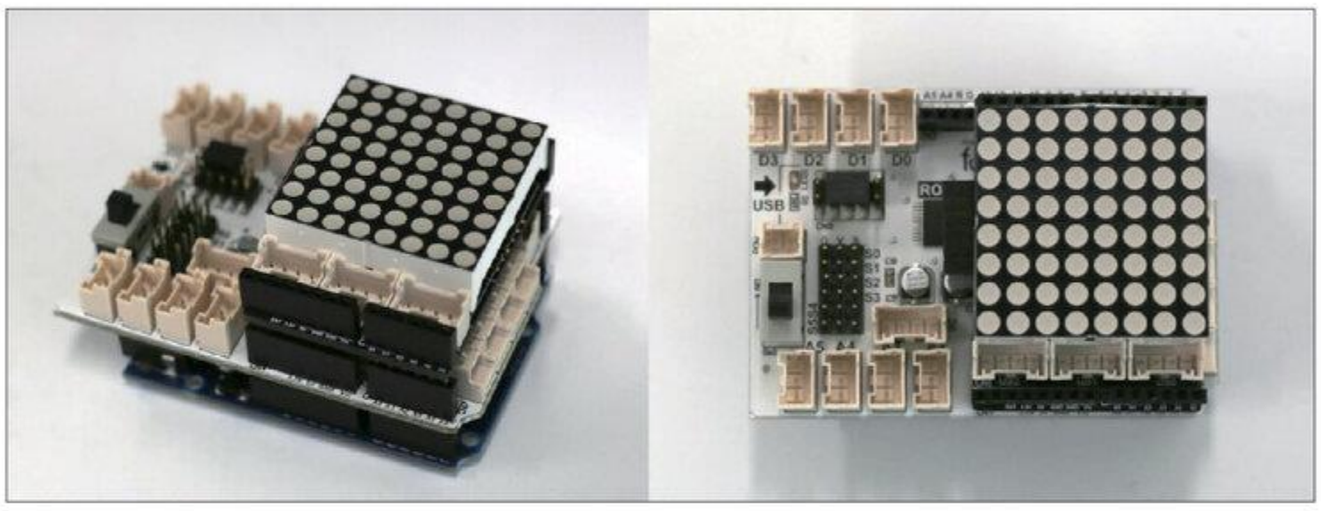


図 1 -0　マトリクス LED シールドの組み立て完成図

＜組み立て手順①＞

マトリクス LED をマトリクス LED シールドに取り付けます。マトリクス LED のさし込む向きに注意して、ピンを折らないように慎重に進めましょう。

まず、マトリクス LED シールドのソケットと、マトリクス LED のピンがある側の側面の凹凸を目印に、さし込む向きを合わせましょう。**図 1-1** のような向きになります。

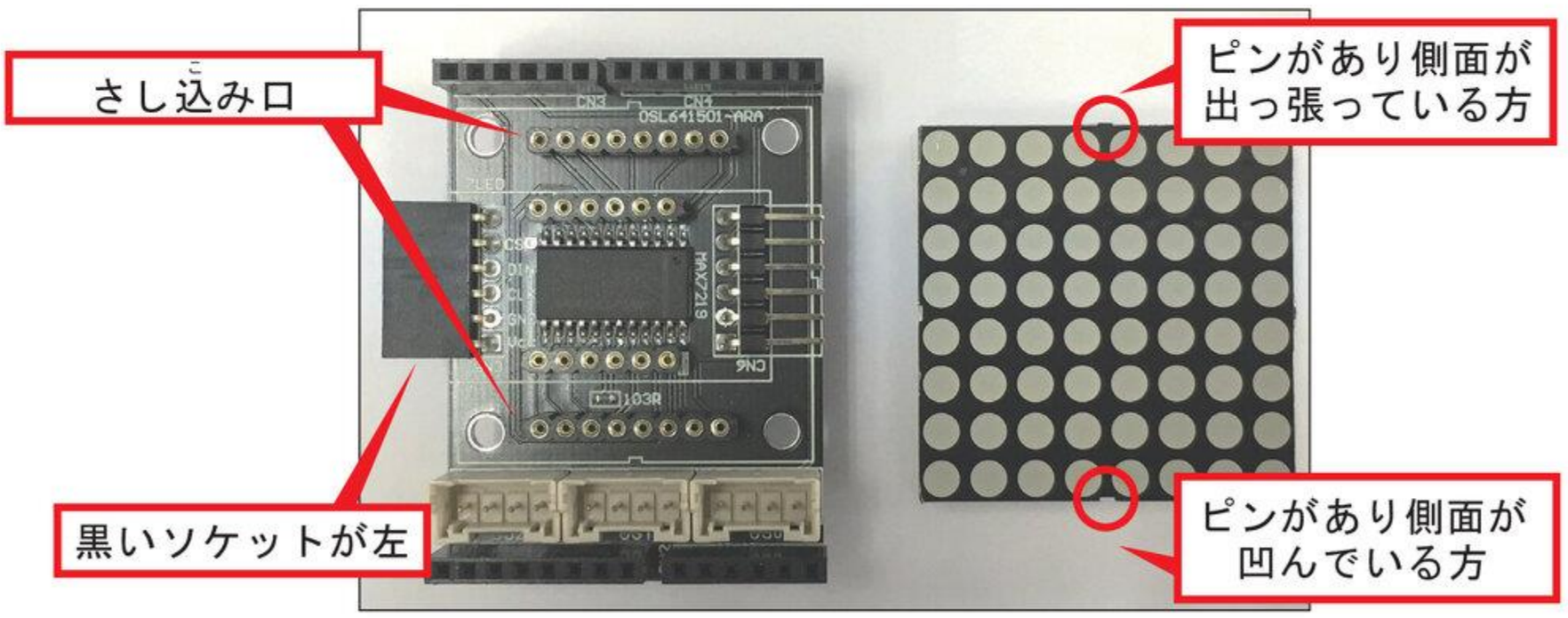


図 1-1　マトリクス LED シールド（左）とマトリクス LED（右）の向きの確認

＜組み立て手順②＞

向きを合わせたら、そのままさし込んでいきます。少しでもピンが曲がっていると入りません。さす前にピンを確かめて、曲がっていたら直しながらはめ込みます。なお、最初はななめにピンをさし込んでいくのがコツです。

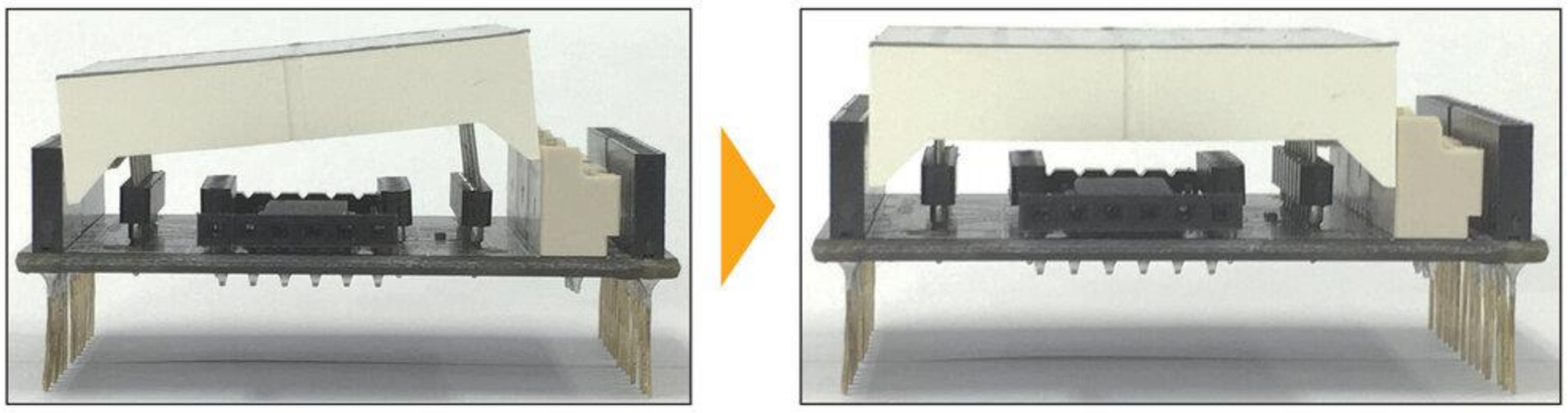
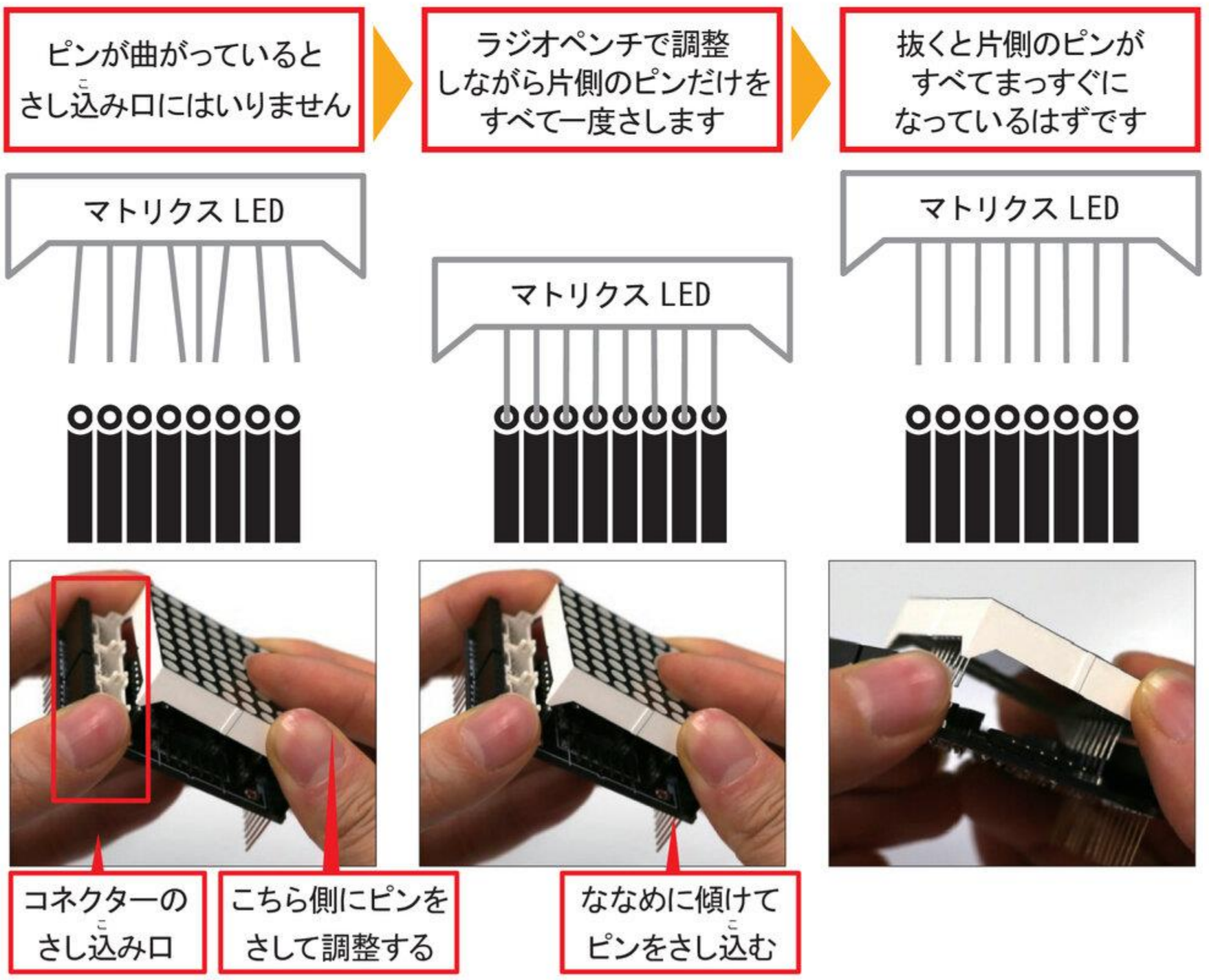


図 1-2　マトリクス LED の組み立て

POINT

マトリクス LED の裏側のピンが曲がっていると、うまくマトリクス LED シールドにさし込めません。そんなときは、次の方法を試してみてください。
マトリクス LED の片側のピンを、マトリクス LED のコネクターのさし込み口がない方にさします。うまくささった後に抜くと、ピンがまっすぐになります。
もう片方も 180 度回転させて同じようにすると、両側のピンがまっすぐになるので、さし込み口にさしやすくなります。



＜組み立て手順③＞

組み合わせたマトリクス LED シールドをマイコンボードとロボプロシールドに接続します。

図 1-3 のように、下からマイコンボード、ロボプロシールド、マトリクス LED シールドとなるようにしましょう。それぞれのはしがそろうように組み合わせましょう。さし込む向きをまちがえないように注意しましょう。

なお、マトリクス LED シールドは奥までさしても 2mm くらいはういています。

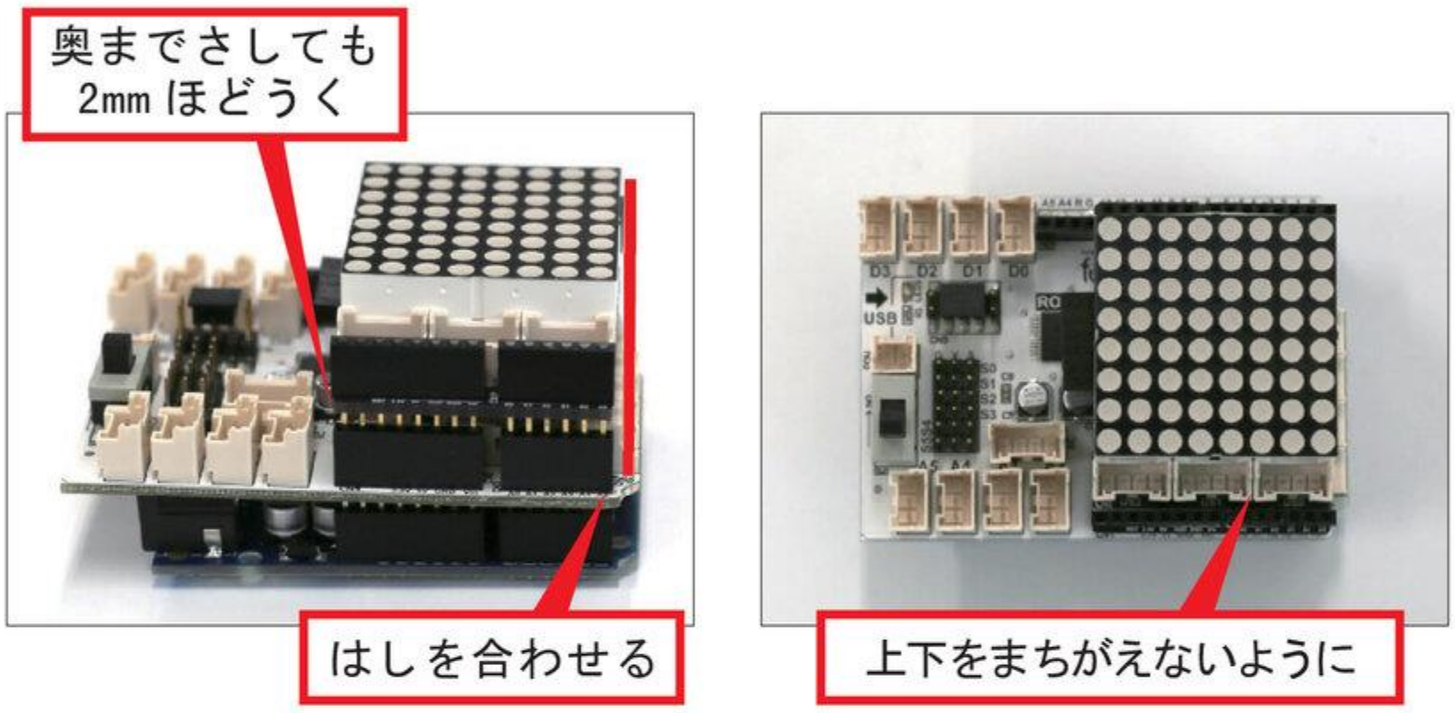


図 1-3　マトリクス LED シールドとロボプロシールドの接続

講 ピンヘッドは、力を入れて無理にさし込まないよう慎重に行わせてください。ピンソケットからピンヘッドへ電気信号が送られますので、ピンヘッドが一本でも抜けていますとマトリクス LED に命令通りの表示がされません。横方向から点検しながら行わせてください。

1.1. マトリクスLEDの動作確認

マトリクスLEDが使える準備が整ったら、プログラムを書き込んで動かします。
USBケーブルを準備して、パソコンとマイコンボードのUSBコネクターへ接続してください。
接続したら、ロボプロシールドのLEDが**図1-4**のように点灯しているか確認しましょう。もし点灯しない場合は、マイコンボードとロボプロシールドが正しく接続できているか確認しましょう。
なお、今回使わないモーターやセンサーが接続されていたらロボプロシールドから外しておいてください。

図1-4　ロボプロシールドのLED1の点灯

続いて、以下のプログラムを書き込んでください。
プログラムを書き込む方法を忘れてしまった人は、スタートアップのテキストで確認しましょう。

プログラムの書き込み
RoboticsProfessorCourse1 > MagicItemA1 > MatrixTest

実行結果：マトリクスLEDに以下のような点滅が表示されます。

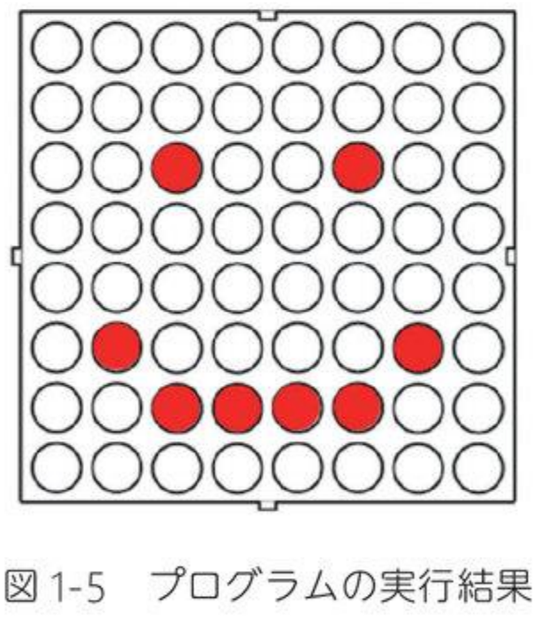


図1-5　プログラムの実行結果

2．座標を使って LED を動かそう（目安 45 分）

2.0.　LED を点滅させてみよう

1）プログラムの実行

動作確認を終えたら、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 > MagicItemA1 > MatrixBlink

実行結果：マトリクス LED のドットが 1 つだけ点滅します。

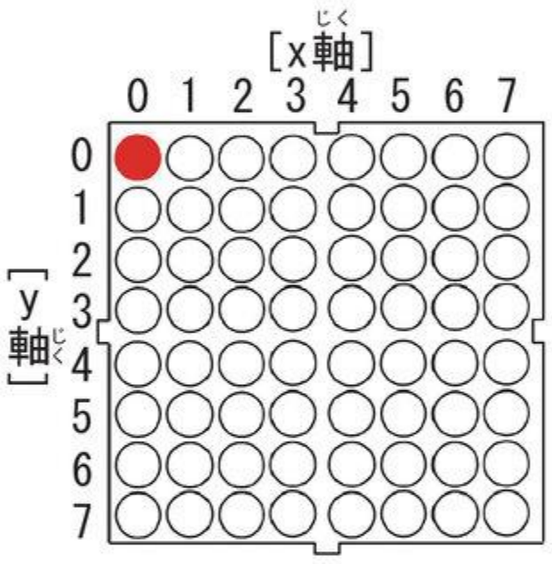


図 2-0　プログラムの実行結果

やってみよう！

プログラム「MatrixBlink」を以下のように変更してみよう。

プログラム「MatrixBlink」より抜粋

```
void loop(){

    myMatrix.write(0, 0, HIGH);   // LEDを点灯させる
    delay(1000);
    myMatrix.write(0, 0, LOW);    // LEDを消灯させる
    delay(1000);
}
```

1, 1, HIGH
500
1, 1, LOW
500

実行結果：点滅する場所が変わって、点滅がはやくなります。
さて、どうしてこのようになるのでしょうか？ いっしょに考えていきましょう。

2）マトリクスLEDへの命令

それでは、プログラム「MatrixBlink」の中身を見ていきましょう。
このプログラムでは、マトリクスLEDをつけたり消したりするための命令が使われています（黄色の部分）。

プログラム「MatrixBlink」より抜粋

```
void loop(){

    myMatrix.write(0, 0, HIGH);    // LEDを点灯させる
    delay(1000);
    myMatrix.write(0, 0, LOW);     // LEDを消灯させる
    delay(1000);
}
```

「myMatrix.write」の使い方は以下のようになります。

命　　令　**「myMatrix.write」**
実行結果：好きな場所のLEDをつける、消す
使 い 方：myMatrix.write(0, 2, HIGH);
　　// x座標が0、y座標が2のLEDを点灯する

myMatrix.write という命令を使うと、マトリクスLEDのねらった位置のLEDだけをつけたり消したりすることができますね。

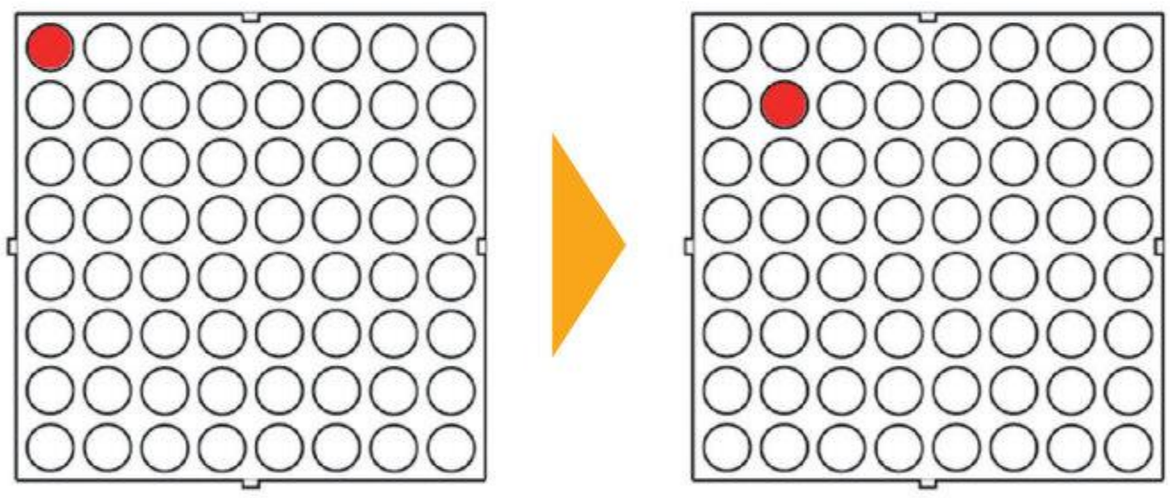


図2-1　点滅場所の変更

LEDの位置は「x座標」と「y座標」とよばれる2つの値で指示します。これは後で詳しく説明します。

POINT

HIGH と書いてある部分を LOW に書きかえると「LEDを消す」という命令になります。

3）delay

次に、delay(); について説明します。
「delay」を直訳すると、「遅れる」や「延ばす」という意味になります。プログラムでも意味合いは同じで「遅らせる」、「待つ」という命令になります。（）内の数字は1/1000秒の単位で入力します。
delay(1000); で次のプログラムの実行まで1秒間の間隔を空けることになります。

プログラム「MatrixBlink」より抜粋

```
void loop(){

    myMatrix.write(0, 0, HIGH); // LEDを点灯させる
    delay(1000);
    myMatrix.write(0, 0, LOW);  // LEDを消灯させる
    delay(1000);
}
```

命　　令「delay();」
実行結果：指定の時間だけ、上の命令を続ける
使 い 方：delay(1000);
　　// 上の命令を、1000ミリ秒（1秒）だけ続けたあと次の行へ進む

さきほどのプログラムの変更では、「delay(1000);」から「delay(500);」にしました。
LEDを点灯させたり、消灯させたりしたまま止まっている時間が1000ミリ秒（1秒）からその半分の500ミリ秒（0.5秒）になったというわけです。
その結果、点滅のはやさが2倍になりました。

講　マトリクスLEDの座標の位置は、ロボプロシールドの上下方向と逆になります。詳しくは次の2.1. 座標で解説します。生徒全員のLEDの点滅が確認できたら次に進みます。

2.1. 座標

プログラムに出てくるx,yは、一般的には座標と呼ばれるものです。座標とは、**図 2-2** に示したように、場所を示す「住所」のようなものです。x軸は横方向を、y軸は縦方向を示します。

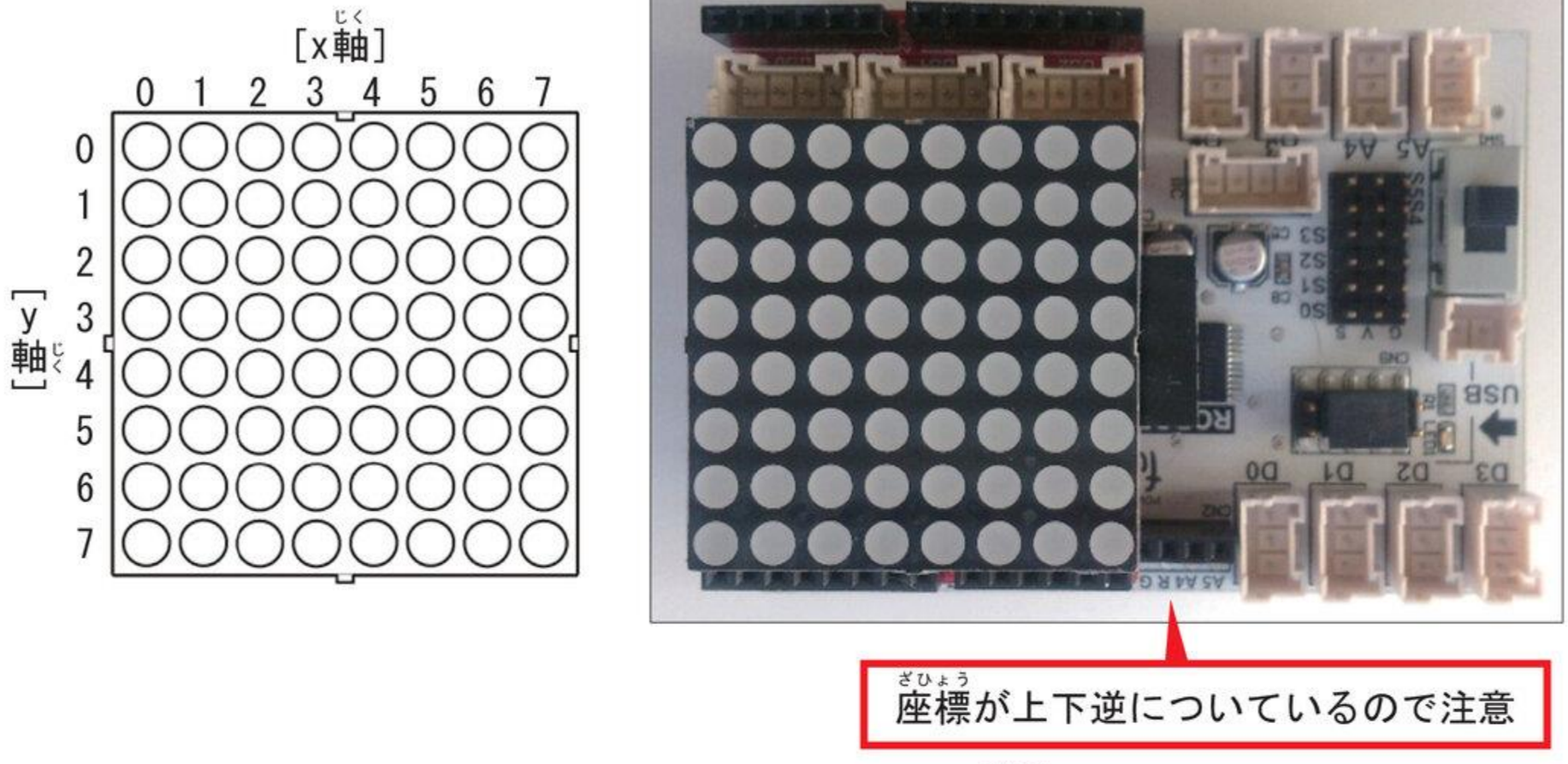


図 2-2　マトリクス LED の座標

myMatrix.write(0, 0, HIGH); とプログラム中に書けば、**図 2-2** の左上から右へ0番目、下へ0番目のLEDを指定したことになります。
myMatrix.write(1, 1, HIGH); とすれば、左上から右へ1番目、下へ1番目になります。

やってみよう！

プログラム「MatrixBlink」を変更して、座標と点滅のはやさを好きなように変えてみよう。
同時に2つ以上のLEDに命令することもできるかな？

プログラム「MatrixBlink」より抜粋

```
void loop(){

    myMatrix.write(0, 0, HIGH);  // LEDを点灯させる
    delay(1000);
    myMatrix.write(0, 0, LOW);   // LEDを消灯させる
    delay(1000);
}
```

コラム　座標とは

●座標

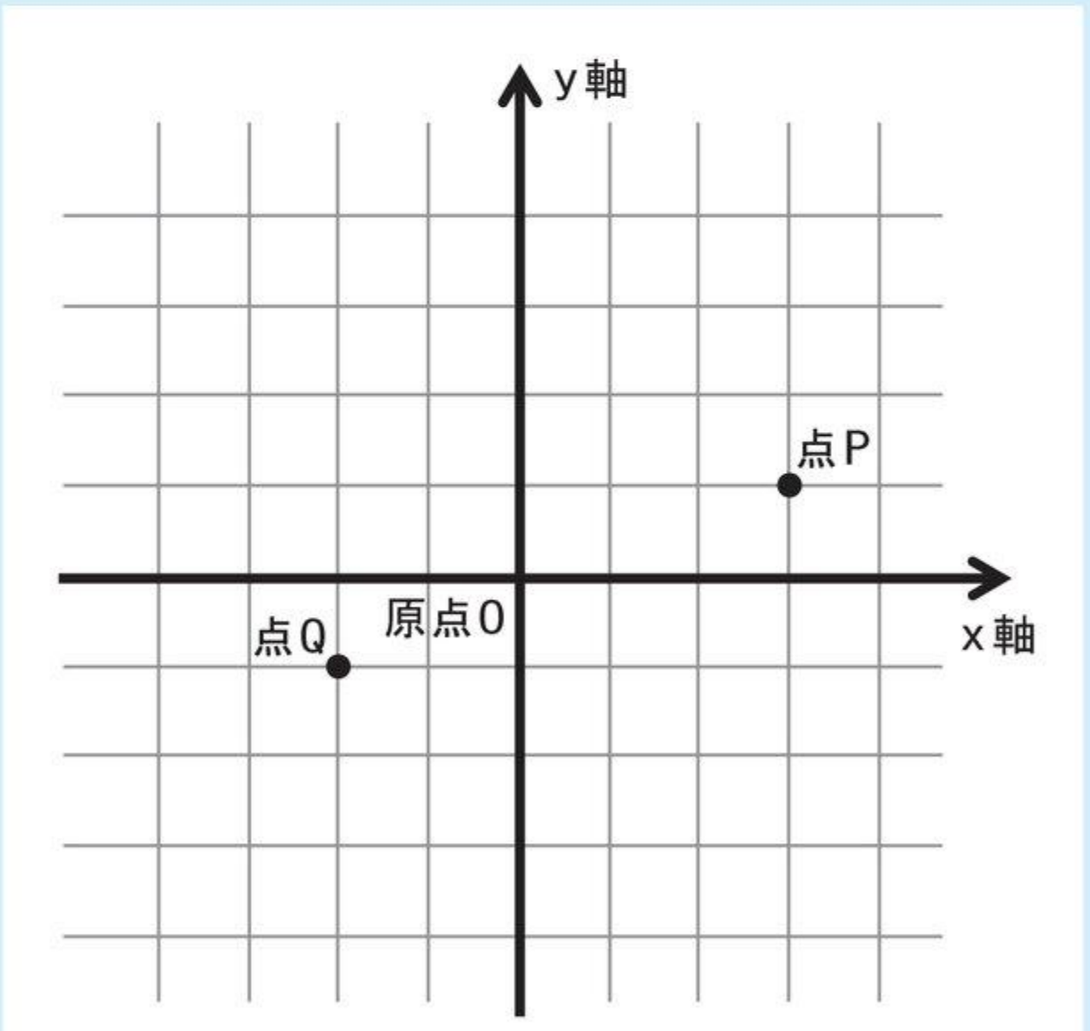
数学の世界では、平面上の点の位置を表すために下図のような「座標平面」というものを用います。

中央の縦と横の太線が交わっている場所を「原点」といい、ここをスタート地点とします。横向きの太線は「x軸」、縦向きの太線は「y軸」といい、原点からそれぞれの軸の方向にどれだけ移動したか数えるのです。移動した距離のことを「座標」といいます。マトリクスLEDの場合、いちばん左上に原点があり、x軸は右に、y軸は下に移動すると座標が大きくなっていきますが、数学の座標平面ではy軸は上に移動するほど数が大きくなるのが基本です。混乱しないように注意しましょう。

●点の表し方

たとえば、座標平面上に、下図のようにPという名前の点があるとします。この点はスタートである原点O（「原点」は英語で「Origin」というので、頭文字である「O」が名前として用いられる事が多いです）からx軸と同じ方向である右方向に3マス、y軸と同じ方向である上方向に1マス移動した先にあります。1マス移動したときの距離を1とすれば「点Pのx座標は3、y座標は1」という言い方になり、さらにまとめて「Pの座標は(3, 1)」と表現することができます。

ちなみにこの座標平面だと、原点よりも左にある点はx座標がマイナスの値に、原点よりも下にある点はy座標がマイナスの値になります。たとえば、下の図の点Qはx座標が－2で、y座標が－1なので、座標は(－2, －1)となります。マトリクスLEDにはマイナスの座標をもつ点は存在しませんが、数学にはその考え方がよく登場するので覚えておくとよいでしょう。



2.2. LEDを自由に移動させてみよう

点滅させるだけではつまらないので、ここではプログラムを使って自由にLEDを動かしてみましょう。LEDをプログラムで動かすということは、今後ロボットを動かすプログラムと同じテクニックをマスターすることになります。では、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 > MagicItemA1 > LedMove1

実行結果：一番上の段で、LEDが左から右へ動きます。
さきほどのプログラム「MatrixBlink」の命令を連続して、座標をずらしながら行っているだけで、LEDが動いて見えます。

やってみよう！

プログラム「LedMove1」を変更して、左から右へ動いたLEDをさらに右から左に動かして、LEDを往復させるプログラムにしてみよう。プログラムができ上がったら、実行結果を確認してみよう。

ヒント

LEDが左から右へ動くプログラムをコピー＆ペーストして座標の部分だけ書き直してみよう。
【コピー＆ペーストのやり方】
コピーしたい部分を選択して、キーボードの Ctrl キーを押しながら C キーを押す。
ペーストしたい場所を選択して、キーボードの Ctrl キーを押しながら V キーを押す。

講　解答例は巻末に記載します。所要時間は10分程度を目安としてでき上がった生徒から順に実行結果を確認してください。

2.3. for 命令を使ってみよう

前のページでは、一行一行時間をかけてプログラムを変更しましたが、LED を動かすたびにプログラムを書き足すのは大変な作業になりますね。そこで、プログラムをくり返すときに便利な方法を紹介します。それで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 > MagicItemA1 > LedMove2

実行結果：プログラムを見てみると、ずいぶん短いですが、LED を動かす数だけ増やしたときと、同じ実行結果となります。
これは「for」という命令を使ったことで、プログラムを短くすることができたのです。

プログラム「LedMove2」より抜粋

```
int i;

void loop(){
    for(i = 0; i < 8; i++){
        myMatrix.write(i, 0, HIGH);        // LEDを点灯させる
        delay(100);
        myMatrix.write(i, 0, LOW);         // LEDを消灯させる
    }
}
```

for については以下のようなポイントを覚えておきましょう。

命　　令　**「for()」**
実 行 結 果：{}　で指定された部分を決めた回数だけくり返し実行する
使　い　方：for(はじめの決まり ; 終わりの決まり ; くり返し実行すること){
　　くり返す内容
　　}

くわしくは次のページで解説します。

プログラム「LedMove2」のfor文を分解して読み取っていきましょう。

POINT

はじめの決まり　→　i=0

次回に説明しますが、「i」というのは数字を入れておく箱と思っていてください。
「はじめの決まり」 i=0 は、「最初に箱の中に0を入れておいてね」という意味です。

終わりの決まり　→　i<8

「終わりの決まり」 i < 8 は、「<」は左側のほうが右側より小さいという意味なので、「箱に入っている数字が8未満でなくなったらくり返しはやめてね」という意味になります。
逆に言えば「箱に入っている数字が8未満のときはくり返してね」という意味です。

くり返し実行すること　→　i++

「くり返し実行すること」は、{} の間の命令を実行するたびに処理される命令です。
この場合の i++ とは「iに1を足してください」という意味です。少し不思議な書き方をしますがなれておきましょう。
このプログラムは「MatrixBlink」と同様に、1つのLEDを点滅させるものです。ただし、
x座標が数字ではなく「i」になっていることがポイントです。
ちなみに、「<」という記号は、左側が右側未満であることを表します。「i<8」であればiは8未満なので、iが8と等しくなることはありません。
もし「iは8以下」、つまりiが8と等しくなるようにしたいときは「≦」という不等号を用い、「i≦8」と表します。プログラム内に書くときは「i<=8」と表します。iが8以上、つまり「i≧8」であれば「i>=8」と書きます。

くり返す内容　→　{} の間にある命令

{ } の間では以下のような命令をくり返していることになります。

やってみよう！

for 命令を試してみよう。以下のように（　）の中を変えると、何が起きるかな？
実行結果を記入しよう。

1.　for(i = 0; i < 8; i++)　→　for(i = 0; i < 4; i++)

解答例：0 から 7 まで移動していた点滅が 0 から 3 までで繰り返される。

2.　for(i = 0; i < 8; i++)　→　for(i = 4; i < 8; i++)

解答例：0 から 7 まで移動していた点滅が 4 から 7 までで繰り返される。

3.　for(i = 0; i < 8; i++)　→　for(i = 0; i < 8; i += 2)

解答例：1個ずつで移動していた点滅が 1 つおきに点滅する。

やってみよう！

プログラム「LedMove2」の for 文を書きかえて、光が点滅（てんめつ）していく方向を逆向きに、つまり右から左に移動していくように変更（へんこう）してみよう。

ヒント

for 文に必要なのは「はじめの番号」「終わりの番号」「番号がどう変化していくか」の 3 つだね！

講

解答は以下の通りです。
for(i = 7;i >= 0;i--)

ステップアップ

for 命令を使って以下のような動きをつくってみよう。
さきほど覚えた for 命令を使ったプログラム「LedMove2」を参考にして完成させよう。

①ナイトライダー

LED が行ったり来たりする。

ヒント

「左から右へ点滅」「右から左へ点滅」の2つは作成したね！　2つのプログラムを1つにまとめられれば完成だ！

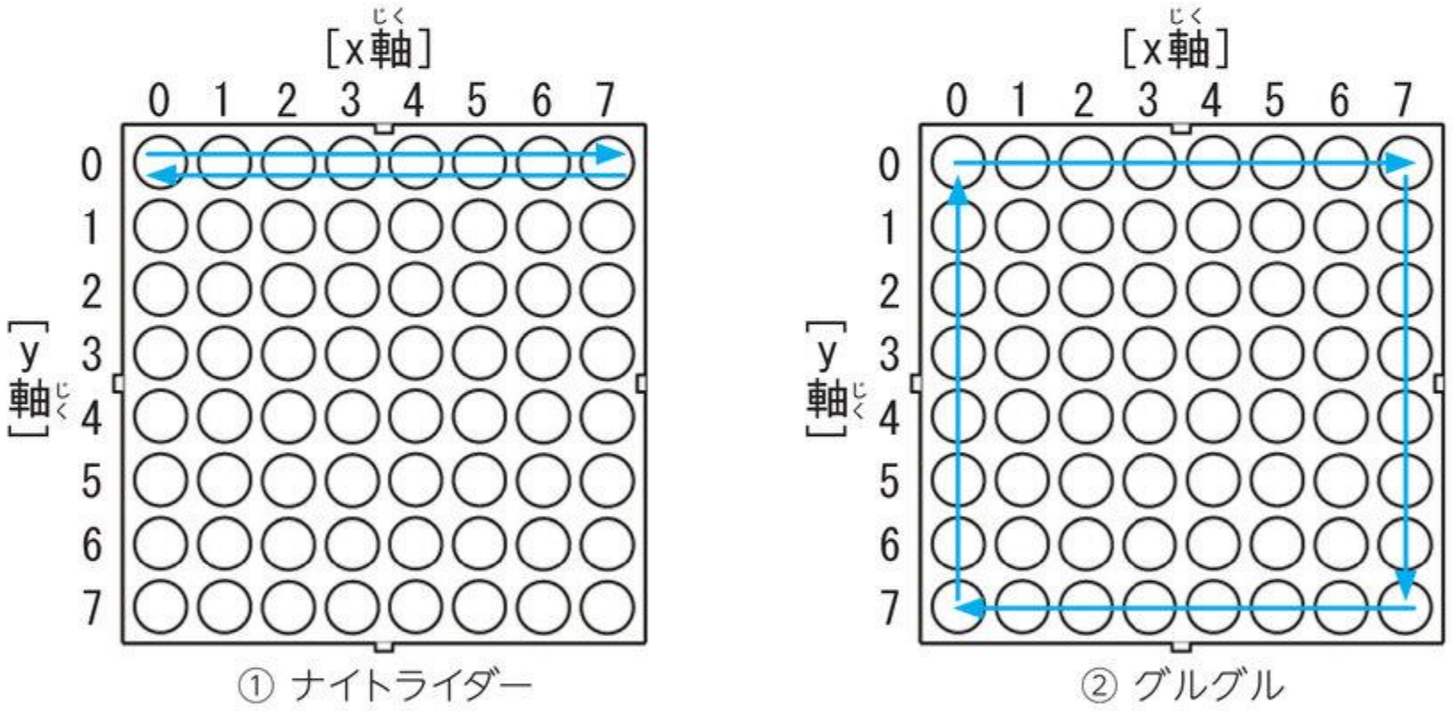
②グルグル

LED が左上から右上、右下、左下、左上と一周する。

ヒント

一周するまでに4種類の動きが必要だね。それぞれの動きで、x 座標や y 座標が「1ずつ増える」「1ずつ減る」「変化しない」のどれにあたるのか、整理してからプログラミングに取り組もう！

動きのイメージは、以下の図を参考にしてみよう。



① ナイトライダー　② グルグル

プログラムが完成したら、書き込んで実行結果を確認しましょう。

講

所要時間は15分程度以内を目安にしてください。少し難易度が高い問題がふくまれますが、解答例プログラムと LED の動きを確認させて理解を深めてください。
なお、「ステップアップ」の問題は、「やってみよう！」よりややハイレベルなものの、1年目の段階でできるようになっておきたい必達問題です。ここまでは全員が取り組めるよう時間調整をお願いします。

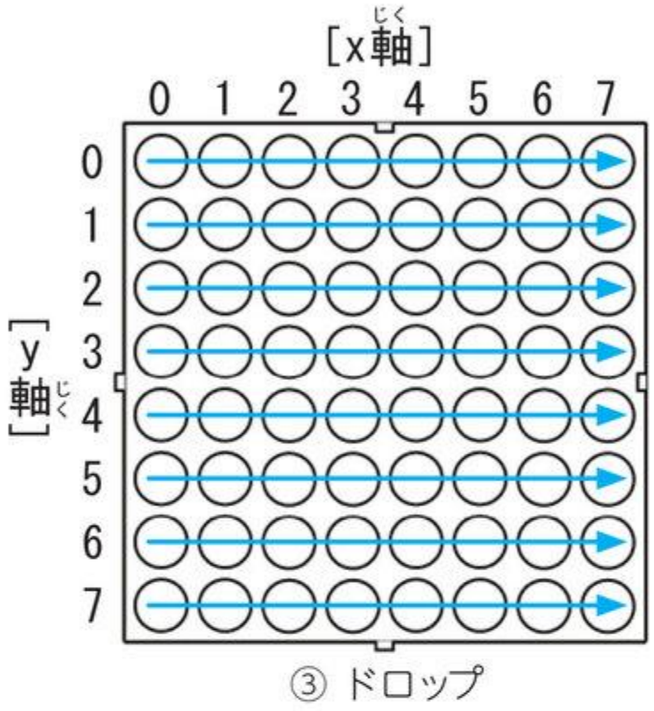
解答例は以下のプログラムです。
①ナイトライダー解答例：RoboticsProfessorCourse 1 ＞ MagicItemA1 ＞ LedMove3
②グルグル解答例：RoboticsProfessorCourse 1 ＞ MagicItemA1 ＞ LedAround

チャレンジ課題

プログラム「LedMove2」を変更し、以下のような動きをつくってみよう。

ドロップ

LEDが一番上の段で左から右へ動き、一番右まできたら一段下へ移りまた左から右へ動き、…というのを一番下の段までくり返す。



③ ドロップ

ヒント

左から右へ動くのにx座標が1ずつ増加、上から下まで動くのにy座標も1ずつ増加するね。文字がiだけでは足りないからもう1つ用意しよう。もちろん、for文も2つ必要だ。xとyで1を足すタイミングが異なるから、どこにfor文を書くかも重要だね。試行錯誤しながら正解を目指してみよう。

講

チャレンジ課題は、テキストの進行が早い生徒のための応用問題です。時間に余裕のある生徒は授業時間内に実施させてください。難度の高い問題が多いため、必ずしも全員がクリアする必要はありません。
解答プログラムは以下となります。
ドロップ解答例：RoboticsProfessorCourse1 > MagicItemA1 > LedMove4

3. 座標を使って絵を動かそう（目安40分）

3.0. 2進数について

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 > MagicItemA1 > MatrixSprite1

実行結果：スマイルマークが表示されます。
実行結果はプログラム「MatrixTest」と同じになりますが、プログラムがちがいます。どこがちがうかプログラムを見比べてみましょう。

プログラム「MatrixTest」より抜粋

```
myMatrix.write(1, 5, HIGH);
myMatrix.write(2, 2, HIGH);
myMatrix.write(2, 6, HIGH);
myMatrix.write(3, 6, HIGH);
myMatrix.write(4, 6, HIGH);
myMatrix.write(5, 2, HIGH);
myMatrix.write(5, 6, HIGH);
myMatrix.write(6, 5, HIGH);
```

プログラム「MatrixSprite1」より抜粋

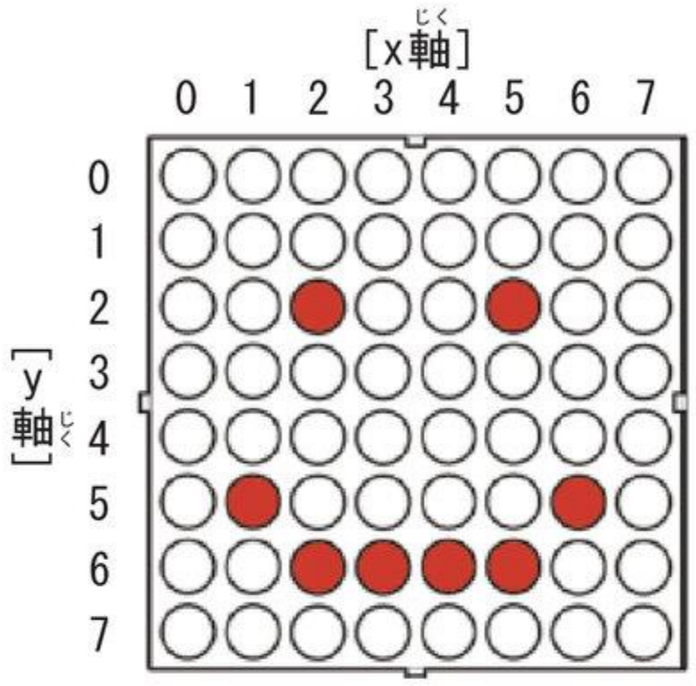
```
Sprite pattern1 = Sprite(
    8, 8,
    B00000000,
    B00000000,
    B00100100,
    B00000000,
    B00000000,
    B01000010,
    B00111100,
    B00000000
    );
```

まず左側のプログラム「MatrixTest」では myMatrix.write を使ってLEDが点灯（点滅）する座標を指定していました。
そして、右側のプログラム「MatrixSprite1」では青文字で書かれた B00100100 のような数字でLEDが点灯（点滅）する座標を指定しています。この暗号のような0と1の並びは「2進数」と呼ばれるもので、コンピューターが一番理解しやすいプログラムの形になります。

次のページでプログラム「MatrixSprite1」のプログラムがどのようにLEDの表示を命令しているのか説明していきます。

図 3-0 は、今表示されているスマイルマークです。表示している絵に 0 と 1 の 2 種類の数字を当てはめてみましょう。白丸は 0、赤丸は 1 と変換します。

〈座標系〉



〈2 進数〉

[x軸]

[y軸]	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	1	0
6	0	0	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0

図 3-0　パターン表示と 2 進数変換

絵を数字に変換すれば、その数字をそのままプログラムに使えそうですね。
プログラムには、`B00100100` のように最初に `B` と書かれていますが、これは 2 進数を使うときの約束ごとで「B= ビット」を表します。

プログラム「MatrixSprite1」より抜粋

```
Sprite pattern1 = Sprite(
    8, 8,
    B00000000,
    B00000000,
    B00100100,
    B00000000,
    B00000000,
    B01000010,
    B00111100,
    B00000000
    );
```

[x軸]

[y軸]	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	1	0
6	0	0	1	1	1	1	0	0
7	0	0	0	0	0	0	0	0

このように、同じ結果を表示するにも、命令の出し方が複数ありました。同じものをつくるのに何通りもの書き方があるのがプログラムの面白いところです。
では 2 進数の使い方を覚えたところで、今度は自分のつくった絵を表示させてみましょう。

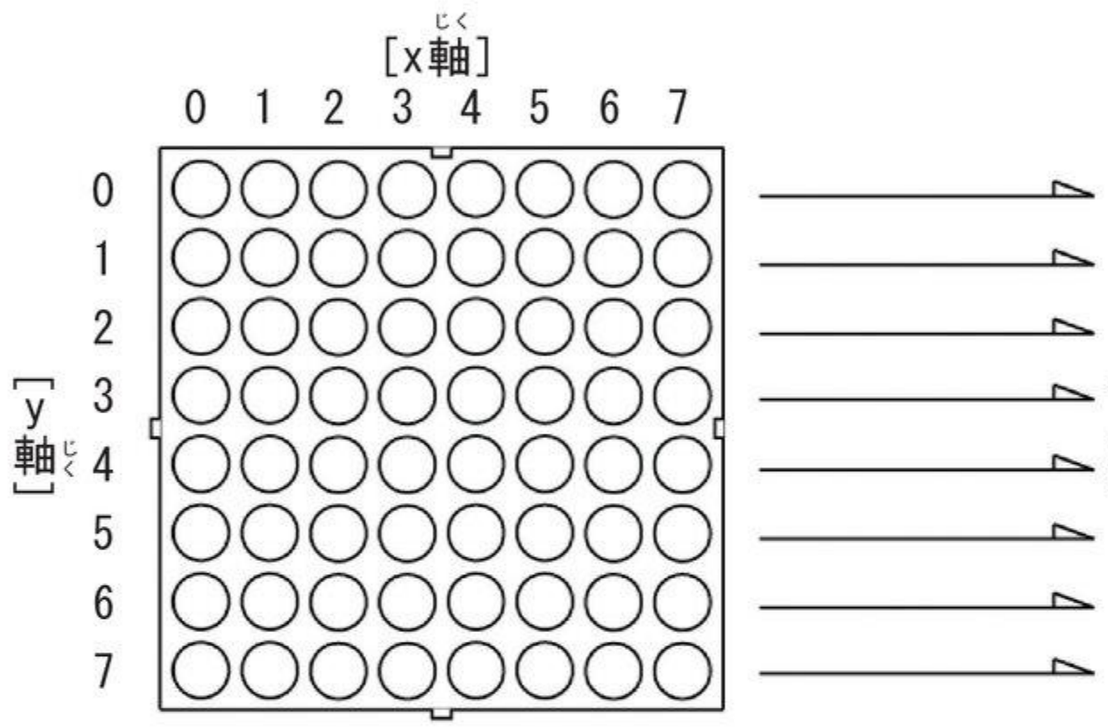
3.1. 好きなパターンを表示しよう

やってみよう！

プログラム「MatrixSprite1」を変更して、自分のつくった絵を表示させてみよう。
やり方は簡単。まずは、下の左側の図で点滅させたいところを塗りつぶしてみよう。
次に、右側の表に点滅させるところを「1」、消灯させるところを「0」として、数字を入れよう。
最後に、プログラムに変更する内容を入力して実行してみてね。

下の図を使って下書きをし、2種類のパターンをつくってみましょう。

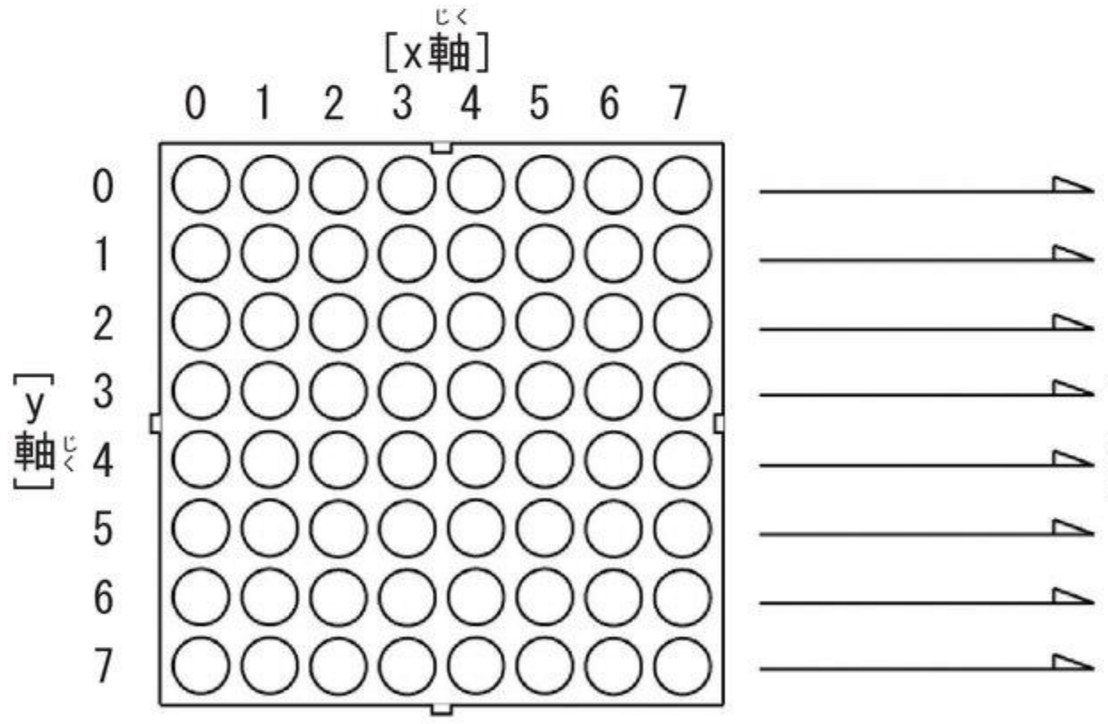
【パターン1】



[x軸]

[y軸]	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

【パターン2】



[x軸]

[y軸]	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

図3-1　オリジナルキャラを書いて表示してみる

3.2. アニメーションにして動かそう

今度は絵をアニメーションにして動かしてみましょう。以下の２つ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 > MagicItemA1 > MatrixSprite2

実行結果：絵が左から右に動いていきます。
絵の書き方は、さきほどの LED を動かすプログラムと同じですが、パターンが 2 つ書かれていますね。
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 > MagicItemA1 > MatrixSprite3

実行結果：絵がアニメーションになって動き出します。
パターン（絵）が２枚用意してあって、それをパラパラまんがのようにさしかえて表示しているのです。

【パターン１】

[x軸]

[y軸]	0	1	2	3	4	5	6	7
0	0	0	0	1	1	0	0	0
1	0	0	1	1	1	1	0	0
2	0	1	1	1	1	1	1	0
3	1	1	0	1	1	0	1	1
4	1	1	1	1	1	1	1	1
5	0	0	1	0	0	1	0	0
6	0	1	0	1	1	0	1	0
7	1	0	1	0	0	1	0	1

【パターン２】

[x軸]

[y軸]	0	1	2	3	4	5	6	7
0	0	0	0	1	1	0	0	0
1	0	0	1	1	1	1	0	0
2	0	1	1	1	1	1	1	0
3	1	1	0	1	1	0	1	1
4	1	1	1	1	1	1	1	1
5	0	0	1	0	0	1	0	0
6	0	1	1	0	0	1	1	0
7	0	1	0	1	0	0	1	0

図 3-2　プログラム「MatrixSprite2/3」

やってみよう！

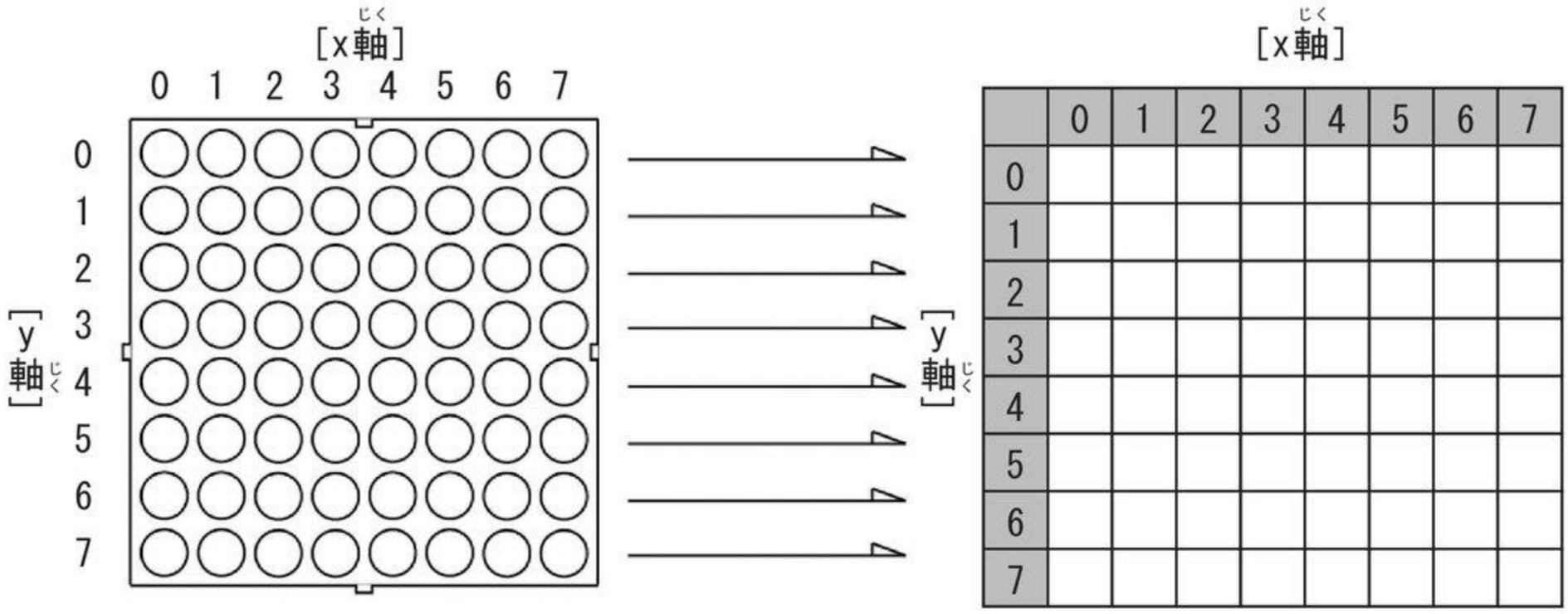
自分のオリジナルパターンをアニメーションにしてみよう。絵が動くように、以下のプログラムの数字の部分を変更してみよう。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA1 > MatrixSprite4

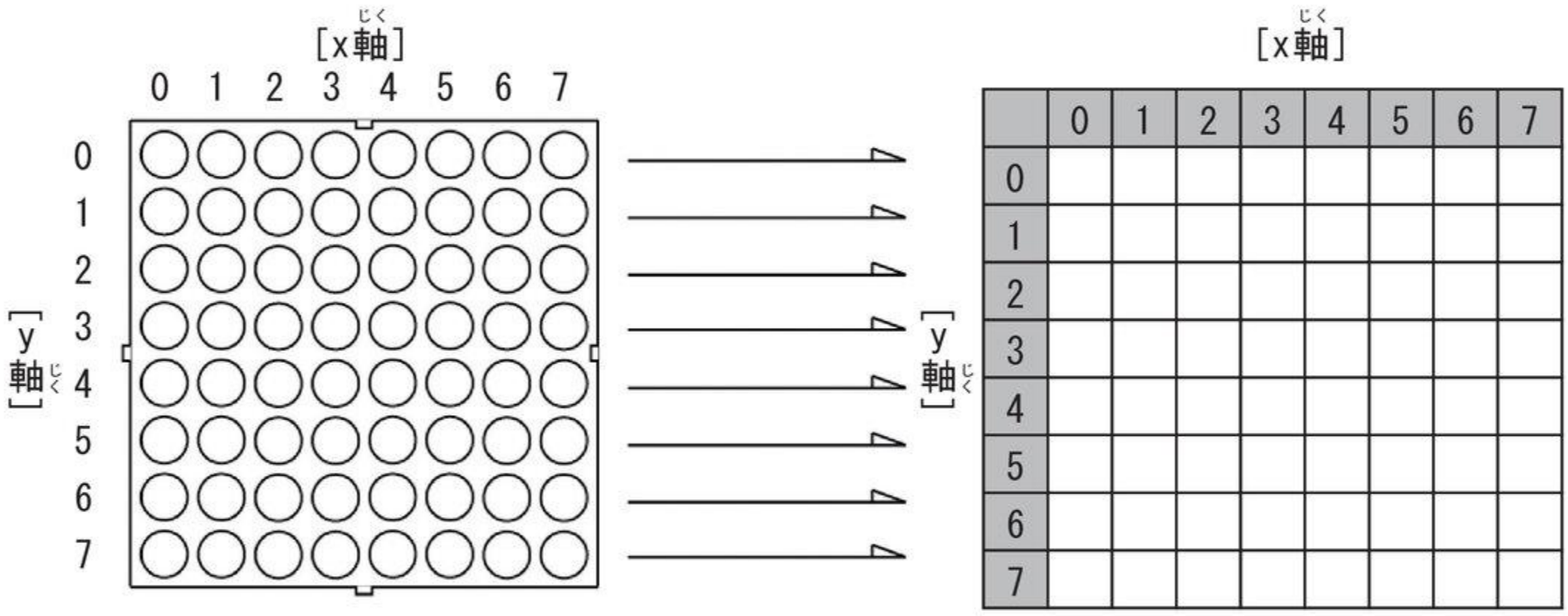
次ページの図を使って下書きをして、そのパターンをプログラムに入れて実行すると、自分のつくった絵がアニメーションしながら移動していくよ。

【パターン1（1コマ目）】



	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

【パターン2（2コマ目）】



	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

図 3-3　オリジナルキャラを書いてアニメーションにしてみる

下書きが終わったら、プログラム「MatrixSprite4」を変更してみましょう。

講

サンプルプログラムでは、Sprite pattern● = Sprite(8, 8, ▲); という命令を使い、2つの画像にそれぞれ pattern1 、pattern2 という名前をつけています。
また、void loop() 内では myMatrix.write を使い、「 pattern1 を点灯する」などといった命令を出しています。
もしアニメーションのコマ数を増やす場合、上記2か所の編集が必要になります。

座標を使った絵の動かし方、アニメーションのつくり方がわかってきたところで「応用編」のチャレンジ課題にトライしてみましょう。時間に余裕のある人は授業時間内にやってみましょう。

チャレンジ課題

① プログラム「MatrixSprite1」を変更して、スマイルマークを上下方向に動かしてみよう。

ヒント
図の表示位置が上や下に変化していくようなプログラムに修正してみよう。

② プログラム「MatrixSprite3」を変更して、アニメーションを３つにして動かしてみよう。

ヒント
pattern1、pattern2と同じように、pattern3を作成しよう。

プログラムをコピー＆ペーストしながらつくってみてね。

講 解答プログラムはそれぞれ以下となります。
① RoboticsProfessorCourse1 > MagicItemA1 > MatrixSprite_Challenge1
② RoboticsProfessorCourse1 > MagicItemA1 > MatrixSprite_Challenge2

3.3. 16進数について

2進数の使い方は理解できたかと思います。しかし、アニメーションを増やせば増やすほどデータが増えて大変です。これを短くできる方法があります。その方法が「16進数」を使ったプログラムです。
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 > MagicItemA1 > MatrixSprite5

実行結果：プログラム「MatrixSprite1」と同じ実行結果になります。
プログラムを見てみると、データの部分が違います。

プログラム「MatrixSprite1」より抜粋（ばっすい）

```
Sprite pattern1 = Sprite(
    8, 8,
    B00000000,
    B00000000,
    B00100100,
    B00000000,
    B00000000,
    B01000010,
    B00111100,
    B00000000
    );
```

プログラム「MatrixSprite5」より抜粋（ばっすい）

```
Sprite pattern1 = Sprite(
    8, 8,
    0x00,
    0x00,
    0x24,
    0x00,
    0x00,
    0x42,
    0x3C,
    0x00
    );
```

頭に「B」とついているのが 2 進数ですが、「0x」とついているのが 16 進数を表します。
10 進数と 2 進数と 16 進数は、次のように対応しています。

表 3-0　10 進数と 2 進数と 16 進数の対応表

10 進数	2 進数	16 進数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

10 進数は日常よく使っているからわかりますね。2 進数は 0 と 1 の数字 2 つですべての数字を表します。16 進数はすべての数字を 16 個の数字（実際はアルファベットが入ってくる）を使って表すものです。この 16 進数は、2 進数の 4 けたまでを、わずか 1 けたに圧縮（あっしゅく）でき、2 進数と非常に相性が良いために、プログラムでよく使用されます。
プログラム「MatrixSprite5」のデータは**表 3-0** を見れば、変換（へんかん）されていることがわかります。たとえば、B00111100 は、上 4 けたと下 4 けたに分けて、0011, 1100 とすると、
上 4 けた 0011 → 3
下 4 けた 1100 → C
となります。上下つなぎあわせて、
B00111100 → 0x3C
となりますね。

やってみよう！

さきほどつくった自分のオリジナルパターンを 16 進数に変換（へんかん）して、同じように表示できるかやってみよう！

コラム　10進法と2進法

●10進法

ふつう数を数えるときは0,1,2,3,4,5,6,7,8,9と数字が増えていき、9の次は十の位が加わって「10」となりますね。このような数え方を「10進法」といいます。数字を10個使い切ったら次の位に進むからです。

位が2つになると、十の位にも一の位にもそれぞれ0～9の10個ずつ数字が使えますから、つくれる数は10×10（10^2と表します）で100個（0～99）になります。そして、次の数である「100」からは位がもう1つ必要になるわけです。1つ位が増えれば、百の位にも10通りの数字が使えるので10×10×10（10^3）で1000個（0～999）がつくれます。次の数である「1000」になるとまた位が増えます。つまり、10進法では、数が10^1、10^2、10^3…になるごとに位が増えていく、ということになります。

この性質をうまく利用すると、大きい数を分解して考えられるようになります。

2976　＝**2**×1000　＋**9**×100　＋**7**×10　＋**6**×1

＝**2**×10^3　＋**9**×10^2　＋**7**×10^1　＋**6**×1

小学校の算数でもこのような式をつくった記憶がありませんか。

「こんな式、当たり前じゃないか」と思うかもしれませんが、この知識がロボットの世界を考えるうえで非常に重要なのです。

●2進法

さて、意外なことに、コンピューターは「0」と「1」という2つの数字しか用いることができないのです。よって､コンピューターの数の数え方は「10進法」ではなく「2進法」となります。

2進法では、1けたで表すことができるのは0と1の2つだけです。その次の数は、10進法では「2」と表されますが、2進法にはその数字が存在しないため、ここで位を1つ増やし「10」と表さなければなりません。10進法では数が10^1、10^2、10^3、・・・になるごとに位が増えていきましたが、2進法では数が2^1、2^2（2×2＝4)、2^3（2×2×2＝8)、…になるごとに位が増えていきます。

先ほど、10進法の数を各位の数字ごとに分解しましたね。同じルールで、2進法の数字を分解してみましょう。たとえば、2進法で「**1101**」と表される数の場合、次のようになります。

1101　＝**1**×2^3　＋**1**×2^2　＋**0**×2^1　＋**1**×1

＝**1**×8　＋**1**×4　＋**0**×2　＋**1**×1＝13

この計算で、2進法で「1101」という数は、10進法での「13」という数と等しいことがわかりました。

このように、かけ算とたし算を組みあわせることで、2進法で表されている数を10進法に変換できます。

逆に、10進法の数を2進法に変換していく場合は、2で割っていくことになります。

たとえば10進法で「**11**」という数を2進法に変換してみましょう。

11÷2＝5あまり**1**

5÷2＝2あまり**1**

2÷2＝1あまり**0**

1÷2＝0あまり**1**

このように、商が0になるまで2で割っていき、出てきたあまりを逆の順番に並べていくことで2進法の数に変換できます。よって、11を2進法で表すと「**1011**」となります。

やってみよう！

10進数で25という数は、2進数では何になるかな？

25 ÷ 2=12 あまり 1、12 ÷ 2=6 余り 0、6 ÷ 2=3 あまり 0、3 ÷ 2=1 あまり 1、

1 ÷ 2=0 あまり 1、よって、2進法では 11001 となる。

コラム　16進法

先ほど学んだとおり、コンピューターは2進法を用いて計算をしています。しかし2進法は少し数が大きくなるだけでけた数が大幅に増えてしまうため、人間が読んだり書いたりするときは非常に不便です。そのため、プログラミングで大きい数字をたくさん取り扱うときなどは「16進法」を用いることがあります。2進法でけたを4つ使うと $2^4 = 16$ 通りの数がつくれますが、これを1けたで表せるようにしたものです。

16進数は0から9までの数字とAからFまでのアルファベットを使って数を表現します。数は、0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F と順に増え、次に位が増えて10になります。

アルファベットはそれぞれ10進数で、

A → 10、B → 11、C → 12、D → 13、E → 14、F → 15　を表します。

このようにして、16進数は 16^0（1）、16^1（16）、16^2（256）、16^3（4096）…と位がくり上がります。

たとえば、4E5F という16進数は、数式で以下のように表すことができます。

$\mathbf{4} \times 16^3 + \mathbf{E} \times 16^2 + \mathbf{5} \times 16^1 + \mathbf{F} \times 16^0$

$= \mathbf{4} \times 4096 + \mathbf{E}（14） \times 256 + \mathbf{5} \times 16 + \mathbf{F}（15） \times 1 = 20063$（10進法）

逆に、10進数から16進数へ変換（へんかん）するには、10進数を16で割って、その商をさらに16で割る、またその商を16で割って…と、余りを出しながら、商が0になるまでくり返します。

そして**最後の余りを先頭に下から順に並べます。**

たとえば10進数で1000という数は、以下のように計算することができます。

1000 ÷ 16=62 あまり 8、62 ÷ 16=3 あまり 14（E）、3 ÷ 16=0 あまり 3

よって16進法では3E8となります。

やってみよう！

10進数で2000という数は、16進数では何になるかな？

2000 ÷ 16=125 あまり 0、125 ÷ 16=7 あまり 13 (D)、7 ÷ 16=0 あまり 7、

よって、16進法では 7D0 となる。

表 3-1 は、0 から 255 までの 10 進数に対応する 16 進数 (00 ～ FF) の一覧表です。10 進数の右側が対応する 16 進数になります。

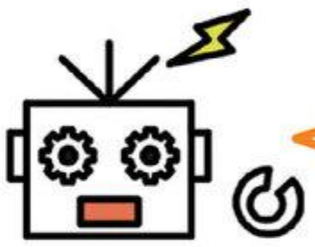
表 3-1　10 進数と 16 進数の対応表

10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数	10進数	16進数
0	00	1	01	2	02	3	03	4	04	5	05	6	06	7	07	8	08	9	09
10	0A	11	0B	12	0C	13	0D	14	0E	15	0F	16	10	17	11	18	12	19	13
20	14	21	15	22	16	23	17	24	18	25	19	26	1A	27	1B	28	1C	29	1D
30	1E	31	1F	32	20	33	21	34	22	35	23	36	24	37	25	38	26	39	27
40	28	41	29	42	2A	43	2B	44	2C	45	2D	46	2E	47	2F	48	30	49	31
50	32	51	33	52	34	53	35	54	36	55	37	56	38	57	39	58	3A	59	3B
60	3C	61	3D	62	3E	63	3F	64	40	65	41	66	42	67	43	68	44	69	45
70	46	71	47	72	48	73	49	74	4A	75	4B	76	4C	77	4D	78	4E	79	4F
80	50	81	51	82	52	83	53	84	54	85	55	86	56	87	57	88	58	89	59
90	5A	91	5B	92	5C	93	5D	94	5E	95	5F	96	60	97	61	98	62	99	63
100	64	101	65	102	66	103	67	104	68	105	69	106	6A	107	6B	108	6C	109	6D
110	6E	111	6F	112	70	113	71	114	72	115	73	116	74	117	75	118	76	119	77
120	78	121	79	122	7A	123	7B	124	7C	125	7D	126	7E	127	7F	128	80	129	81
130	82	131	83	132	84	133	85	134	86	135	87	136	88	137	89	138	8A	139	8B
140	8C	141	8D	142	8E	143	8F	144	90	145	91	146	92	147	93	148	94	149	95
150	96	151	97	152	98	153	99	154	9A	155	9B	156	9C	157	9D	158	9E	159	9F
160	A0	161	A1	162	A2	163	A3	164	A4	165	A5	166	A6	167	A7	168	A8	169	A9
170	AA	171	AB	172	AC	173	AD	174	AE	175	AF	176	B0	177	B1	178	B2	179	B3
180	B4	181	B5	182	B6	183	B7	184	B8	185	B9	186	BA	187	BB	188	BC	189	BD
190	BE	191	BF	192	C0	193	C1	194	C2	195	C3	196	C4	197	C5	198	C6	199	C7
200	C8	201	C9	202	CA	203	CB	204	CC	205	CD	206	CE	207	CF	208	D0	209	D1
210	D2	211	D3	212	D4	213	D5	214	D6	215	D7	216	D8	217	D9	218	DA	219	DB
220	DC	221	DD	222	DE	223	DF	224	E0	225	E1	226	E2	227	E3	228	E4	229	E5
230	E6	231	E7	232	E8	233	E9	234	EA	235	EB	236	EC	237	ED	238	EE	239	EF
240	F0	241	F1	242	F2	243	F3	244	F4	245	F5	246	F6	247	F7	248	F8	249	F9
250	FA	251	FB	252	FC	253	FD	254	FE	255	FF	---	--	---	--	---	--	---	--

4. まとめ（目安５分）

今回の授業では、マトリクス LED という電子部品を使って、絵やアニメーションを表示させるプログラムを学習しました。また、2 進数、16 進数については、今後の授業でもたくさん活用します。少しずつでよいので、身につけていきましょう。

マトリクス LED は、さまざまなロボットに取り付けて、文字やキャラクターを表示させるために使っていきます。ピンヘッドを曲げないように大切に保管しましょう。



次回はタッチセンサーとスピーカーの使い方を学んでいきます。
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。

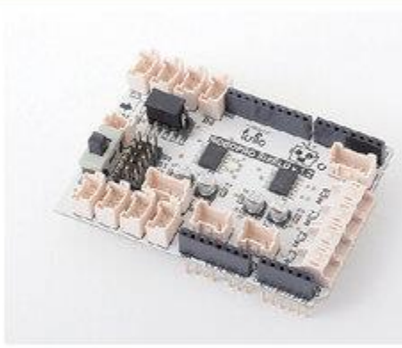
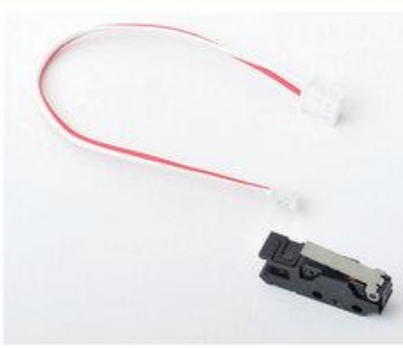
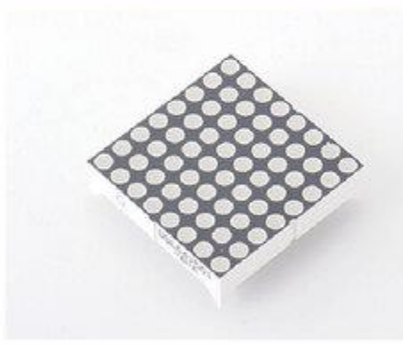
USB ケーブル 1	マイコンボード 1	ロボプロシールド 1	タッチセンサー 2
			
マトリクス LED シールド 1	マトリクス LED 1	スピーカー 1	
			

図 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。
　・座標を使いこなす
　・for 命令の使い方をマスターする
　・2 進数と 16 進数をマスターする
○　次回のテーマは「タッチセンサーとスピーカーで遊ぶ」であることを告知します。

P.12　やってみよう！　解答例（void loop 以下を抜粋）

```
void loop(){
    myMatrix.write(0, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(0, 0, LOW);   //LEDを消灯させる
    myMatrix.write(1, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(1, 0, LOW);   //LEDを消灯させる
    myMatrix.write(2, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(2, 0, LOW);   //LEDを消灯させる
    myMatrix.write(3, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(3, 0, LOW);   //LEDを消灯させる
    myMatrix.write(4, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(4, 0, LOW);   //LEDを消灯させる
    myMatrix.write(5, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(5, 0, LOW);   //LEDを消灯させる
    myMatrix.write(6, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(6, 0, LOW);   //LEDを消灯させる
    myMatrix.write(7, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(7, 0, LOW);   //LEDを消灯させる
    myMatrix.write(7, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(7, 0, LOW);   //LEDを消灯させる
    myMatrix.write(6, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(6, 0, LOW);   //LEDを消灯させる
    myMatrix.write(5, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(5, 0, LOW);   //LEDを消灯させる
    myMatrix.write(4, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(4, 0, LOW);   //LEDを消灯させる
    myMatrix.write(3, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(3, 0, LOW);   //LEDを消灯させる
    myMatrix.write(2, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(2, 0, LOW);   //LEDを消灯させる
    myMatrix.write(1, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(1, 0, LOW);   //LEDを消灯させる
    myMatrix.write(0, 0, HIGH);  //LEDを点灯させる
    delay(100);
    myMatrix.write(0, 0, LOW);   //LEDを消灯させる

}
```