

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 - 1 ③

(第5回/第6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5回授業日 2024年 月 日

だい かい じゅ ぎょう び
第6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9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I-1③

第5回

ラーメンタイマーをつくる

講師用

目 次

0. ラーメンタイマーをつくる

0.0. 「ラーメンタイマーをつくる」でやること

0.1. 必要なもの

1. タッチセンサーとスピーカーの取り付け

1.0. タッチセンサーの取り付け

1.1. スピーカーの取り付け

2. 文字と図形の表示をふり返ろう

2.0. 文字の表示

2.1. 図形の表示

3. ラーメンタイマーの作成

3.0. ラーメンタイマーの作成プラン

3.1. 時間をはかる

3.2. 時間を設定する

3.3. 時間がきたらアラームを鳴らす

3.4. 「switch-case 文」を理解しよう

3.5. ラーメンタイマーを使ってみよう

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。

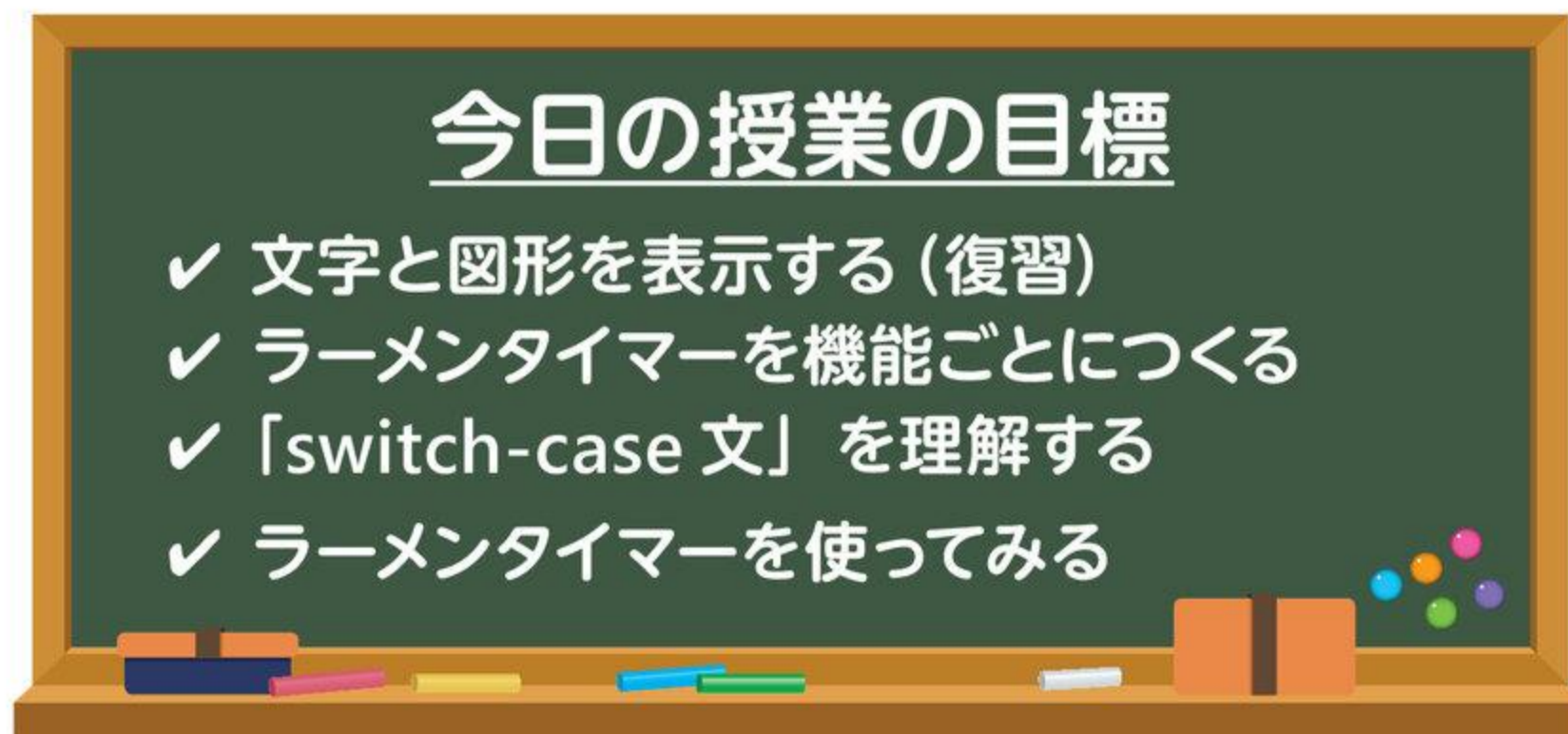
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(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. ラーメンタイマーをつくる (目安 10 分)

0.0. 「ラーメンタイマーをつくる」でやること



今回は、前回に引き続きCGの基礎きその部分の復習をし、そのうえで追加命令を改めて学びましょう。

そして、メインは「オリジナルラーメンタイマー」の作成です。今まで勉強したことを結集してつくりましょう！ マイコンは計算することも得意ですが、時間をはかることも、とても得意です。分、秒はもちろんのこと、秒の1000分の1まではかれます。今までとはやや違い、今回は実用的ですよ！ 家族や友達に見せてあげて、「とても便利なものもつくれるようになったよ！」と自慢じまんしちゃいましょう。売っているタイマーにはないオリジナル要素をたくさん入れて、自分だけのラーメンタイマーをつくりましょう！



0.1. 必要なもの

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。

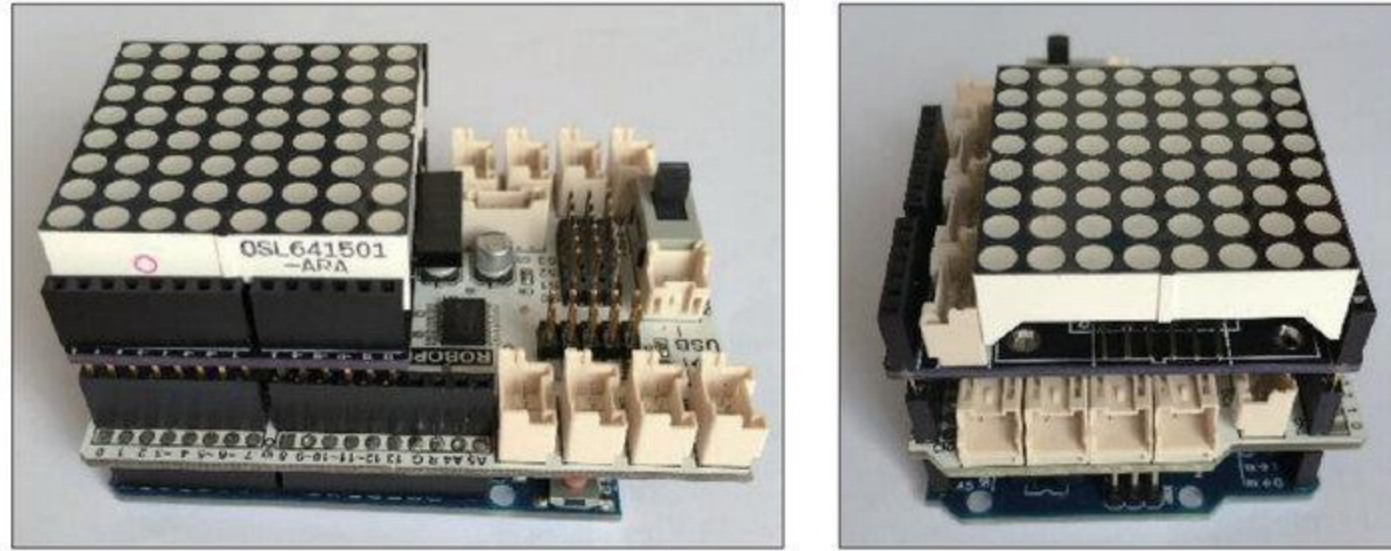


図0-0 前回までに組み立てたもの




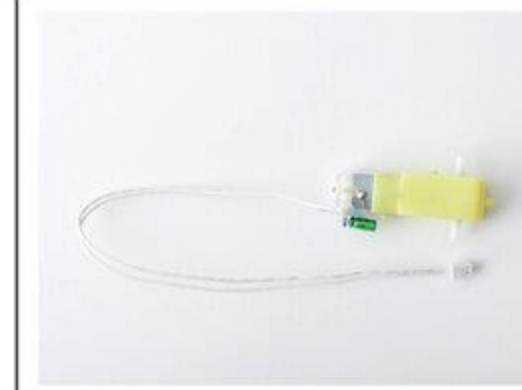




USBケーブル	1	マイコンボード	1	ロボプロシールド	1	ギアドモーター	2
							
タッチセンサー	2	マトリクスLEDシールド	1	マトリクスLED	1	スピーカー	1
							

図0-1 必要なもの

1. タッチセンサーとスピーカーの取り付け (目安 10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] のコネクタにしっかりとさし込みます。

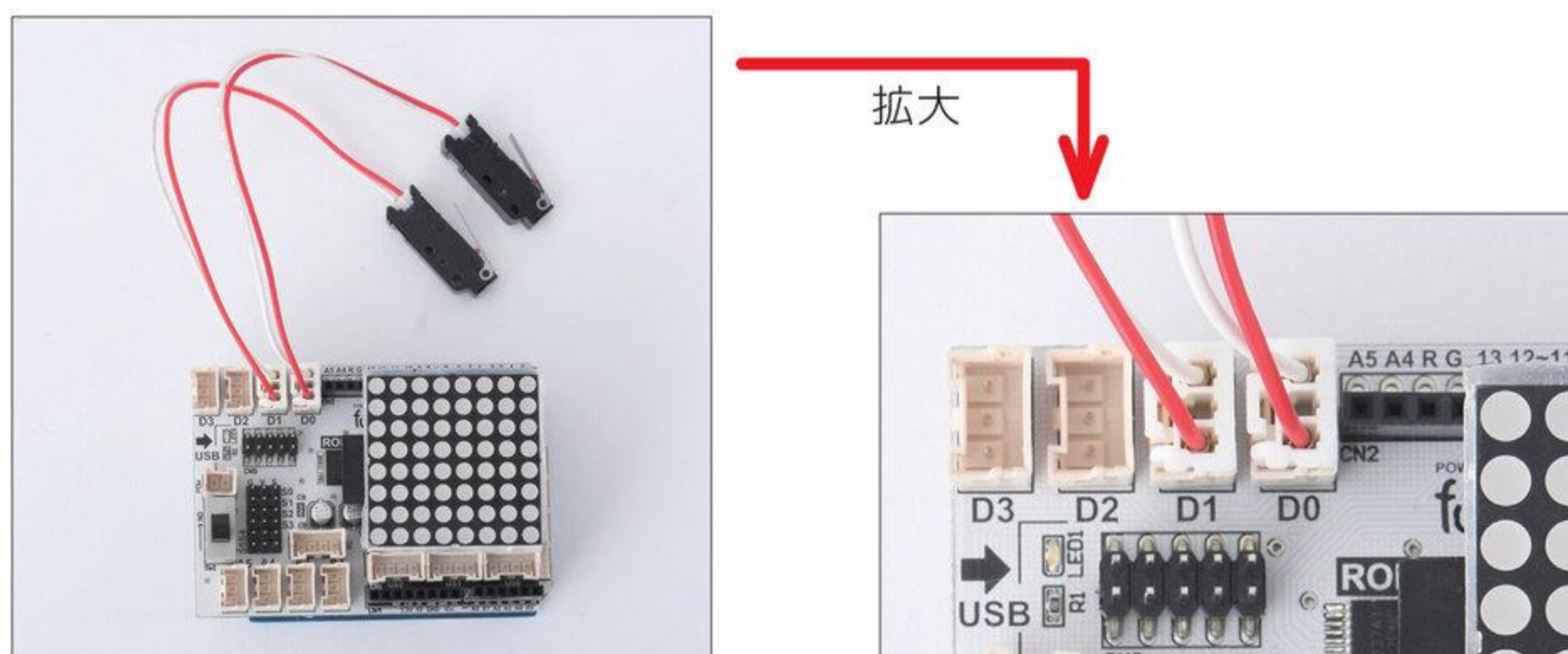


図1-0 ロボプロシールドとタッチセンサーの接続完成図 (全体図と拡大図)

ケーブルをぬく場合は、ケーブルの爪^{つめ}の部分を押さえながら引きぬきましょう。無理に引っ張るのは、ケーブルを痛めることになるので、やめましょう。

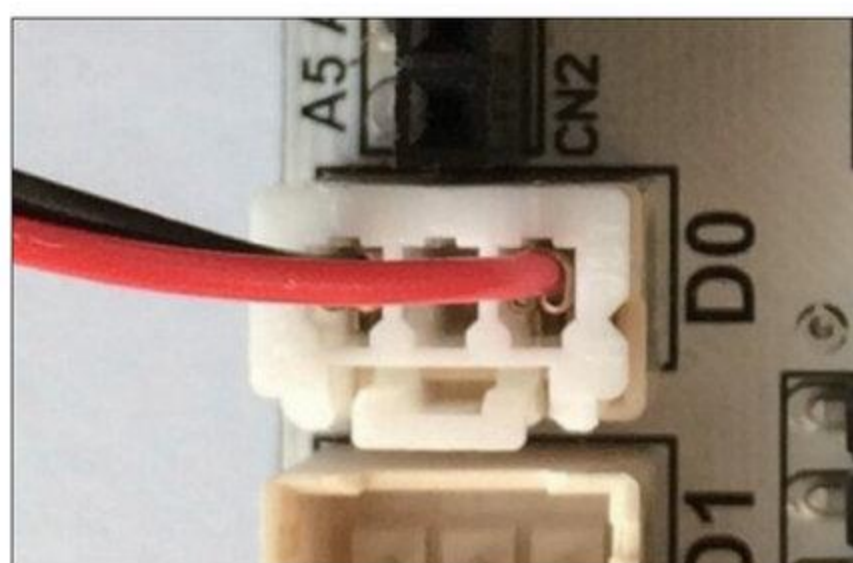


図1-1 ケーブルの爪^{つめ}の部分

1.1. スピーカーの取り付け

スピーカーを図1-2のように [D2] に接続しましょう。

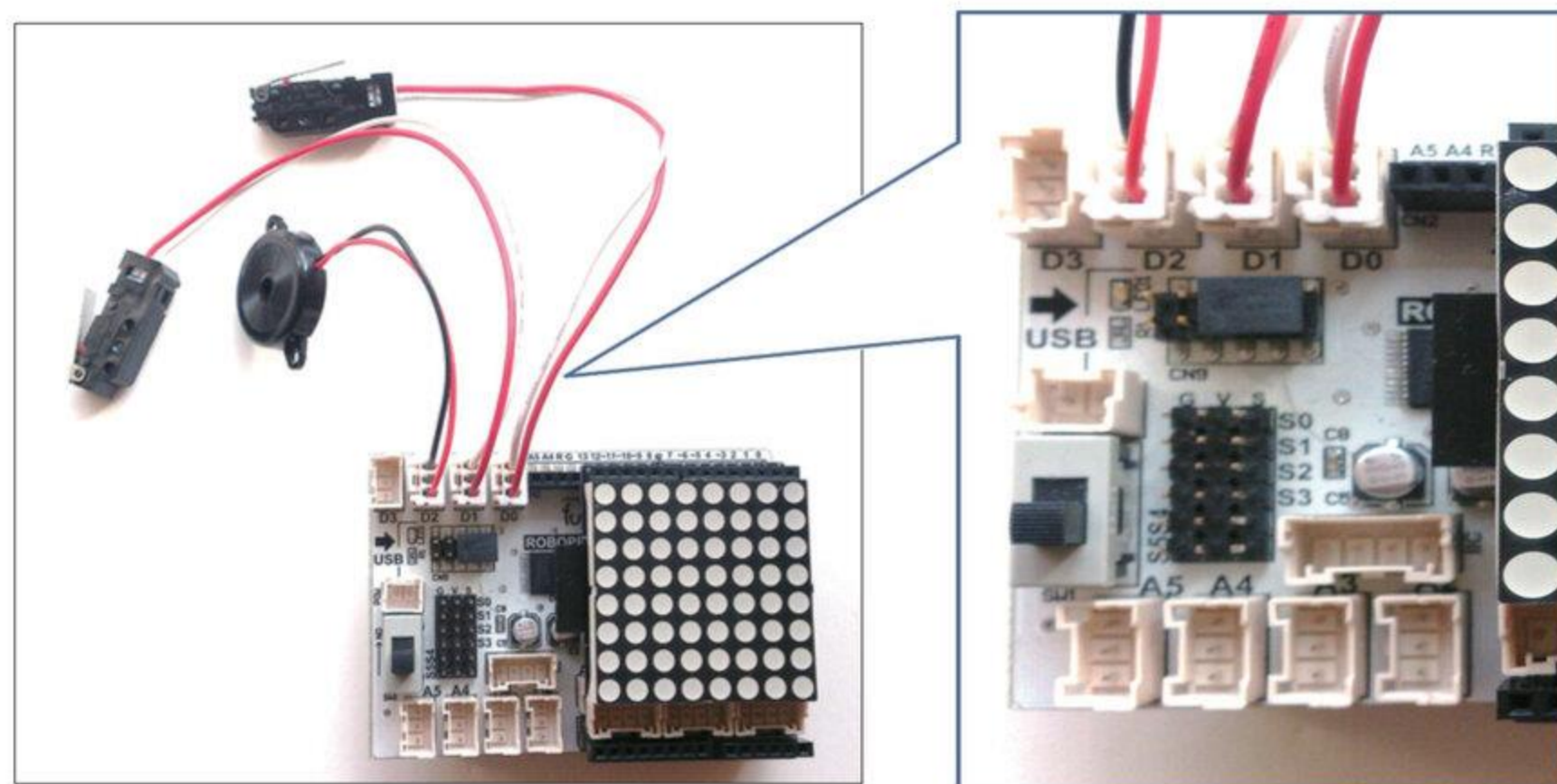


図1-2 スピーカーの取り付け

2. 文字と図形の表示をふり返ろう (目安 30 分)

2.0. 文字の表示

1) 文字コード

前は文字コードを学びましたね。マイコンは欧米で生まれたので日本語の扱いが苦手で、ひらがなを英数字で表していました。それでは、文字コード表の全部の欄を確認してみましょう。

表2-0 文字コード表

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x00																
0x10																
0x20	/	!	"	#	\$	%	&		()	*	+	,	-	.	/
0x30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0x60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	/
0x80	/	あ	い	う	え	お	か	が	き	ぎ	く					
0x90	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
0xa0	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
0xb0	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
0xc0	む	め	も	や	ゃ	ゆ	ゆ	よ	よ	ら	り	る	れ	ろ	わ	わ
0xd0	ゐ	ゑ	を	ん	づ	°	←	→	↑	↓	┌	└	【	】	※	〒
0xe0	☆	★	○	●	◎	◇	◆	□	■	△	▲	▽	▼	℃	≦	≧
0xf0	∞	♂	♀	♪	±	×	÷	≠	～	…	年	月	日	時	分	秒

実は前は、**緑色の部分**は隠しキャラとして表にのせていませんでした。また、グレーの部分には絵のデータが入っていましたね。文字以外にも、さまざまなキャラクターを表示できるのです。では、次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 > MagicItemA5 > textTest1

実行結果：隠しキャラだけ表示される。
続いて、次のプログラムも実行してください。

🌀 プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > textTest2

実行結果：文字コード「0xea」の「▲」がグルグル回る。

2) putch命令とは

では、ここで新たに出てきた命令を学びましょう。

命 令 「putch」

実行結果：指定の位置に、指定の向きの文字を表示する

使 い 方：myMatrix.putch(0, 0, 0x41, 0);

// 一番左上の座標を(0, 0)として、「A」の文字を上向きで表示



POINT

`putch` の命令は、文字を表示する位置（左上の座標）、表示する文字、文字の向きの3つが指定できます。

表示する文字は表2-0の文字コードで書くか、`' '`（シングルクォーテーション）で文字を囲むことで指定できます。

文字の向きは `UP`（上向き）、`DOWN`（下向き）、`RIGHT`（右向き）、`LEFT`（左向き）の4種類が指定できます。なお、0, 1, 2, 3という数字をかわりに書いても構いません（それぞれ `UP`、`DOWN`、`RIGHT`、`LEFT` と同じはたらきをします）。ちなみに、文字の向きを指定せずに命令することもできます。その場合は自動的に `UP`（上向き）になります。

たとえば、下の3つは全て同じ命令になります。

```
myMatrix.putch(0, 0, 0x41, 0);  
myMatrix.putch(0, 0, 'A', 0);  
myMatrix.putch(0, 0, 'A', UP);
```

やってみよう!

プログラム「textTest2」を書きかえ、以下のようにしてみよう。

- ① 表示するものを▲から★にする。
- ② 自分のイニシャル（アルファベットの頭文字）を表示する。

チャレンジ課題

プログラム「textTest2」を^{へんこう}変更して、「★」が転がりながら、左から右に1マスずつ移動するようにしよう。

 ヒント

表示する図形、図形の位置、図形の向きと3つの指定が必要だね。位置は左端から右端まで移動するのに8マスもあるから、for文を使うとラクに書けるよ！

講

解答プログラムは以下となります。

RoboticsProfessorCourse1 > MagicItemA5 > textTest3

また、for文を使った上級者用の解答例も用意しました。以下となります。

RoboticsProfessorCourse1 > MagicItemA5 > textTest4

解答例は巻末に記載します。

2.1. 図形の表示

今回は、図形の表示の復習です。今回は、直線しかかけなかったのですが、直線を組み合わせずに円と四角をかいてくれる命令があります！

まずは、次の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 > MagicItemA5 > MatrixCircle1

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > MatrixRectangle1

実行結果：「MatrixCircle1」は円の図を、「MatrixRectangle1」は四角の図を表示する。では、ここで新たに出てきた命令を学習しましょう。

命 令 「myMatrix.circle」

実行結果：指定の位置に、指定の半径の円を表示する

使 い 方：myMatrix.circle(2, 4, 3);

// 座標(2, 4)を中心に、半径3マス分の円を表示

命 令 「myMatrix.rectangle」

実行結果：指定の位置に、指定の大きさの長方形を表示する

使 い 方：myMatrix.rectangle(0, 0, 6, 4);

// 座標(0, 0)から、横6マス縦4マス伸ばしてできる長方形を表示

やってみよう!

- 1.プログラム「MatrixCircle1」の中を以下のように変更してみよう。
 「.」(ピリオド)と「,」(カンマ)のちがいに注意してね。

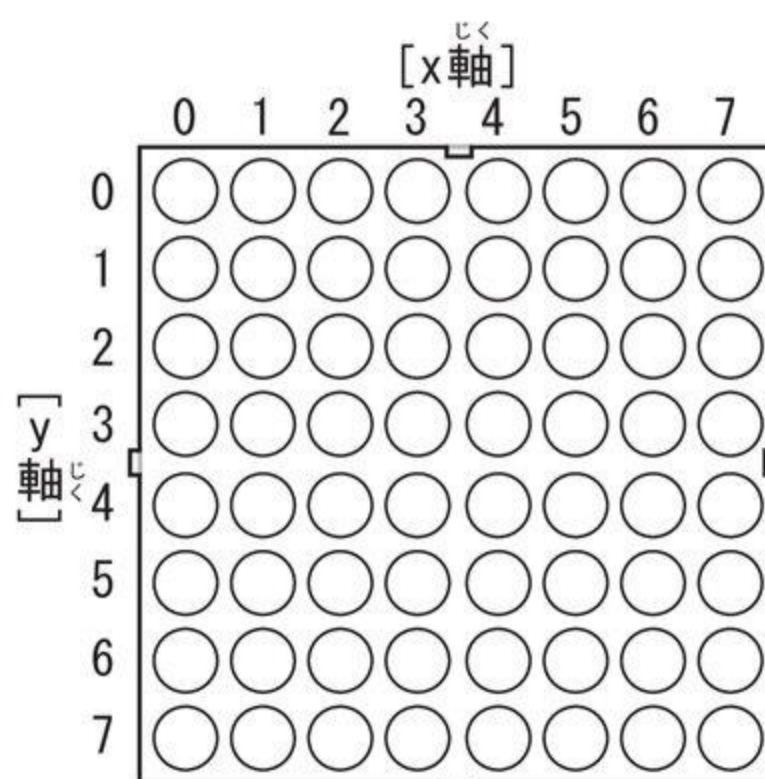
□ プログラム「MatrixCircle1」より**抜粋**

```
myMatrix.circle(4, 4, 3.5);
```

```
myMatrix.circle(3, 3, 2.5);
```

どのような表示になるか予想してみよう!

できたら、実際にプログラムを書きかえて確認してみよう!



- 2.プログラム「MatrixRectangle1」の中を以下のように変更してみよう。

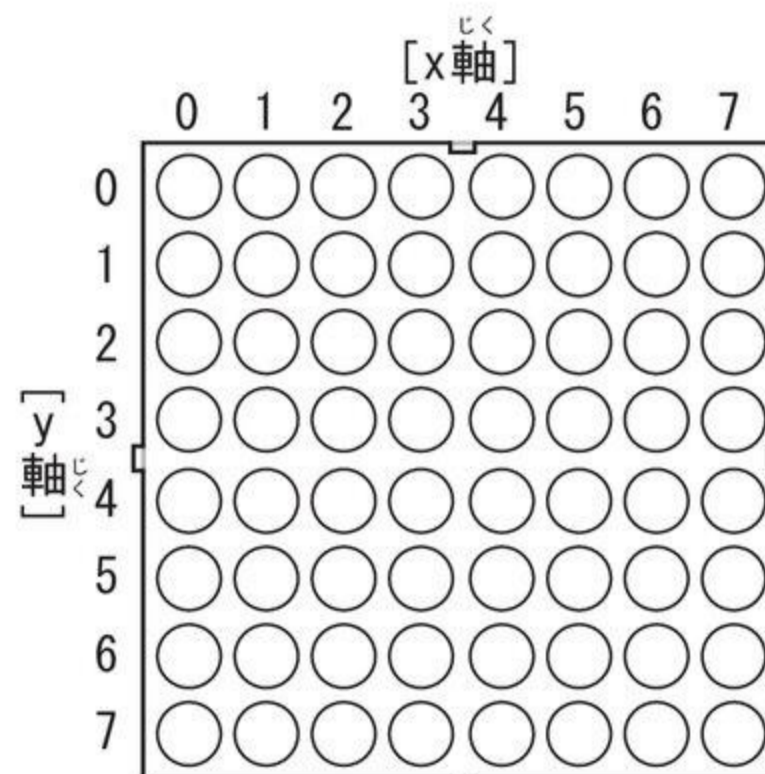
□ プログラム「MatrixRectangle1」より**抜粋**

```
myMatrix.rectangle(0, 0, 3, 3);
```

```
myMatrix.rectangle(0, 0, 7, 7);
```


どのような表示になるか予想してみよう!

できたら、実際にプログラムを書きかえて確認してみよう!



チャレンジ課題

プログラム「MatrixCircle1」と「MatrixRectangle1」を^{へんこう}変更して、それぞれ円と四角が大きくなったり小さくなったりするアニメーションをつくってみよう。

 ヒント

半径や一辺の長さを変数として、for文でくり返そう。

講

解答プログラムはそれぞれ以下となります。

RoboticsProfessorCourse1 > MagicItemA5 > MatrixCircle2

RoboticsProfessorCourse1 > MagicItemA5 > MatrixRectangle2

解答例は巻末に記載します。

3. ラーメンタイマーの作成 (目安 50 分)

3.0. ラーメンタイマーの作成プラン

それでは、これまでに学んできた命令をうまく組み合わせて「ラーメンタイマー」をつくってみましょう。

「ラーメンタイマー」は文字通り、カップ^{めん}麺にお湯を注いだあと、ちょうどよい時間を教えてくれるタイマーです。



図3-0 ラーメンタイマー

「カップ^{めん}麺」と一言で言っても、お湯を注いでから3分でできあがるものや5分かかかるものなど、はかりたい時間はさまざまです。よって、まずは「カウントする時間を設定する」という機能が必要でしょう。

では、他にはどのような機能が必要でしょうか？

やってみよう!

「カウントする時間を設定する」以外にどのような機能があれば「ラーメンタイマー」が完成するかな？

ぜひで3つの機能に分けて考えるとして、残り2つの機能を考えてみよう!

図3-0もヒントにしてみてくださいね。

 時間が経つのにあわせてカウントを進める、残り時間が0になったら知らせる。

うまくまとめられましたか？

例として、今回は以下のような3機能に分けてプログラムをつくっていきます。



POINT

ラーメンタイマーを実現するのに必要な機能は…

- ① カウントする時間を設定する。
- ② 時間が経つのにあわせてカウントを進める。
- ③ 残り時間が0になったら知らせる。

3.1. 時間をはかる

まずは、簡単なところからいきましょう！ 次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 > MagicItemA5 > CountBom

実行結果：LEDに表示された10の数字がカウントダウンし、0になると音が鳴る。

このプログラムはfor文を使って時間をはかっています。for文の使い方はわかってきましたか？ では、このプログラムを改造しながら時間のはかり方を学びましょう。

ステップアップ

プログラム「CountBom」を書きかえ、以下のようなようにしてみよう！

1. forの `()` 内を書きかえ、カウントできる時間を20秒にする。
2. forの `()` 内は変更せずに、カウントできる時間を20秒にする。

1. の解答例は以下の通りです。

```
for(i = 20; i > 0; i--)  
スタートの数字を10から20に変更しています。
```

2. の解答例は以下の通りです。

```
void loop(void){  
    for(i = 10; i > 0; i--){  
        myMatrix.putd2(0, 0, i); // マトリクス LED に 2 けたの数字を表示  
        tone1.play(2252, 50);  
        delay(2000);  
        myMatrix.putd2(0, 0, --i); // マトリクス LED に 2 けたの数字を表示  
        tone1.play(2000, 50);  
        delay(2000);  
    }  
}
```

delayの時間を1秒から2秒にのばしました。これにより、10カウントで20秒が経過するようになります。

3.2. 時間を設定する


1) スイッチを使って設定する

時間をはかるには、`delay();` 命令がポイントになりますね！しかし、設定時間を変更^{へんこう}するために、プログラムをいちいち変更^{へんこう}するのは面倒^{めんどろ}ですよね。ラーメンをつくるたびに、パソコンを持ってきていたら大変です。そこで、はかる時間もタッチセンサーを使って設定できるようにしましょう！

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > CountTimer1

タッチセンサーをさわって動作確認をして、結果を以下に記入しましょう。

[SW1] を押すと  押すたびに1つずつ数字が上がる。

[SW2] を押すと  数字が1つずつ下がり、0になると音楽が鳴る。

時間を設定してスイッチを押すと、カウントダウンが始まり、0になったらアラームが鳴る、というプログラムを確認できましたか？

2) while命令とは

このプログラムには見慣れない命令があります。学習しましょう。

命 令 [while]

実行結果：指定通りの状況である間は、`{ }` 内の処理をくり返し続ける

使 い 方：`while(i < 10){ // 変数iが10未満である間は、{ }の処理をくり返す`

if文と書き方が似ていますが、ifは条件を満たしていてもそうでなくても判定後は次の処理に進むのに対し、whileは条件を満たしている間は次の処理に進まない、という特徴^{とくちょう}があります。「ボタンを押したら動きだすロボット」など、キッカケの前後で動作を分けたいときなどに便利です。

やってみよう!

プログラム「CountTimer1」の以下の黄色の数字を「5」に変更^{へんこう}してみよう。そして、[SW1]を押してLEDの表示を「10」にしたあと、[SW2]を押してみよう。

 プログラム「CountTimer1」より^{ぼっすい}抜粋

```
while(n > 0) //nがカウントダウンされ0以下になるとくり返しをやめる
```

起きたことを記入しよう。

 数字が1つずつ下がり、5になるとカウントダウンが止まって音楽が鳴る。

ステップアップ

プログラム「CountTimer1」を書きかえ、[SW2]を押したら数字が2ずつ上がって行って、50になったら曲が流れるようにしてみよう!

 ヒント

ループのたびに「nの値を2ふやす」という処理が必要だね!

講

解答例は以下の通りです。(「// 時間をカウント」部)

```
while(n < 50){  
    n = n + 2;  
    myMatrix.putd2(0, 0, n);  
    delay(100);  
}
```

3.3. 時間がきたらアラームを鳴らす

ここまででだいぶ、ラーメンタイマーらしくなってきました。いよいよアラームですが、人と同じではつまらないですよね？ アラームの種類を選んで、自分だけのラーメンタイマーにしましょう！好きな時間に好きな曲を演奏させましょう。

まずは順番を追って、曲を選ぶようにするところからはじめます。次の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 > MagicItemA5 > Jukebox1

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > Jukebox2

タッチセンサーを押すと何が起きるか書いておきましょう(どちらのプログラムも動作は同じです)。

[SW1] を押すと  表示される数字が変化する。

[SW2] を押すと  0・1・2・3のそれぞれで違う音楽が鳴る。

この2つのプログラム「Jukebox1/2」は見た目の動作に差はありません。同じことを行う、別の方法を使ったプログラムです。プログラムを書いていると、このようなことが多く発生しますが、プログラムはそもそも、シンプルで、だれが見てもわかりやすいものが一番良い、とされています。

どちらのプログラムが、ぱっと見てわかりやすいでしょうか？ 一般的にはプログラム「Jukebox2」の書き方のほうが良いとされています。理由は、今から紹介する新しい命令にあります。

3.4. 「switch-case 文」を理解しよう

ここで紹介する命令は「switch-case 文」とよばれるものです。分岐が多いプログラムでぜひ使用してもらいたい命令です。

実際のプログラムを見ながら解説していきます。以下は、プログラム「Jukebox2」の抜粋です。

□ プログラム「Jukebox2」より抜粋

```

switch(n){
  case 0: // nが0の場合
    tone1.play_rtttl(song0); // song0という変数に入った曲を流す
    break;
  case 1: // nが1の場合
    tone1.play_rtttl(song1); // song1という変数に入った曲を流す
    break;
  case 2: // nが2の場合
    tone1.play_rtttl(song2); // song2という変数に入った曲を流す
    break;
  case 3: // nが3の場合
    tone1.play_rtttl(song3); // song3という変数に入った曲を流す
    break;
}
    
```

breakが実行されるとくり返しのループから脱出する

このように、「switch-case文」はたくさんの分岐をよりわかりやすい形で記述することができます。複雑な動きをするロボットには必要になってきますので、覚えておきましょう！

命 令 「switch case」

実行結果：変数の値に応じた行の処理だけを実行する

使 用 方：switch(i){

case 0: // 変数iの値が0のときは、この処理を実行する
(中略)

case 1: // 変数iの値が1のときは、この処理を実行する

やってみよう!

プログラム「Jukebox2」の「switch-case文」内にある `break;` の命令をすべて消してみよう。何が起きるかな? 起きたことを記入しよう。

 song0 から song3 まですべての曲が流れる。

`break;` という命令がないと大変なことになることがわかったかと思います。必ず入れるようにしてください。さて、順番にプログラムをパワーアップさせていきますよ! 次のプログラムを実行してください。操作の仕方は「Jukebox1/2」と同じでタッチセンサーを使います。

 プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > Jukebox3

プログラムを実行すると、何が起きましたか?

 各 song の表示が流れて、その後音楽が鳴る。

では、ここでギアドモーターをロボプロシールドの `MC1` / `MC2` に接続しましょう。

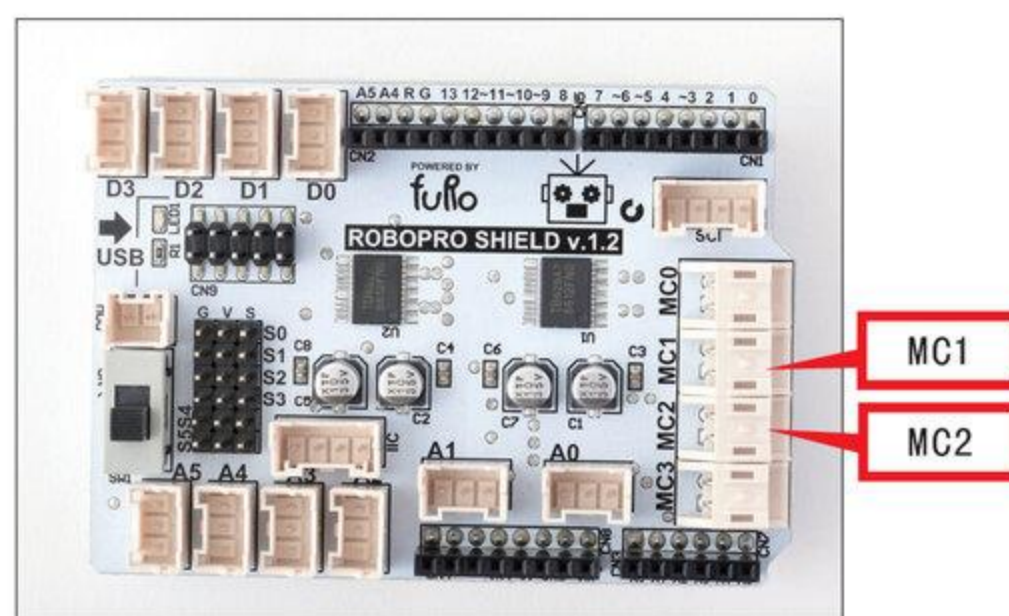


図3-1 ロボプロシールドとモーターケーブルの接続

やってみよう!

プログラム「Jukebox3」を書きかえ、`song0` を選んだときだけ `MC1` と `MC2` のモーターが回転し続けるようにしてみよう!

講

解答例プログラムは以下となります。

RobotixsProfessorCourse1 > MagicItemA5 > Jukebox4



コラム 時間の決め方

「1秒とはどれくらいの時間か」と問われるとなかなか説明しづらいものですが、実は国際的にきちんとした取りきめがあります。1967年に開かれた「第13回国際度量衡総会」というさまざまな単位について話し合う会議で、「秒は、セシウム133の原子の基底状態の二つの超微細構造準位の間の遷移に対応する放射の周期の9,192,631,770倍の継続時間である」と決められたのです。しかし、数字もきりが悪いですし、言葉の意味もさっぱりわかりません。なぜ、このような単位が生まれたのでしょうか。

●十二進法と時間の単位

1年は12か月、1日は12×2=24時間、1時間は12×5=60分です。「空にうかぶ月は、1年間でおよそ12回満ち欠けをするから」「日時計をつくる時、円を10等分するより12等分する方が簡単だったから」「12は2等分、3等分、4等分、6等分とさまざまな分け方ができて便利だから」などいろいろな説がありますが、とにかく時間にまつわる単位には、「12」という数字に関係するものが多いです。ここから、「分」をさらに12×5=60分割した単位もつくられました。「秒」の誕生です。

つまり「秒」は元々「1日」という時間を24×60×60=86,400分割したものでした。

●時間の定義の変化

しかし、科学や技術が発達していくことで、とんでもない事実が明らかになりました。なんと、1日の長さは日によって異なるというのです。1日あたり1～2ミリ秒というとても小さなずれではありますが、これで「秒とは、1日を86,400分割してできた時間である」と定義することはできなくなってしまいました。

世界中で同じ日を基準にする、などの対策もとられましたが、しだいに「1日の長さという一定でないものを基準にするより、どんな時も一定なものを基準にしよう」という意見が大きくなります。そこで、1億年で1秒しかずれないとされている「セシウム原子時計」というものを基準に、改めて秒という単位を決めることになりました。これを「国際原子時」とよびます。それまでの「1秒」と国際原子時での「1秒」がなるべく同じ長さになるようにした結果、上に書いたような単位ができあがったというわけです。

●メートル

みなさんもよく知っている「メートル」も、秒と同様にその定義が変化してきました。もともとは「地球1周分の距離を4,000万分割したもの」を1メートルとしていましたが、地球が完全な球ではなく、1周するルートによって距離が異なるという事実がわかったのです。そこで1889年に「これが1メートル」と決められた目盛りが記された金属棒（メートル原器）を何本もつくり、各国で保管するという取りきめがなされました。

しかしこのメートル原器も、精密測定をすると国によって長さがほんの少しずつ異なることがわかりました。何回かの見直しを経たあと、1983年からは、メートルもつねに一定で誤差のない「光の速さ」を基準に定義されています。

ちなみに、「キログラム」にもその基準である「キログラム原器」がありましたが、メートル原器と似たような経緯で廃止されました。

しかし、キログラム原器が廃止されたのはなんと2019年5月です。メートルと比べると、キログラムはつい最近まで「原器」の時代だったと言えますね！

3.5. ラーメンタイマーを使ってみよう

「switch-case文」命令を使ってプログラム「CountTimer1」を改造したものが、次のプログラムになります。

🌀 プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA5 > CountTimer2

実行結果：[SW1] を押すとLEDの表示の数字が増えていき、[SW2] を押すとカウントダウンが始まりゼロになると音が鳴る。

これで、ラーメンタイマーとして必要な機能は完成しました。あとは、学習の時間が許すだけ好きなようにアレンジを加えてみてもよいですね。

チャレンジ課題

プログラム「CountTimer2」をよく見ると、[callsong] という変数を使って流す曲を選ぶようになっているね。でも、[callsong] の値が0から変化しないので、[song0] しか流すことができなかつたんだ。「CountTimer2」を書きかえて、はじめに流す曲を設定できるようにしてみよう！ 時間設定の部分を参考にしてみてね。

💡 ヒント

[callsong] の値は3をこえられないよね。もし [callsong] が4になりそうなら0に、5になりそうなら1に、6になりそうなら2に、…と変換する処理が必要だ。こういうときは、「[callsong = callsong % 4]」という式を使うといいよ！

「[callsong] を4でわった時のあまりを [callsong] として使う」という意味の文だ。4でわったらあまりが4以上になることは絶対がないから、今回のような場合に役立つよね！

チャレンジ課題

プログラム「CountTimer2」では9.9秒までしかはかれないよね。99秒まではかれるように^{へんこう}変更してみよう。また、1秒に1回動くような「秒針」をつくれなかな？

講

解答例プログラムはそれぞれ以下となります。

RoboticsProfessorCourse1 > MagicItemA5 > CountTimer3

RoboticsProfessorCourse1 > MagicItemA5 > CountTimer4

解答例は巻末に記載します。



コラム マイコンの秘密：クロック

●クロックとは

みなさんの目の前にあるマイコンボードのマイコンだけでなく、コンピューターは基本的に「クロック」とよばれる、一定の周波数で発生する電気信号を利用しています。簡単に言えば、回路に一定のペースで電流を流し、マイコンやコンピューターはそれに合わせて計算をしたり、今回のプログラムで行っていたような時間計測をしたりしているのです。手拍子に合わせて歌を歌うようなものですね。当然、手拍子のテンポがはやいほうが、短時間でたくさんの歌を歌えます。また、手拍子のリズムが狂わないほうが、歌を正確に歌えます。コンピューターも同じように、このクロック信号が素早く、正確に発信されるほうが処理能力が高くなります。

●発振回路の種類

クロック信号を何を利用して作り出すかで、クロック作成回路（発振回路）の種類や名前、性質などがちがいます。主に次のようなものがあります。

① RC発振

「抵抗」と「コンデンサ」という2つの部品を利用します。これらの部品の精度が高くない、また温度変化の影響が大きいいため、あまり正確なクロック信号はつくれません。しかし部品が少なくて済み、安価で壊れにくいという長所があります。そこまで素早く正確なクロックが必要ない場面（たとえばゲーム機など）に使われています。

② セラミック発振

「セラミック発振子」という部品を利用します。周波数の基準とのずれが0.5%ほどと少なく、RC発振より正確なクロック信号が必要な場面で使われます。ただし電圧の変化に弱く、不安定な電圧では発振が止まってしまうこともあるため、安定した電気の供給が必要です。

③ クリスタル発振

「水晶振動子」という部品を利用します。この3種類の中ではもっとも正確なクロック信号をつくることができ、周波数の基準とのずれは0.001%程度と非常に高精度です。そのかわり値段が高いというデメリットがありますが、正確なクロック信号が必要な時計、通信装置などに使われます。

●クロック信号の周波数

「周波数」という言葉はすでに学習しました。音などの波が1秒間に何回振動するか、の値でしたね。クロック信号にも周波数があります。20Hz（ヘルツ）のクロック信号は、1秒間に20回電気の波があるということです。

みなさんの目の前にあるマイコンボードは16MHz（メガヘルツ）、つまり1,600万Hzのクロック回路をもっています。1秒間に1,600万回も電流をオン/オフさせているのです。これを正確にコントロールしているのですから、発振回路というのは大したものですね！

ちなみに、いまの家庭用コンピューターの演算装置（CPU）にはクロック周波数が3GHz（ギガヘルツ）、つまり30億Hzをこえるようなものもありますし、これを数千個、数万個もいっせいに使う「スーパーコンピューター（スパコン）」という大規模な研究用コンピューターも存在します。コンピューターの性能はどんどん上がっています。みなさんの中にも、将来このようなコンピューターの発展にたずさわりたい人がいるかもしれませんね。

4. まとめ (目安5分)

さて、理解できたでしょうか？今回は、たくさんの命令という「部品」をつなぎ合わせ、さらに大きな「部品」にして、ロボットの脳みそをつくる方法を学びました。プログラムもかなり長くなってきましたが、順番を追って部分を切り取ってそれぞれバラバラに見れば、どれも理解できたはずですよ。

プログラムを書くときのコツは、最初から長いプログラムを書かずに、細かく機能別につくって、後から合体させることです。ロボットのハードウェア、部品の組み立てでも同じですよ。ロボットのギアボックスを組み立てたり、タイヤを組み立てたりすることと、プログラムの部品をつくっていくことはまったく同じです。おそれる必要はありません。プログラムもちょっとずつつくっていきましょう！

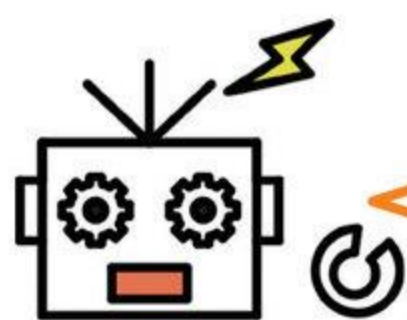
今後も、部品の一つである「命令」とその組合せ方を勉強していきます。自分で部品を自由に組み立てられるようになれば、ロボティクスプロフェッサーに一気に近づけますよ！

《次回必要なもの》

今回は、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	リボンケーブル 1
			
コントローラー 1	無線受信モジュール 1	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



プログラムをマスターして、僕をもっとカシコクしてヨー！

講

- 以下の理解度を確認します。
 - ・文字と図形を表示する(復習)
 - ・ラーメンタイマーを機能ごとにつくる
 - ・「switch-case 文」を理解する
 - ・ラーメンタイマーを使ってみる

- 次回のテーマは「シンセサイザー演奏と人工生命」であることを告知します。

P.7 チャレンジ課題 解答例 (textTest3)

```
Matrix myMatrix = Matrix(11, 13, 1); // マトリクスLEDを使うときのオマジナイ

void setup(){
  myMatrix.clear(); // マトリクスLEDの表示を消す
  myMatrix.setBrightness(3); // マトリクスLEDの明るさを設定(0-8)
}

void loop(){
  myMatrix.putch(0, 0, 0xe1, UP); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(1, 0, 0xe1, RIGHT); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(2, 0, 0xe1, DOWN); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(3, 0, 0xe1, LEFT); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(4, 0, 0xe1, UP); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(5, 0, 0xe1, RIGHT); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(6, 0, 0xe1, DOWN); // 文字コード0xe1の文字を表示
  delay(500);
  myMatrix.clear();
  myMatrix.putch(7, 0, 0xe1, LEFT); // 文字コード0xe1の文字を表示
  delay(500);
}
```


P.10 チャレンジ課題 解答例 (MatrixCircle2)

```
Matrix myMatrix = Matrix(11, 13, 1);

void setup(){
  myMatrix.setBrightness(3);
  myMatrix.clear();           // マトリクスLEDの表示を消す
}

float r;
float x = 4;                 // 変数floatをxという名前で使う(はじめは4)
float y = 4;                 // 変数floatをyという名前で使う(はじめは4)

void loop(){
  // 大きくする
  for (r = 0; r <= 5; r += 0.5){ // 半径を指定した範囲内で0.5ずつ変える
    myMatrix.circle(x, y, r);
    delay(100);
    myMatrix.clear();
  }
  // 小さくする
  for (r = 5; r > 0; r -= 0.5){ // 半径を指定した範囲内で0.5ずつ変える
    myMatrix.circle(x, y, r);
    delay(100);
    myMatrix.clear();
  }
}
```

P10 チャレンジ課題 解答例 (MatrixRectangle2)

```
Matrix myMatrix = Matrix(11, 13, 1);

void setup(){
  myMatrix.setBrightness(3);
  myMatrix.clear();           // マトリクスLEDの表示を消す
}

int x = 0;                   // 変数intをxという名前で使う(はじめは0)
int y = 0;                   // 変数intをyという名前で使う(はじめは0)
int r;

void loop(){
  for(r = 0; r < 8; r++){    // 辺の長さを指定した範囲内で変える
    myMatrix.rectangle(x,y,r,r);
    delay(100);
    myMatrix.clear();
  }

  for(r = 7; r > 0; r--){   // 辺の長さを指定した範囲内で変える
    myMatrix.rectangle(x,y,r,r);
    delay(100);
    myMatrix.clear();
  }
}
```


P.19 チャレンジ課題 解答例 (CountTimer4)

```
// 時間をカウント
while(n > 0){
    myMatrix.putd2(0,0,--n);
    // 秒針を表示
    for(int ss = 0; ss < 10; ss++){
        myMatrix.write(7 - ss,7,HIGH);
        delay(100);
        myMatrix.write(7 - ss,7,LOW);
    }
}
```