

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4回

でんしかいろ

電子回路とプログラミング
(7セグメントLED)

講師用

目 次

0. 電子回路とプログラミング（7セグメントLED）

0.0. 「電子回路とプログラミング（7セグメントLED）」でやること

0.1. 必要なもの

1. 7セグメントLED を使ってみる

1.0. 7セグメントLED

1.1. シールドの組み立てと数字の表示

1.2. 7セグメントLED を細かく使う

1.3. 数字をカウントする

2. センサーを使う

2.0. ボリュームセンサーの値を確認する

2.1. 超音波距離センサーを活用する

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。

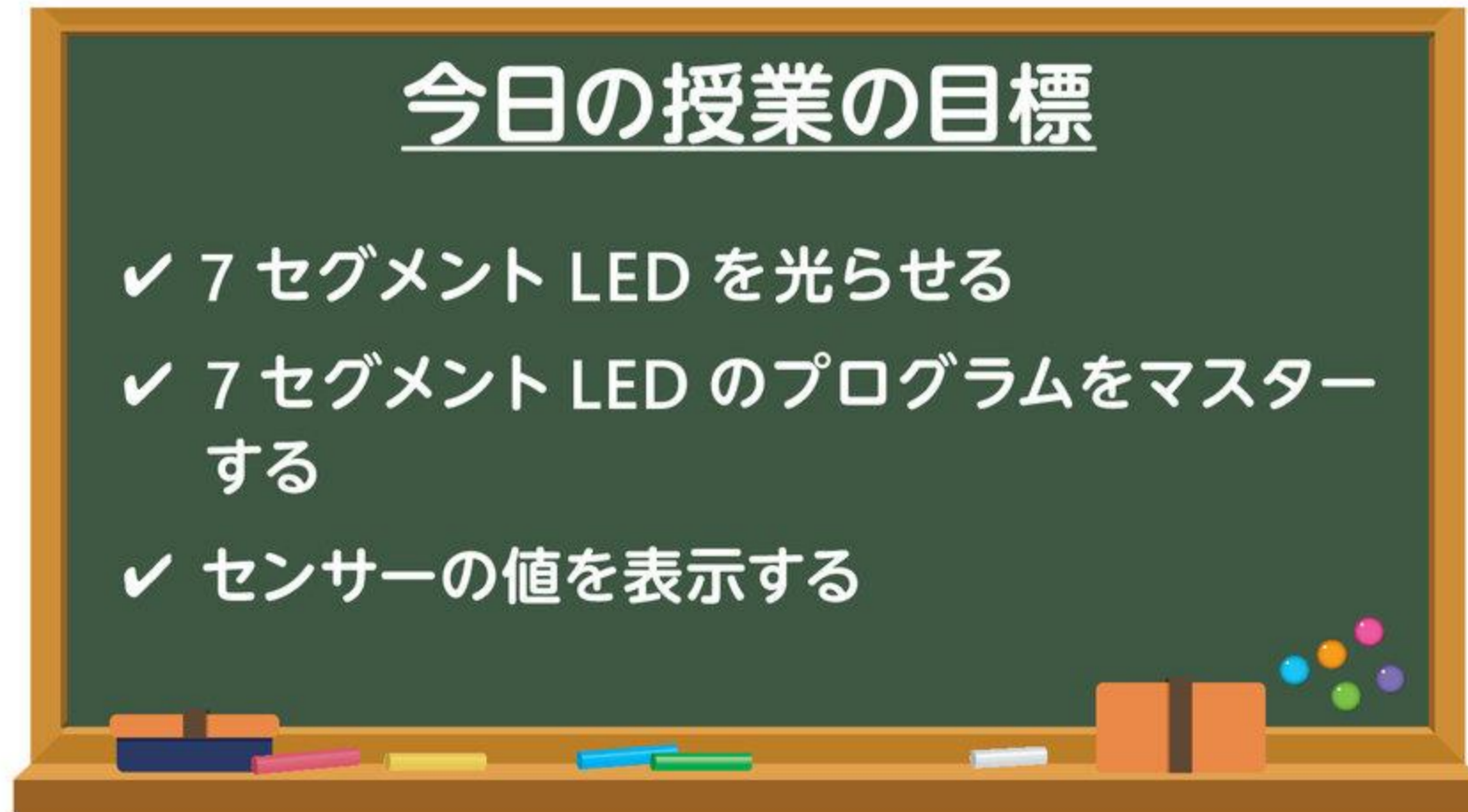
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

（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. ^{でんしかいろ}電子回路とプログラミング (7セグメントLED) (目安10分)

0.0. ^{でんしかいろ}「電子回路とプログラミング (7セグメントLED)」でやること

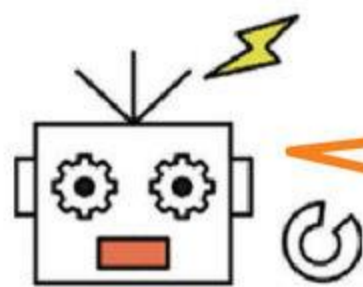


今回の授業では「7セグメントLED」を使いこなせるようになりましょう！

「7セグメントLED」とは「7つに分かれているLED」のことです。1つのLEDでは数字は表示できませんが、7つあれば、0～9までの数字を表現することができます。このパーツは非常に歴史が古く、「真空管^{しんくつかん}」と呼ばれていたものが、「半導体^{はんどうたい}」というものに置きかわる時期にはすでにありました。

昔は、コンピューターで計算することはできても、結果を見ることができず、紙に穴を開けて、それを読み取っていました。ちょうど点字^{てんじ}のようなイメージです。それを数字にして表示させたのが「7セグメントLED」です。マイコンが、「今何を考えているのか？」を表示させて、プログラムがきちんと動いているかを確認するにはもってこいのパーツなのです。

今回は、4つの数字を表示できる7セグメントLEDを用意しました。まずは、これを使いこなすところから始めましょう！



7セグメントLEDでいろんな表示をマスターしよう。

0.1. 必要なもの

以下のパーツを準備しておきましょう。



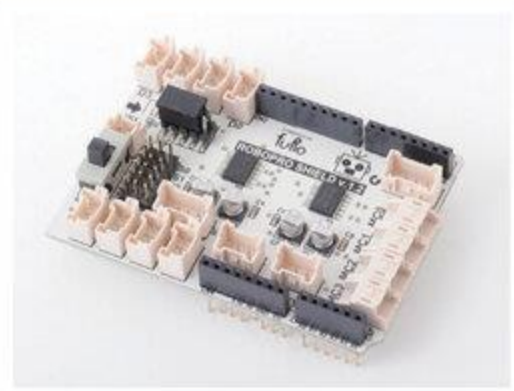
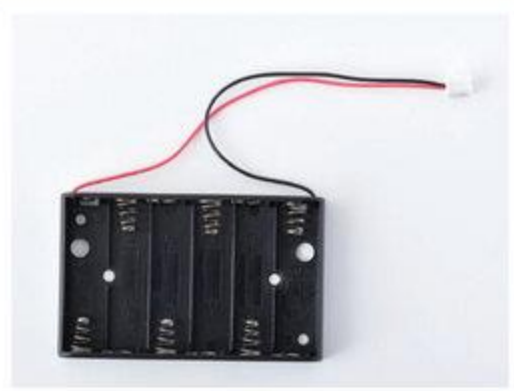


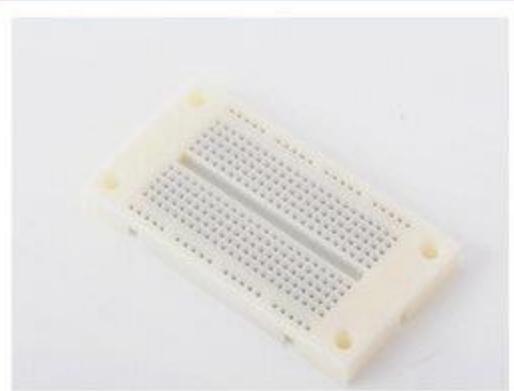


USB ケーブル	1	マイコンボード	1	ロボプロシールド	1	電池ボックス	1
							
マトリクスLED シールド	1	超音波距離センサー	1	センサーケーブル	1	301 ブレッドボード	1
							
ジャンパー線	65	7セグメントLED	1	可変抵抗ボリューム	2		
							

図 0-0 必要なもの

1. 7セグメントLEDを使ってみる (目安70分)

1.0. 7セグメントLED

さて、突然ですが「デジタル時計」を思い出してみましょう。
細長い六角形などの棒が縦横に組み合わさっていますね。この棒のことを「セグメント」といいます。

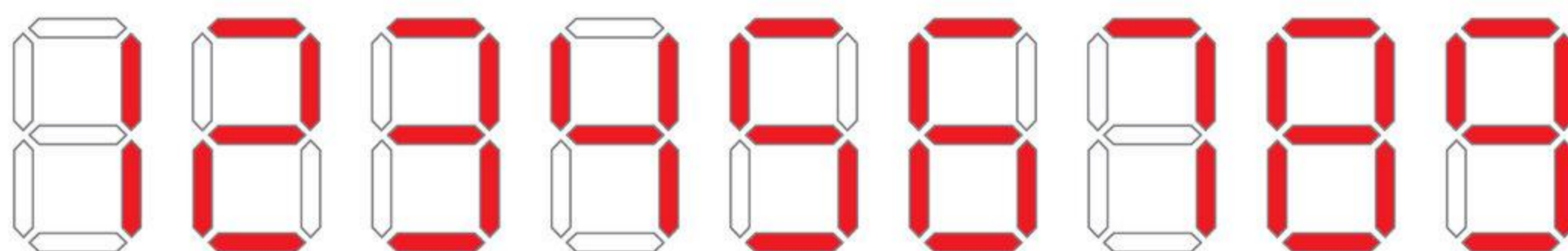


図1-0 デジタル時計の数字表示

数字を1つ表すのに、セグメントは7つ必要ですね。そのため、このような表示ができる装置を「7セグメントディスプレイ」などとよびます。

ロボプロで使用するものは、1つひとつのセグメントが別々のLEDになっているので「7セグメントLED」とよんでいます。7セグメントLEDは「このLEDとこのLEDだけ点灯！」などと命令することで思い通りの表示をさせることができます。ちなみに、7セグメントLEDには数字の表示の7つのセグメントに加え、小数点として使える丸型のセグメントがあります（「8セグメントではないか！」と言われてしまうかもしれませんね）。

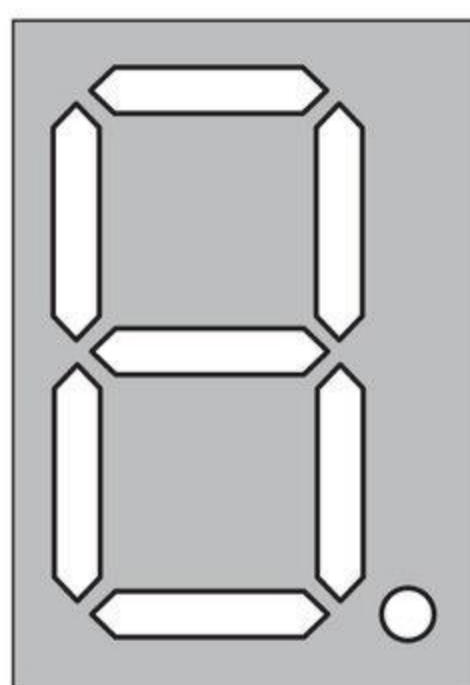


図1-1 7セグメントLED

「特定のLEDだけ点灯させて、ねらった形をつくる」というやり方はマトリクスLEDと似ていますね。詳しくは後ほど説明しますが、プログラムの考え方もマトリクスLEDと似ています。

7セグメントLEDもマトリクスLEDと同じように、マトリクスLEDシールドに接続します。まずは、7セグメントLEDを取り付けてみましょう。

1.1. シールドの組み立てと数字の表示

1) 組み立て

<組み立て手順①>

まず、7セグメントLEDとマトリクスLEDシールドを組み立てます。各パーツの上下の向きに気を付けましょう。マトリクスLEDシールドのソケットと7セグメントLEDの小数点の位置を目印に向きをそろえます。ピンと穴がズレないように、注意深く差し込みましょう。なお、7セグメントLEDは正しく接続しても、ピンが完全に奥まで差し込めるわけではありません。図1-3程度に差し込めていれば大丈夫です。



図1-2 7セグメントLEDとマトリクスLEDシールドの組み立て

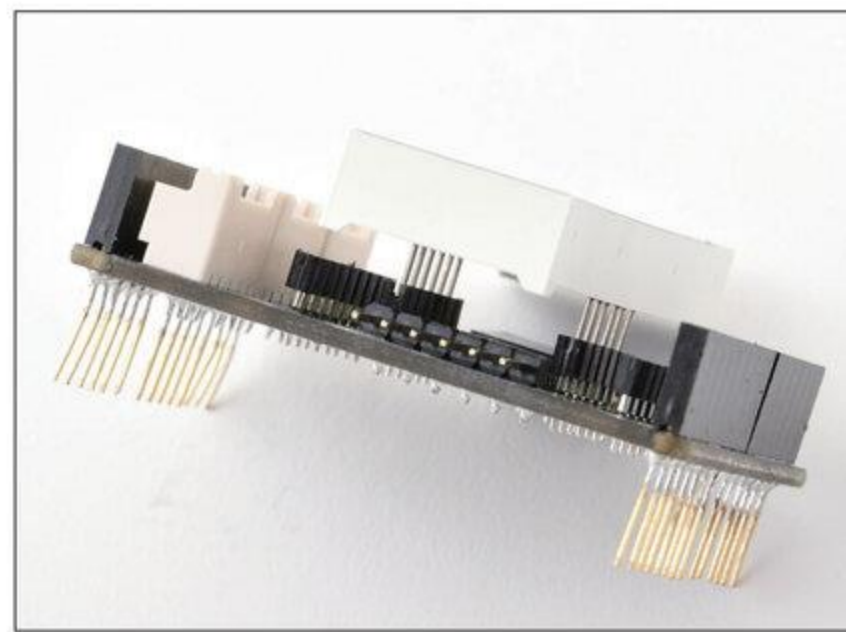


図1-3 7セグメントLEDのピンの差し込み具合

講

7セグメントLEDのピン形状・特性は、マトリクスLEDのそれと同じですので、マトリクスLEDの組み立てのヒント(1年目 7月1回目 P.4の「POINT」)を、必要に応じて生徒にご共有をお願いします。

<組み立て手順②>

次に、マトリクスLEDシールドをロボプロシールドとマイコンボードに接続します。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、7セグメントLEDの順になるように組み立てます。ここでもマトリクスLEDシールドのソケットの位置で接続の向きを確認しましょう。

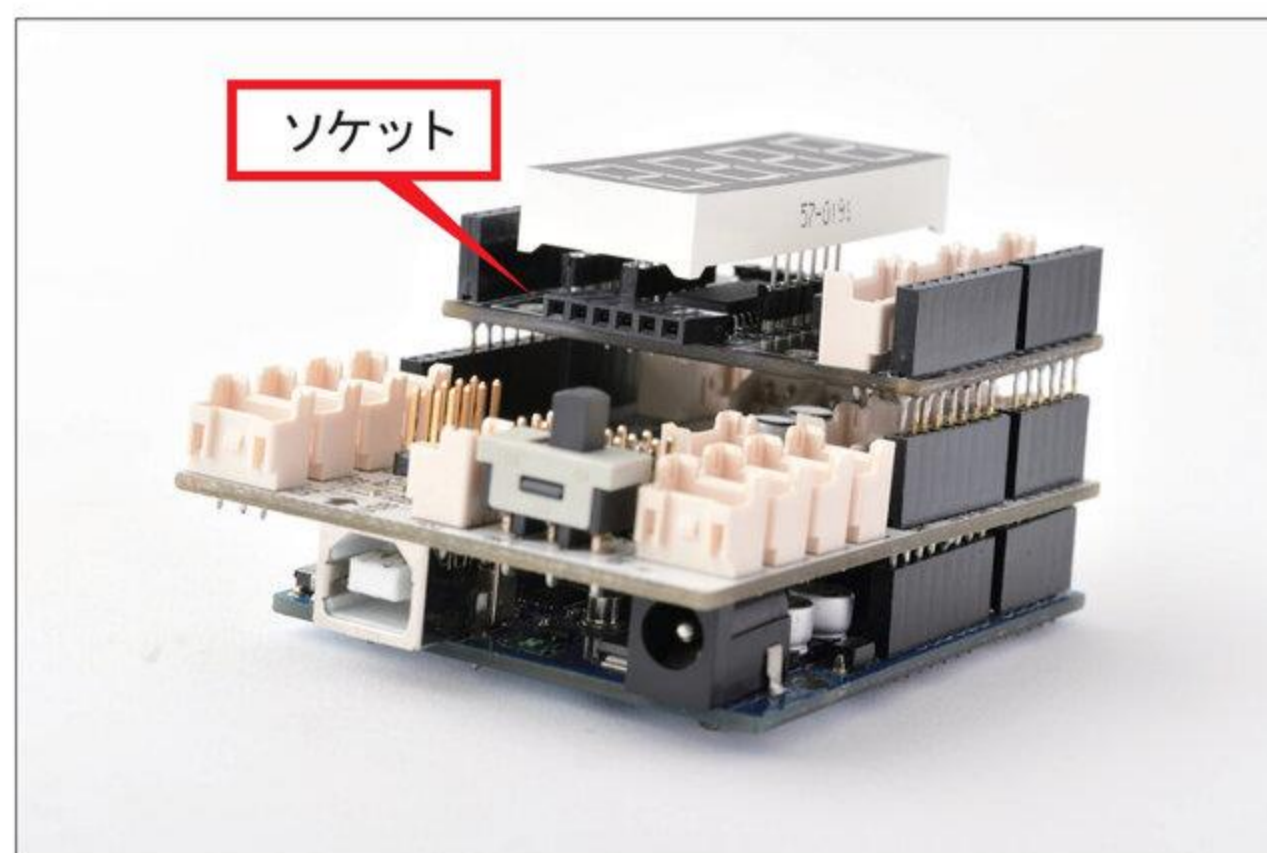


図1-4 接続完成図

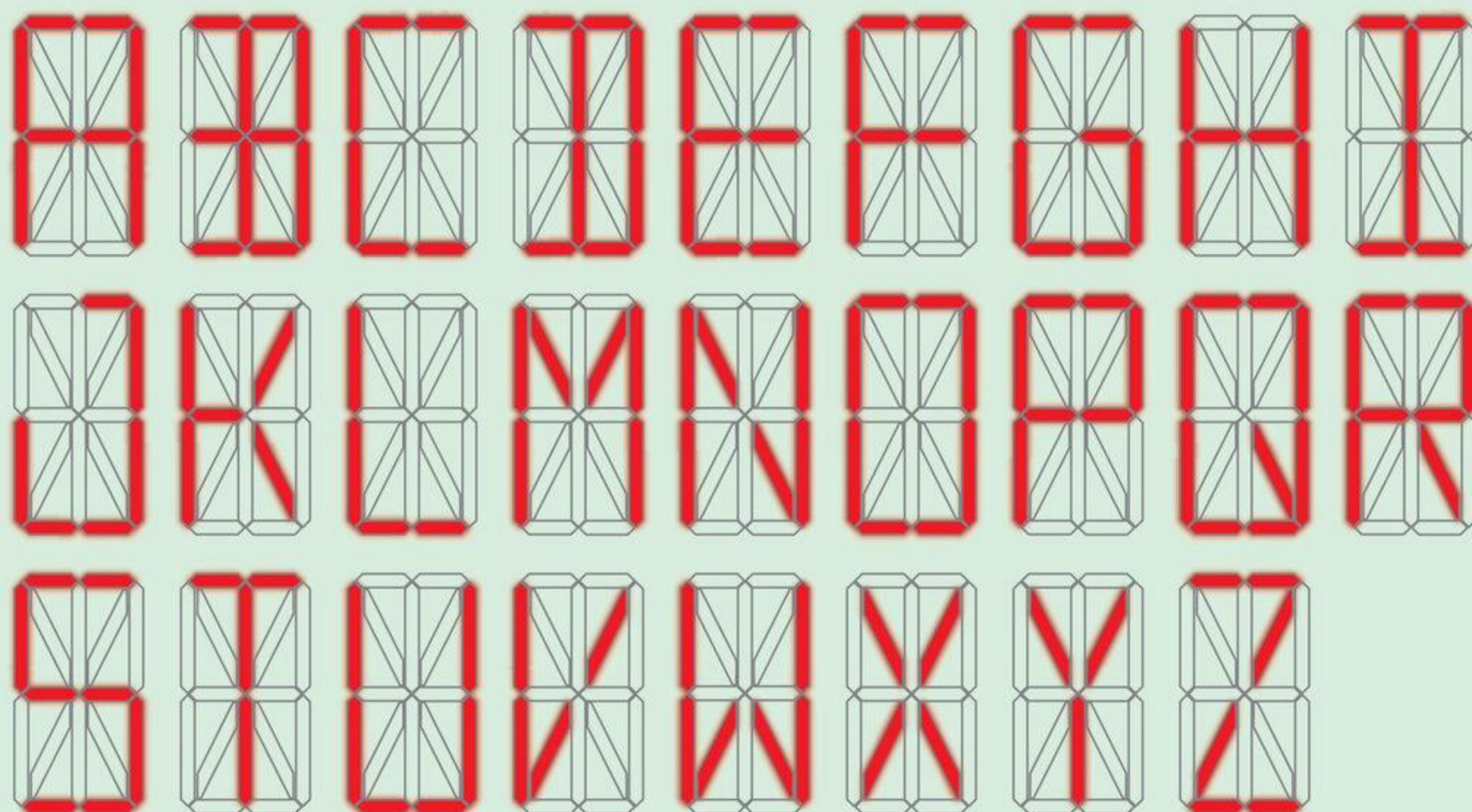
組み立てが完了したら、USBケーブルでパソコンとつないでください。



豆知識

セグメントが7つだと、数字は問題なく表すことができますが、アルファベットなどは難しいですね。ためしに「A」と「R」をつくってみると、まったく同じ形になってしまいます。

この問題を解消した「16セグメントディスプレイ」というものも存在します。



2) 動作確認

では動作を確認します。以下のプログラムを実行しましょう。

🔄 プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA4 > _7segTest

実行結果：7セグメントLEDに数字の「0123」が表示される。

7セグメントLEDへの命令を、大きく3つに分けて見てみましょう。

📄 プログラム「_7segTest」より抜粋 ばっすい

```
#include <LedControl.h>

LedControl lc = LedControl(11, 13, 1); //7セグメントLEDを使うときのオマジナイ

void setup(){
  lc.shutdown(0, false); //7セグメントLEDをリセット
  lc.setIntensity(0, 4); //色の濃さ(0-8)
  lc.clearDisplay(0); //表示クリア
}

void loop(){
  lc.setDigit(0, 0, 0, false);
  lc.setDigit(0, 1, 1, false);
  lc.setDigit(0, 2, 2, false);
  lc.setDigit(0, 3, 3, false);
}
```

緑色LEDやフルカラーLEDを使っていたときは「とにかく〇番ピンから電流を流す！」というシンプルな命令でも制御ができたのですが、7セグメントLEDはさすがにそうはいきません。「7セグメントLED専用の命令」というのが必要になります。

そのため、「7セグメントLED専用の命令がたくさん載っている説明書」のようなプログラムをあらかじめつくっておき、マイコンボードには「まずその説明書を読んで！」という命令を出します。これが、赤枠の部分にあたります。

📄 プログラム「_7segTest」より抜粋 ばっすい

```
#include <LedControl.h>
```

`LedControl.h` というのが説明書プログラムの名前です。これによって、7セグメントLED専用のさまざまな命令が使えるようになりました。

たとえば、緑枠の部分では7セグメントLEDに `lc` という名前をつけ、どのピンに接続するかを指定しています。さらに、7セグメントLEDを使う準備として `setup()` の中ですべてのLEDをオフにしたり、LEDの光る強さ（輝度）を設定したりしています。

□ プログラム「_7segTest」より^{ぼっすい}抜粋

```
LedControl lc = LedControl(11, 13, 1); //7セグメントLEDを使うときのオマジナイ

void setup(){
  lc.shutdown(0, false); //7セグメントLEDをリセット
  lc.setIntensity(0, 4); //色の濃さ(0-8)
  lc.clearDisplay(0); //表示クリア
}
```

命 令 「clearDisplay」

動作内容：7セグメントLEDのすべての表示を消す

使 い 方：`lc.clearDisplay(0);` // `lc` と名付けた7セグメントLEDの表示を消す

赤枠と緑枠の部分は、7セグメントLEDを使うときはいつも同じ文をかくことが多いです。今後「7セグメントLEDを新たに使いたい!」という場面では、このプログラムをお手本にするとよいでしょう。

最後に、青枠の部分で、7セグメントLEDの光り方を指定しています。

□ プログラム「_7segTest」より抜粋 ぼっすい

```
void loop(){  
  lc.setDigit(0, 0, 0, false);  
  lc.setDigit(0, 1, 1, false);  
  lc.setDigit(0, 2, 2, false);  
  lc.setDigit(0, 3, 3, false);  
}
```

命 令 [setDigit]

動作内容：7セグメントLEDの指定したケタに、指定の数字を表示させる

使 い 方：lc.setDigit(0, 2, 1, false);

// lcと名付けたLEDの2ケタ目に「1」を表示させる

講

第1引数(カッコ内のはじめの数値)は画面番号を指しますが、ロボプロでは画面が1つしか存在しませんので0で固定です。

この7セグメントLEDには4ケタ分の数字が表示できるので、左から「0ケタ目」「1ケタ目」…としています。

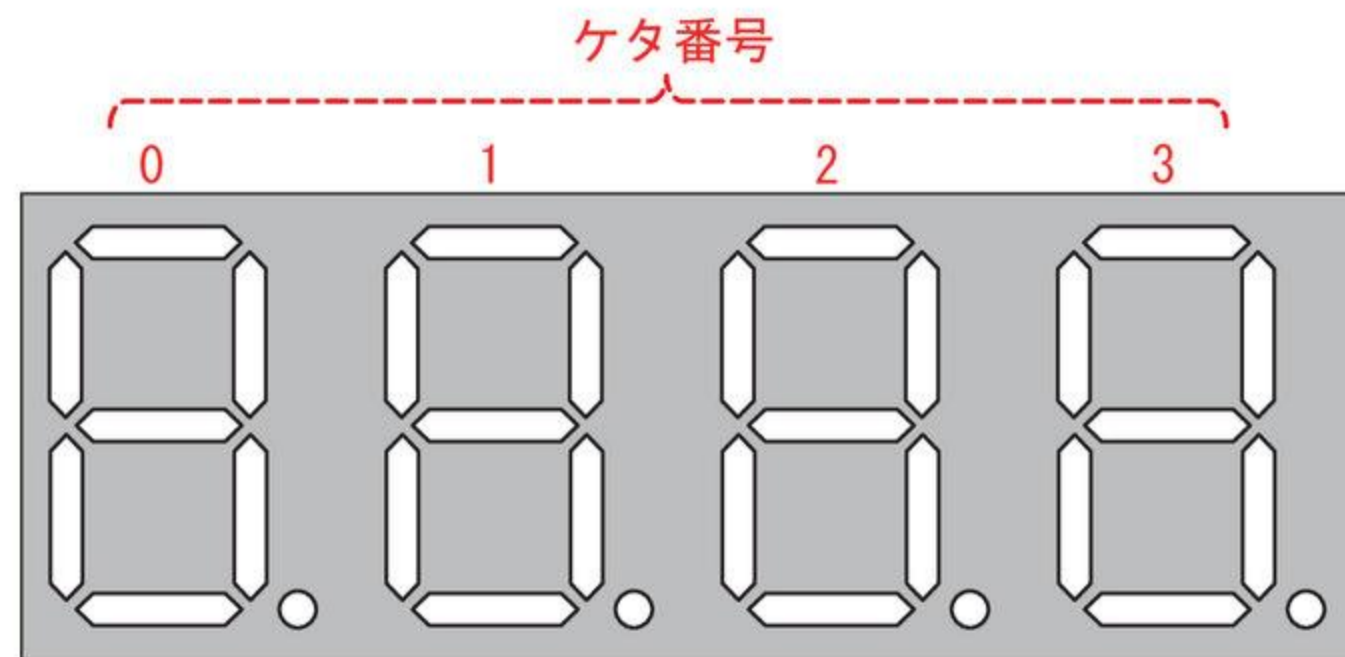
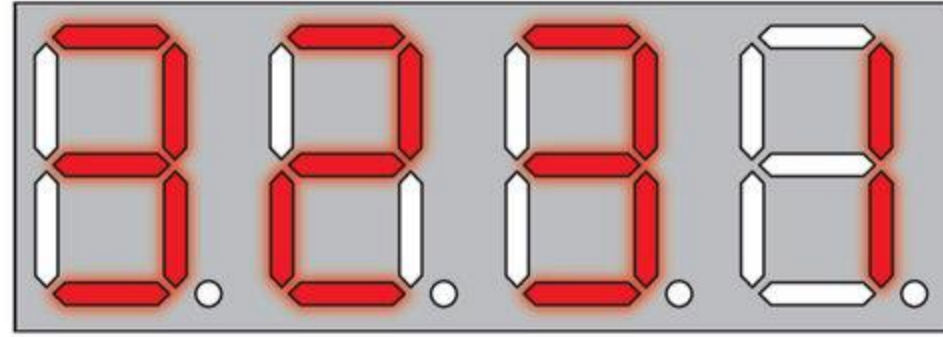


図1-5 7セグメントLEDのケタ番号

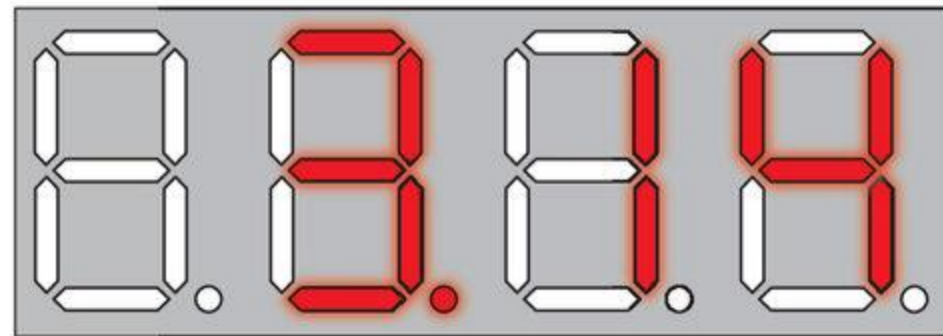
やってみよう！

プログラム「_7segTest」をかきかえ、以下のように表示を^{へんこう}変更してみよう！

① 3231



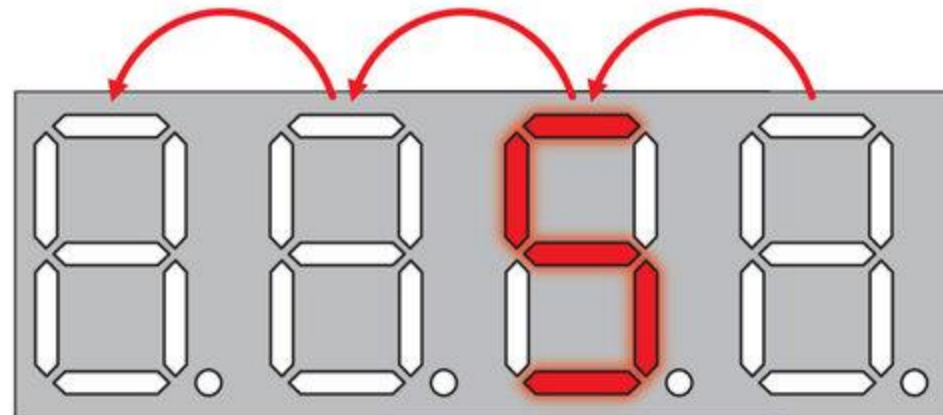
② 3.14



💡 ヒント

小数点の部分を光らせる命令も、`setDigit` の中に入っているよ！ 色々とかきかえてみてね。

③ 「5」 という表示が右から1ケタずつ移動していく



講

- ①②
解答例は巻末に記載します。`setDigit` 内の `false` を `true` に変更すると小数点が点灯します。
- ③
解答例は以下のプログラムです。
RoboticsProfessorCourse2 > MagicItemA4 > _7segChallenge1
数字が移動するごとに `clearDisplay` を挟む必要があります (前に表示していた部分が残ってしまう為)。
また、解答例は for 文を活用し、点灯命令を1つにまとめています。
単に点灯命令を4つかくだけでも狙い通りの動作にはなりますが、よりシンプルにまとめたプログラムを考える力は常に必要になります。さらに今回の後半で for 文を活用する課題が登場しますので、なるべくこちらの答えに行きつくよう誘導してあげてください。

1.2. 7セグメントLEDを細かく使う

数字を好きな場所に表示させることはできるようになりました。ただし、光っている部分には、1列で7個のLEDが入っています。今度はLEDをもっと細かく使ってみましょう。以下のプログラムを実行しましょう。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA4 > _7segTest2

実行結果：0ケタ目にアルファベットの「H」が表示される。

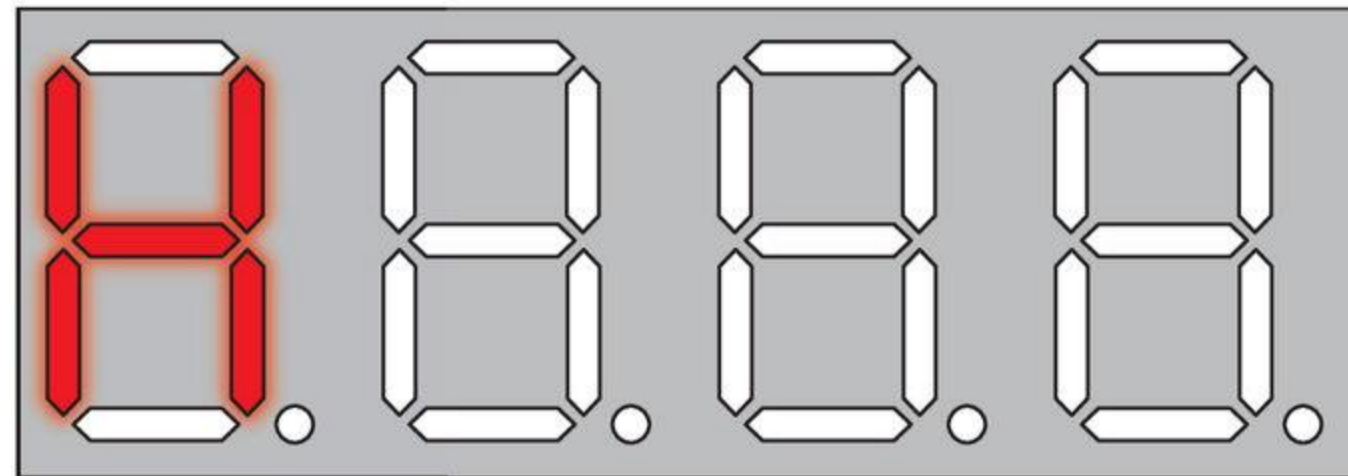


図1-6 7セグメントLEDにアルファベットを表示する

`setDigit` は数字を表示させるときのみ使える命令なので、今回は使うことができません。では、どのように命令して「H」の表示をつくっているのでしょうか。

□ プログラム「_7segTest2」より抜粋 ぼっすい

```
void loop(){
  lc.setLed(0, 0, 2, HIGH);
  lc.setLed(0, 0, 3, HIGH);
  lc.setLed(0, 0, 4, HIGH);
  lc.setLed(0, 0, 5, HIGH);
  lc.setLed(0, 0, 6, HIGH);
}
```

1文字表示するだけで5行も同じ命令をかいていますね。どのような命令でしょうか。

命令 [setLed]

実行内容：7セグメントLEDの指定したケタの指定したセグメントを点灯させる

使い方：`lc.setLed(0, 2, 5, HIGH)`

// lcと名付けたLEDの、2ケタ目の5番セグメントを点灯させる

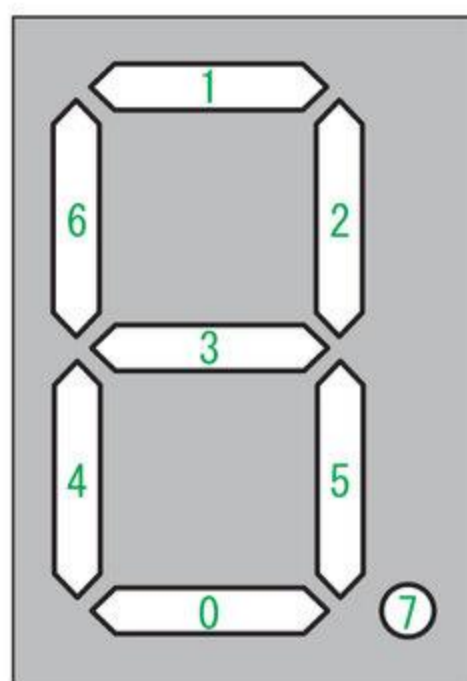
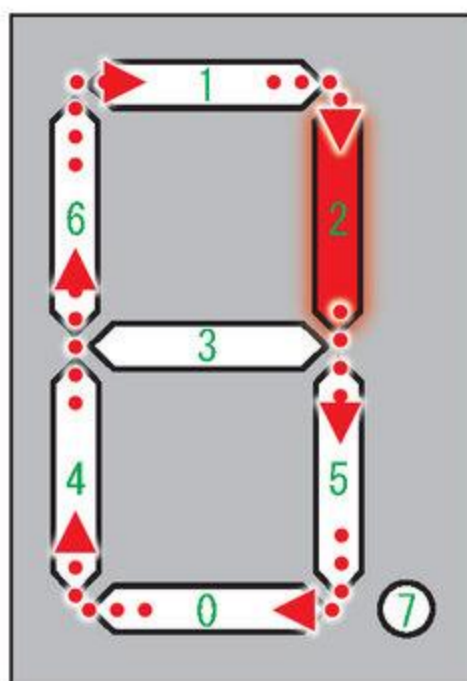


図 1-7 7セグメントLEDのセグメント番号

7セグメントLEDは図 1-7のように、それぞれのセグメントに番号が割り振られています。`setLed` でこの番号を指定すれば、数字以外でも好きな形を表示させることができます。マトリクスLEDでも、光らせる点を1つひとつ指示する命令がありましたが、似たような考え方ですね。

やってみよう！

プログラム「_7segTest2」をかきかえ、0ケタ目のLEDが以下の図のように時計回りにグルグル光るようにしてみよう！



「やってみよう！」のプログラムは完成したでしょうか。

2番セグメントからスタートする場合、2番→5番→0番→4番→6番→1番と光らせていけば一周できます。ただし、次のセグメントを点灯させるのと同時に、前のセグメントを消灯させる必要があります。

よって、以下のようにプログラムすればよいですね。

```
void loop(){
  lc.setLed(0, 0, 2, HIGH);
  delay(100);
  lc.setLed(0, 0, 2, LOW);
  lc.setLed(0, 0, 5, HIGH);
  delay(100);
  lc.setLed(0, 0, 5, LOW);
  lc.setLed(0, 0, 0, HIGH);
  delay(100);
  lc.setLed(0, 0, 0, LOW);
  lc.setLed(0, 0, 4, HIGH);
  delay(100);
  lc.setLed(0, 0, 4, LOW);
  lc.setLed(0, 0, 6, HIGH);
  delay(100);
  lc.setLed(0, 0, 6, LOW);
  lc.setLed(0, 0, 1, HIGH);
  delay(100);
  lc.setLed(0, 0, 1, LOW);
}
```


ここで、プログラムを以下のようにかきかえてみましょう。

```
void loop(){
  int seg[] = {2, 5, 0, 4, 6, 1};

  lc.setLed(0, 0, seg[0], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[0], LOW);
  lc.setLed(0, 0, seg[1], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[1], LOW);
  lc.setLed(0, 0, seg[2], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[2], LOW);
  lc.setLed(0, 0, seg[3], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[3], LOW);
  lc.setLed(0, 0, seg[4], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[4], LOW);
  lc.setLed(0, 0, seg[5], HIGH);
  delay(100);
  lc.setLed(0, 0, seg[5], LOW);
}
```

実行結果：^{へんこう}変更前と同じく、0ケタ目のセグメントが時計回りに光る。

2だったところが seg[0] に置きかわりましたが、動作は変わりませんでしたね。つまり、2 と seg[0] は同じ値であることがわかります。同じように考えていくと、次のような関係が見えてきます。



POINT

```
seg[0] = 2
seg[1] = 5
seg[2] = 0
seg[3] = 4
seg[4] = 6
seg[5] = 1
```

ここまでわかると、このプログラムの意味がつかめてくるのではないのでしょうか。


```
int seg[] = {2, 5, 0, 4, 6, 1};
```

`int seg` とかく変数宣言^{せんげん}であれば、これまでに何度も登場していました。しかし、この方法で変数 `seg` をつくってしまうと、`seg` には1度に1種類の値しか入ることができません。今回のように、変数の中身（セグメント番号）を次々と変更^{へんこう}していきたいときには、あまり向いていないのです。

変数名に `[]` をつけると、変数 `seg` が「配列^{はいれつ}」とよばれるデータ型になります。今回であれば、`seg[0]` ~ `seg[5]` までの6通りを、別々の変数のように扱^{あつか}うことができます。ちなみに、`[]` 内の値を「添え字^そ」とよびます。

「そんなことをして何になるの？」と疑問に思う人もいるかもしれませんが、今回はこの「配列^{はいれつ}」を活用することでプログラムを一気にシンプルにまとめることができるのです！なぜそんなことができるのかは、実際に挑戦しながら考えてみましょう。

やってみよう！

配列^{はいれつ}を用いたプログラムを、さらにシンプルにまとめてみよう！
また、LEDの光る順番を自由に変更^{へんこう}してみよう！他のケタにも移動できるようにしてみると面白いね！

解答例は以下の通りです。

```
void loop(){
  int seg[] = {2, 5, 0, 4, 6, 1};

  for(int i = 0; i < 6; i++){
    lc.setLed(0, 0, seg[i], HIGH);
    delay(100);
    lc.setLed(0, 0, seg[i], LOW);
  }
}
```

講

セグメント番号は 2 → 5 → 0 → … と不規則に変化しますが、これを配列に入れると 0 → 1 → 2 と規則的に変化していく「添え字」で制御できるようになります。つまり、for文が使えるようになるのです。

また、アニメーションの様子を変更^{へんこう}したいときに `int seg[] = {2, 5, 0, 4, 6, 1};` の部分をかきかえるだけで済むという利点もあります。

1.3. 数字をカウントする

今度は、「数字のカウント」を表示してみましょう。
まずは、以下のプログラムを実行しましょう。

プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA4 > Counter

実行結果：7セグメントLEDに表示されている数字が1秒に1ずつカウントアップされる。
ここでは、以下の命令が使われています。

命 令 [setDec]

実行内容：指定した最大4ケタの数字を表示する

使 い 方：lc.setDec(0, 1234); // lcと名付けたLEDに「1234」という数を表示

複数のケタを一気に指定できるので、`setDigit` よりもシンプルです。ちなみに表示できる数は最小-999、最大9999です。

2. センサーを使う (目安 25 分)

2.0. ボリュームセンサーの値を確認する

7セグメントLEDは数値の表示が得意なので、アナログセンサーの値が今いくつなのかを知るのには非常に役立ちます。

ために、^{かへんていこう}可変抵抗ボリュームの値がチェックできるようにしてみましょう。

ステップアップ

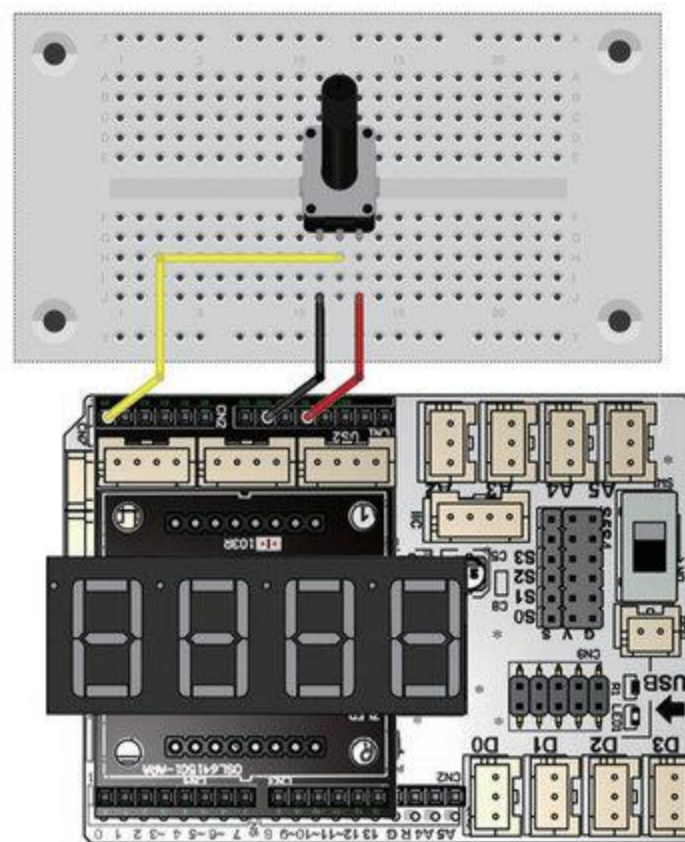
ブレッドボードに^{かいろう}回路をつくり、^{かへんていこう}可変抵抗ボリュームを接続しよう！
完成したら、ボリュームセンサーの回し具合に応じて7セグメントLEDに0～1023の数字が表示されるようなプログラムをつくってみよう！

💡 ヒント

アナログセンサーの値を読みとるプログラムは前回、7セグメントLEDに数値を表示させるプログラムは今回学んでいるね！

第3回・第4回授業の^{しゅうたいせい}集大成のつもりで取り組んでみよう！

回路の製作例は以下です。



講

解答例は以下のプログラムです。

[RoboticsProfessorCourse2 > MagicItemA4 > _7segVol](#)

非常にシンプルなプログラムですが、7セグメントLED用の説明書ファイルの取り込み、7セグメントLEDの初期設定、アナログセンサー値の読み取り、7セグメントLEDでの数値表示と、前回・今回の重要なプログラムが随所に使われています。授業内容の復習・定着を兼ねて、なるべく0からプログラミングさせていく形で、生徒主導で進めさせて下さい。

2.1. ちょうおんぱきより超音波距離センサーを活用する

チャレンジ課題

ロボットとプログラムをつくりかえ、ちょうおんぱきより超音波距離センサーではかった距離が7セグメントLEDに表示されるようにしてみよう！

ヒント

ちょうおんぱきより超音波距離センサーはマトリクスLEDシールドの [US1] に接続しよう！
ちなみに、説明書プログラム [RPlib.h] を取り込むことで、ちょうおんぱきより超音波距離センサーに距離計測をさせる命令 `ussRead(US●)` が使えるようになるよ！

講

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > MagicItemA4 > Ultrasonic

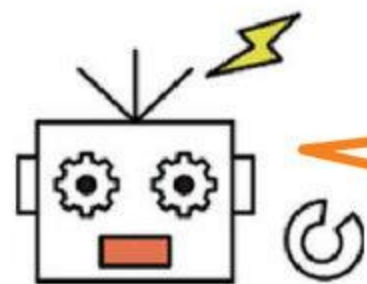
超音波距離センサーの読み取りがうまくいかない場合は電池ボックスからの給電にかえてみるとうまくいくことがあります。

3. まとめ (目安5分)

今回は、新たな電子パーツ「7セグメントLED」を使ってみました。この^{ひょうじき}表示器を使うことによって、マイコンの思考が見えるようになって、プログラムするのが^{あつとうてき}圧倒的にわかりやすくなります！

また、センサーの値を表示することができるので、^{そくてい}測定する機械をつくるのにも役に立ちます。温度、^{しつど}湿度、^{しやうど}高度、^{しょうど}照度、かたむき、におい、音、など世の中のいろいろなセンサーの値を確認できますね。

次回は、さらに応用してみましよう。より深く、ロボットプログラムを勉強していきますよ！



7セグメントLEDをグルグルしすぎて、目がまわった？

講

- 以下の授業の目標を再確認します。
 - ・7セグメントLEDを光らせる
 - ・7セグメントLEDのプログラムをマスターする
 - ・センサーの値を表示する
- 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う。
- 次回テーマは「電子回路とプログラミング (コンビネーション1)」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。

ラジオペンチ 1	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	スピーカー 1	超音波距離センサー 2
			
センサーケーブル 2	301ブレッドボード 1	ジャンパー線 65	7セグメントLED 1
			
抵抗 (100Ω / 1kΩ / 10kΩ) 各10	可変抵抗ボリューム 2	タクトスイッチ 10	
			

図 3-0 次回必要なもの

P.9 やってみよう! ① 解答例

```
void loop(){  
  lc.setDigit(0, 0, 3, false);  
  lc.setDigit(0, 1, 2, false);  
  lc.setDigit(0, 2, 3, false);  
  lc.setDigit(0, 3, 1, false);  
}
```

P.9 やってみよう! ② 解答例

```
void loop(){  
  lc.setDigit(0, 1, 3, true);  
  lc.setDigit(0, 2, 1, false);  
  lc.setDigit(0, 3, 4, false);  
}
```