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II③

第6回

電子回路とプログラミング（コンビネーション2）

講師用

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6月授業分

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。

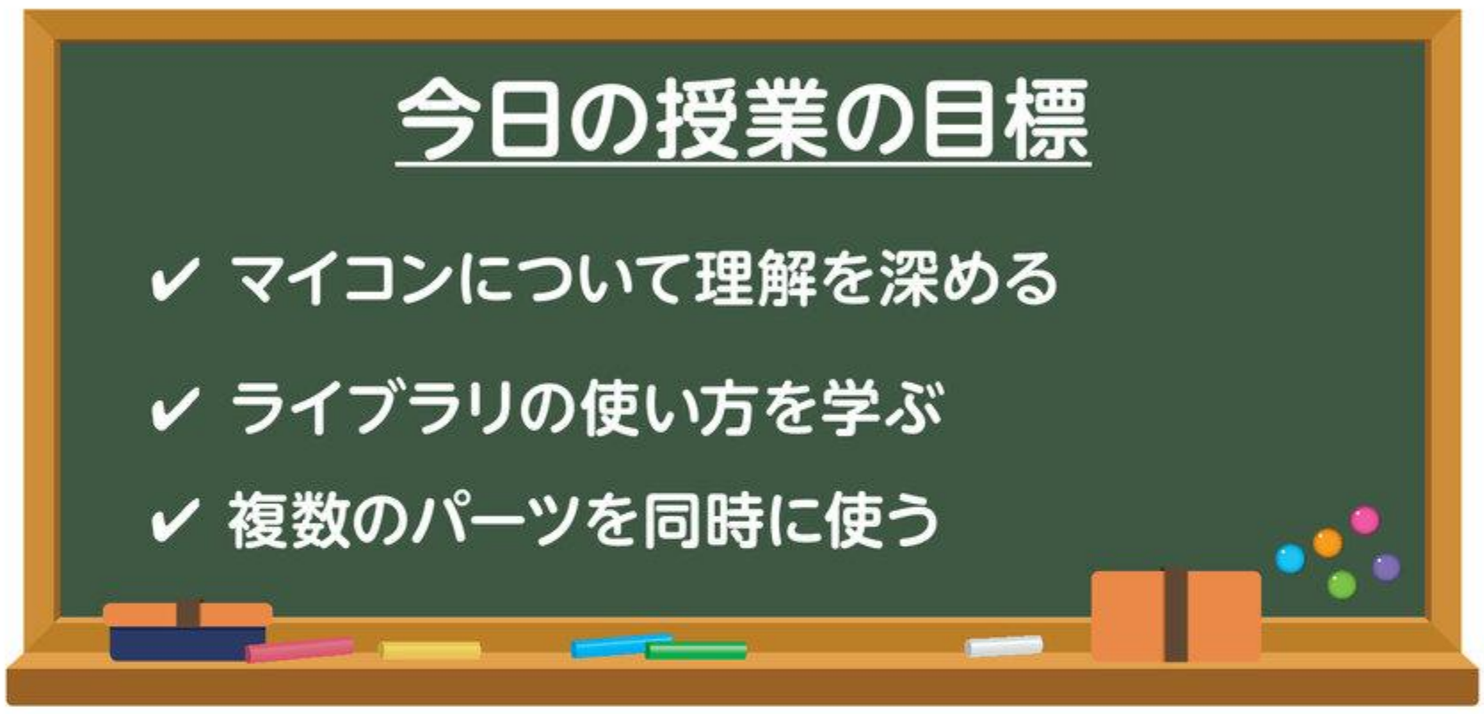
○　今回の目標をパネルで用意するか、黒板に予め書いておきます。

（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．電子回路（でんしかいろ）とプログラミング（コンビネーション２）（目安５分）

0.0. 「電子回路（でんしかいろ）とプログラミング（コンビネーション２）」でやること



今までの授業では、さまざまなセンサー、モーター、スピーカー、LED などのパーツを使ってきました。今回はパーツを同時に使う場合を考えてみます。
さて、多くのパーツを同時に使うにはどうしたらいいのか？　全てのパーツを接続（せつぞく）すればいいのか？　そして、プログラムもどのように書けばいいのか？　さまざまな疑問（ぎもん）が生まれるかと思います。ここでは、その疑問（ぎもん）をズバリ解決（かいけつ）していきましょう。
今回の授業では、大きく二つのことを学びます。一つはマイコンの機能について。二つ目は「ライブラリ」について。
「ライブラリ」とは、よく使うプログラムをひとまとめにしたものです。
サンプルプログラムの中で「〇〇を使うときのオマジナイ」などと書かれている部分です。
このライブラリを理解すると、センサーやモーターなど、授業で使用するパーツを自由自在に使いこなすことができるのです！

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、超音波距離センサーを使う場合は、電池ボックスを使用するようにしましょう。

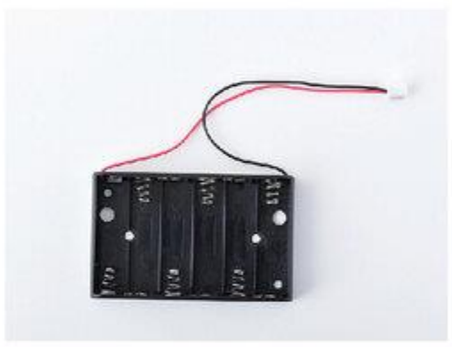
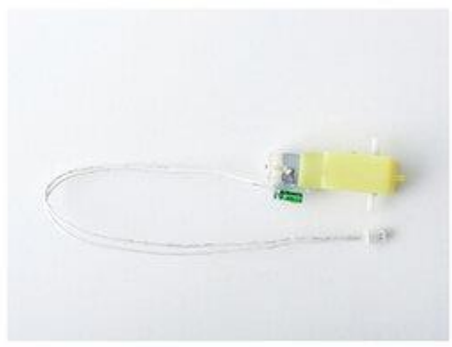
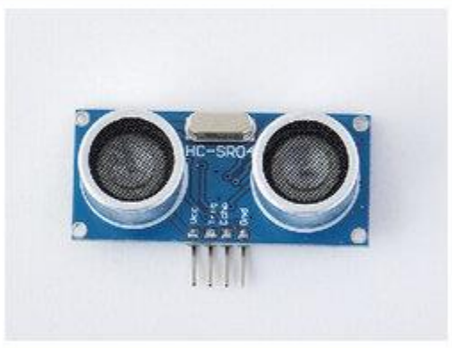
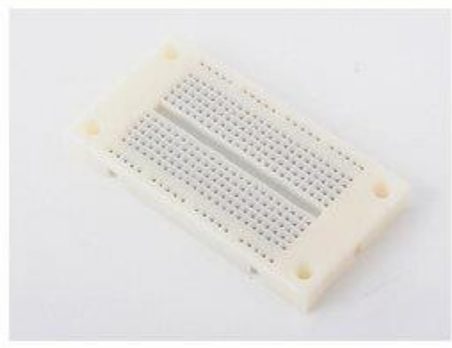
ラジオペンチ 1	USB ケーブル 1	マイコンボード 1	ロボプロシールド 1
			
電池ボックス 1	**ギアドモーター 1**	**マトリクス LED シールド 1**	**スピーカー 1**
			
超音波距離センサー 1	**センサーケーブル 1**	**301 ブレッドボード 1**	**ジャンパー線 65**
			
抵抗 (100 Ω /1k Ω /10k Ω) 各 10	**可変抵抗ボリューム 2**	**タクトスイッチ 10**	**緑色 LED 10**
			

図 0-0　必要なもの

1．ライブラリのはたらき（目安 70 分）

1.0.　各パーツの使用ピン

1）ギアドモーターの使用ピンを調べる

「不思議アイテムⅡ」ではこれまで、マイコンボードに直接ジャンパー線をさし、LED を点灯させたりタクトスイッチのオン・オフを判別したりしていました。
たとえば、2 番ピンに接続（せつぞく）した LED を点灯させたければ、以下のようなプログラムを書いていました。

```
void setup(){
    pinMode(2, OUTPUT);
}

void loop(){
    digitalWrite(2, HIGH);
}
```

まず、「2 番ピンを出力モード（OUTPUT）で使う」という宣言（せんげん） pinMode が必要でしたね。ためしにこの行を削除すれば、LED は点灯しなくなります。

では、これまでジャンパー線を使わずに接続（せつぞく）していた、ギアドモーターなどの電子機器はどのような命令で動いているのでしょうか。
ためしに、マイコンボードにロボプロシールドを取りつけ、MC0 にギアドモーターを接続（せつぞく）します。
その上で、以下のプログラムを書き込（こ）みましょう。

プログラムの書き込み
RoboticsProfessorCourse1 > PreCourse > Motor0

実行結果：MC0 に接続（せつぞく）したギアドモーターが回転する。
1年目の初めに扱（あつか）った、モーターの動作確認プログラムでした。懐かしいですね！
ここで、void loop() 内を書きかえ、以下のようなプログラムにしてみましょう。

```
#include <RPlib.h>

// おまじない(ピン接続設定)
RPmotor mc(MC0);    // MC0につながっているモーターを指定する

void setup(){
}

void loop(){
    digitalWrite(2, HIGH);
}
```

LED を点灯させるのに使っていた digitalWrite です。とはいえ、void setup() 内に pinMode が書かれていないので、このままでは動作しないはずです。
また、モーターを回転させる命令 mc.rotate(100); を消してしまったので、モーターも回転しません。

ここでギアドモーターを取り外し、かわりに以下のような回路(かいろ)をつくってみましょう。

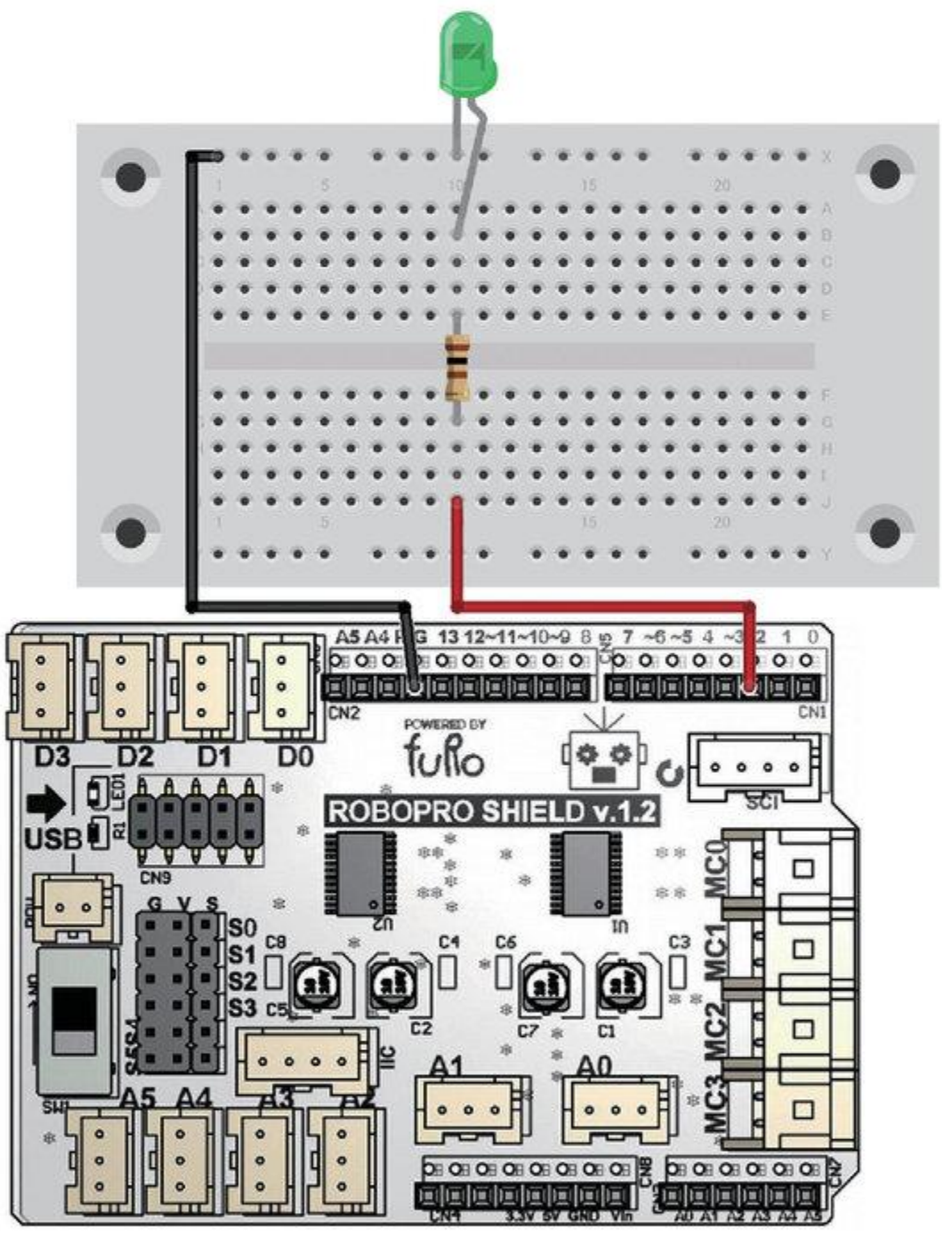


図 1-0　LED を接続(せつぞく)する

⚠注意！
回路(かいろ)をつくる前に、マイコンボードから USB ケーブルを取り外し、電源が切れていることを確認しましょう！

実行結果：2 番ピンの LED が点灯する。

pinMode なしで LED が点灯しています。おかしいですね！

実は、この状況でも LED を点灯させられるピンが、2 番以外にもいくつか存在します。

また、RPmotor mc(MC0); の部分を変更して MC0 以外のモーターを指定すると、LED を点灯させられるピンが変化します。探してみましょう！

やってみよう！

digitalWrite のピン番号を変更して、pinMode なしでも LED が点灯するピンを探してみよう！

また、MC1 、MC2 、MC3 のピンを指定したときも、同様に調べてみよう！

ヒント

0 番ピンと 1 番ピンは、他のピンと少し用途がちがうので、ここでは使用しないようにしよう！

MC0：2 番ピン、3 番ピン、4 番ピン。

MC1：5 番ピン、A0 ピン、A1 ピン。

MC2：6 番ピン、7 番ピン、8 番ピン。

MC3：9 番ピン、12 番ピン、13 番ピン。

まとめると、各モーターを宣言したとき、以下のピンには pinMode なしでも電流を流すことができました。

表　1-0 モーター番号と使用可能ピン

モーター番号	使用可能ピン番号
MC0	2 番、3 番、4 番
MC1	5 番、A0 、A1
MC2	6 番、7 番、8 番
MC3	9 番、12 番、13 番

みなさんはギアドモーターを使用するとき、ロボプロシールド上の MC0 ～ MC3 のコネクターに線を接続していました。
実は、これらのコネクターは裏でマイコンボードのピンにつながっていて、1つのモーターにつき 3 つのピンを使って、モーターを回転させていたのです。
これまでギアドモーターに命令を出すときに書いていた RPmotor という謎のオマジナイは、これら 3 つのピンの使用宣言をまとめて行えるようにしたものです。つまり、RPmotor という命令の中には、pinMode の命令も含まれていたわけです！

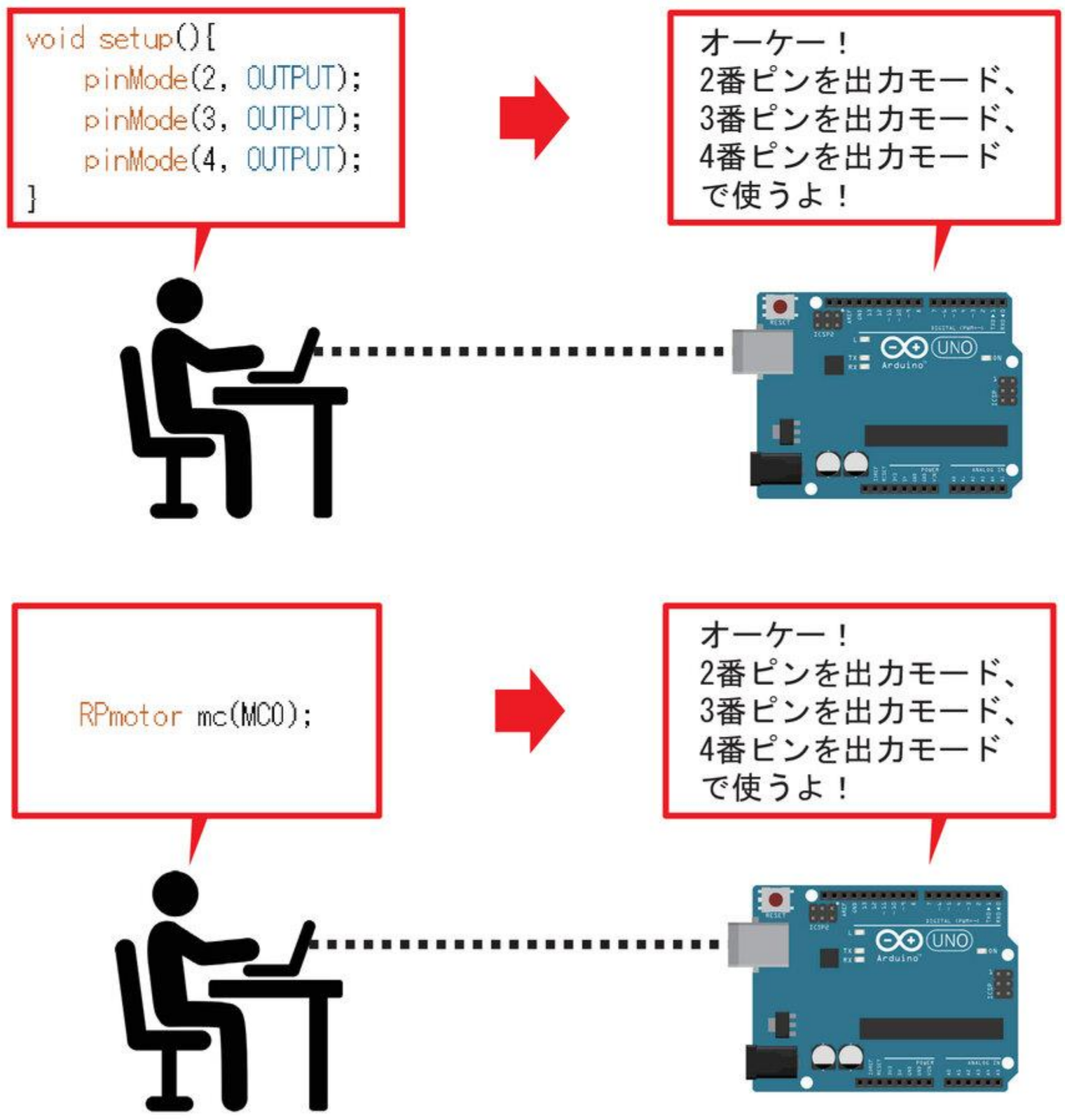


図 1-1　ギアドモーターのオマジナイ（pinMode 部分）

2）ピンの取り合いと分配表

MC0 に接続したギアドモーターが2～4番のピンを使用していることはわかりました。
では、以下のようにしてみるとどうでしょうか。

```
#include <RPlib.h>

// おまじない(ピン接続設定)
RPmotor mc(MC0);    // MC0につながっているモーターを指定する

void setup(){
}

void loop(){
    mc.rotate(100);
    digitalWrite(2, HIGH);
}
```

MC0 のモーターを回転させつつ、2番ピンのLEDを点灯させる、という命令になっています。
実際に、ギアドモーターとLEDを両方接続した状態で実行してみましょう。

実行結果：どちらも動作するが、LEDは暗く、モーターの回転も遅くなる。

どちらも2番ピンからの電流を使っているので、電流のうばい合いから動作がおかしくなってしまっています。
ギアドモーターとLEDはどちらもピンを OUTPUT で使用するので一応動作しますが、もし片方の機器がピンを INPUT で、もう片方の機器が OUTPUT で使用する、というような設定だった場合には、もっと深刻なエラーになってしまうはずです。

ロボプロで使用している各種電子部品は、ギアドモーターのようにマイコンボードのピンをいくつか使用するものが多いため、ロボプロシールドやマトリクスLEDシールドなどのコネクタは、それぞれ裏では決められたピンとつながっています。
同じピンを同時に使ってしまうと、先ほど起こったような不具合が発生してしまうというわけです。
どのコネクタが何番のピンを使用しているかを知っておくことで、複数の電子部品が同じピンを使おうとする、という問題を防ぐことができますね。
次のページに、各ピン・コネクタのマイコンボードの使用ピンを一覧にしたもの（分配表）をのせておきます。

表 1-1　マイコンのピンとロボプロシールドのコネクターの相関図（分配表）

			ロボプロシールド																						マトリクスLEDシールド					姿勢検出シールド				
	マイコンの使用ピン		P10	P9	P2	P3	A0	A1	A2	A3	A4	A5	P2/P3/P4	A0/A1/P5	P6/P7/P8	P9/P12/P13	P0/P1	A4/A5	P6	P7	P8	P5	P2	P3	P10/P11/P12/P13	P0	A3	A2	P1/P11/P13	P1/P11/P13	A4/A5	P0/P9	A2/A3	P10/P12	A4/A5
		ロボプロシールド表記	D0	D1	D2	D3	A0	A1	A2	A3	A4	A5	MC0	MC1	MC2	MC3	SCI	IIC	S0	S1	S2	S3	S4	S5	コントローラー	US0	US1	US2	7セグメントLED	マトリクスLED		ENC0	ENC1	ENC2	ENC3
ロボプロシールド	P10	D0	■																						■									■	
	P9	D1		■												■																■			
	P2	D2			■								■																						
	P3	D3				■							■											■											
	A0	A0					■							■																					
	A1	A1						■						■																					
	A2	A2							■																			■					■		
	A3	A3								■																	■						■		
	A4	A4									■							■													■				
	A5	A5										■						■													■				
	P2/P3/P4	MC0			■	■							■										■	■											
	A0/A1/P5	MC1					■	■						■								■													
	P6/P7/P8	MC2													■				■	■	■														
	P9/P12/P13	MC3		■												■									■				■	■				■	
	P0/P1	SCI（シリアル通信）															■									■			■	■					
	A4/A5	IIC									■	■						■													■				
	P6	S0													■				■																
	P7	S1													■					■															
	P8	S2													■						■														
	P5	S3												■								■													
	P2	S4											■										■												
	P3	S5				■							■											■											
	P10/P11/P12/P13	コントローラ-SPI	■													■									■				☆	☆				■	
マトリクスLEDシールド	P0	US0															■									■						■			
	A3	US1								■																	■								
	A2	US2							■																			■							
	P1/P11/P13	7セグメントLED-SPI														■	■								☆				■	■					
	P1/P11/P13	マトリクスLED-SPI														■	■								☆				■	■					
姿勢検出シールド	A4/A5										■	■						■													■				
	P0/P9	ENC0		■																						■						■			
	A2/A3	ENC1							■	■																							■		
	P10/P12	ENC2	■													■									■									■	
	A4/A5	ENC3	使用しません																																

表1-2　ロボプロシールドの各コネクターの使用ピン（分配表）

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	A0	A1	A2	A3	A4	A5
US0	■																			
US1																		■		
US2																	■			
IIC																			■	■
SCI	■	■																		
A0															■					
A1																■				
A2																	■			
A3																		■		
A4																			■	
A5																				■
D0											■									
D1										■										
D2			■																	
D3				■																
S0							■													
S1								■												
S2									■											
S3						■														
S4			■																	
S5				■																
MC0			■	■	■															
MC1						■									■	■				
MC2							■	■	■											
MC3										■			■	■						
コントローラー											■	■	■	■						
マトリクスLED		■										■		■						
7セグメントLED		■										■		■						

やってみよう！

MC0 にギアドモーターを接続し、さらにロボプロシールド D0 ～ D3 のいずれかのコネクタにタッチセンサーを接続するものとしよう。
このとき、タッチセンサーを接続してはいけないコネクターがあるけど、それはどれかな？
分配表から読みとってみよう！

D2、D3。

1.1. ライブラリを知る

1）「RPlib.h」のはたらき

RPmotor mc(MC0); というオマジナイが、２～４番ピンの使用宣言（せんげん）の代わりになることを学びましたね。
このようなオマジナイを活用すれば、これまでのプログラムもより短く、シンプルにまとめることができそうです。

しかしこの便利なオマジナイを使うためには、ある下準備が必要です。

プログラム「Motor0」より抜粋（ばっすい）

```
#include <RPlib.h>

// おまじない(ピン接続設定)
RPmotor mc(MC0);      // MC0につながっているモーターを指定する

void setup(){
}

void loop(){
    mc.rotate(100);  // 100の速度で正方向に回す！（最大255）
}
```

ためしに黄色の部分を削除してしまうと、プログラムを書き込（こ）む際にエラーが出るようになりますね。
これは、黄色の部分がギアドモーターのオマジナイ RPmotor を使うのに必要な「準備」だからです。

命　　令「#include」
実行内容：指定のヘッダーファイルを取り込（こ）む
使 い 方：#include <RPlib.h>

ギアドモーターのオマジナイ RPmotor は、実ははじめから Arduino に備わった機能ではありません。ロボプロの授業で使うためにつくられた、ロボプロオリジナルの機能です。
3 つの pinMode を１つのオマジナイにまとめ、プログラムをスッキリさせることで、みなさんがプログラムを読んだり書いたりするのをサポートしていたというわけです。
このようなオリジナル機能は、「ライブラリ」とよばれるファイルにまとめられています。

2）ライブラリとは

ギアドモーターにはギアドモーターのオマジナイがあったように、マトリクスLEDにはマトリクスLEDのオマジナイが、スピーカーにはスピーカーのオマジナイがあります。
しかし、ロボプロの授業は毎回スピーカーを使うわけではありません。
ならば、「スピーカーを接続したときだけ、スピーカーのオマジナイを使用可能にする」としたほうが、マイコンの負荷は軽くなるはずですね。
そのため、ロボプロで使う各種オマジナイは、パーツごとに別々のファイルにまとめられています。
このファイルたちの事を「ライブラリ」とよびます。

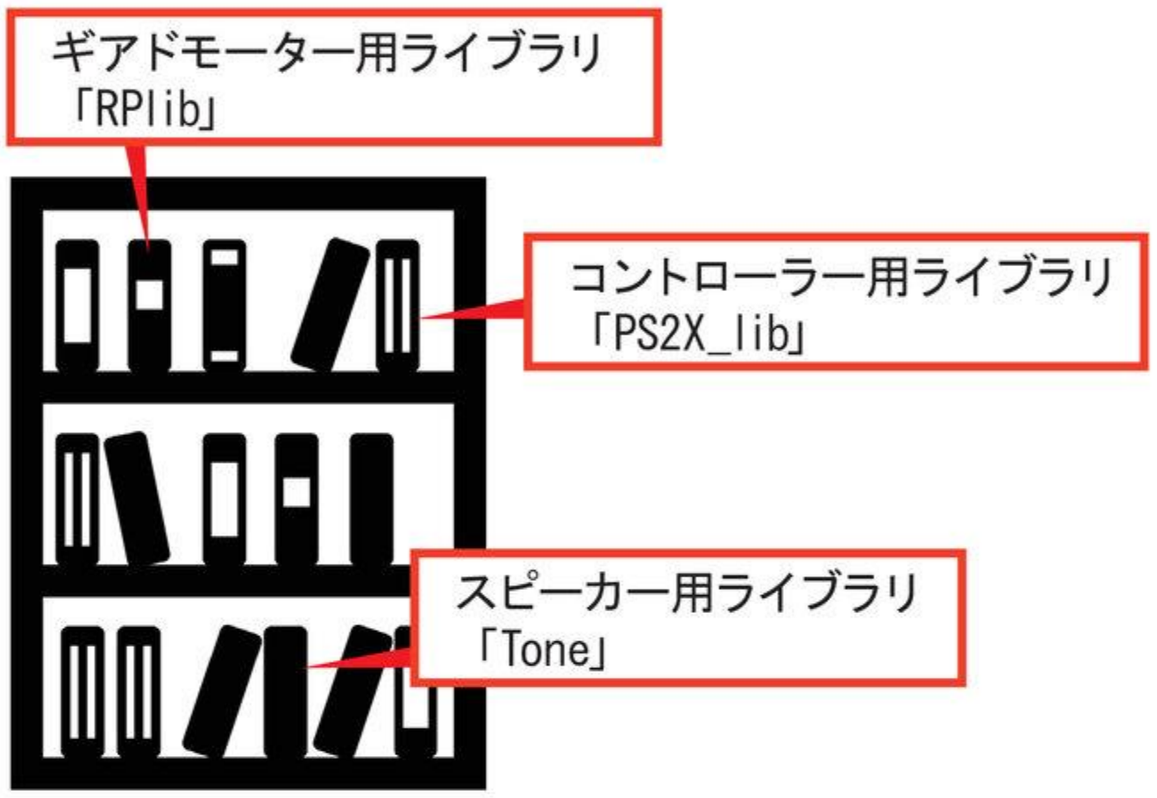


図1-2　別々にまとめられたライブラリ（イメージ）

ライブラリファイルには、「この命令はこういう処理で、この命令は…」というのが詳しくまとめてあります。
Arduinoではこれをいきなり読み込むのではなく、まず「このライブラリにはこういう命令とこういう命令があるよ！」というリストを読み込むことで負荷を減らしています。
このリストのことを「ヘッダーファイル」というのです。

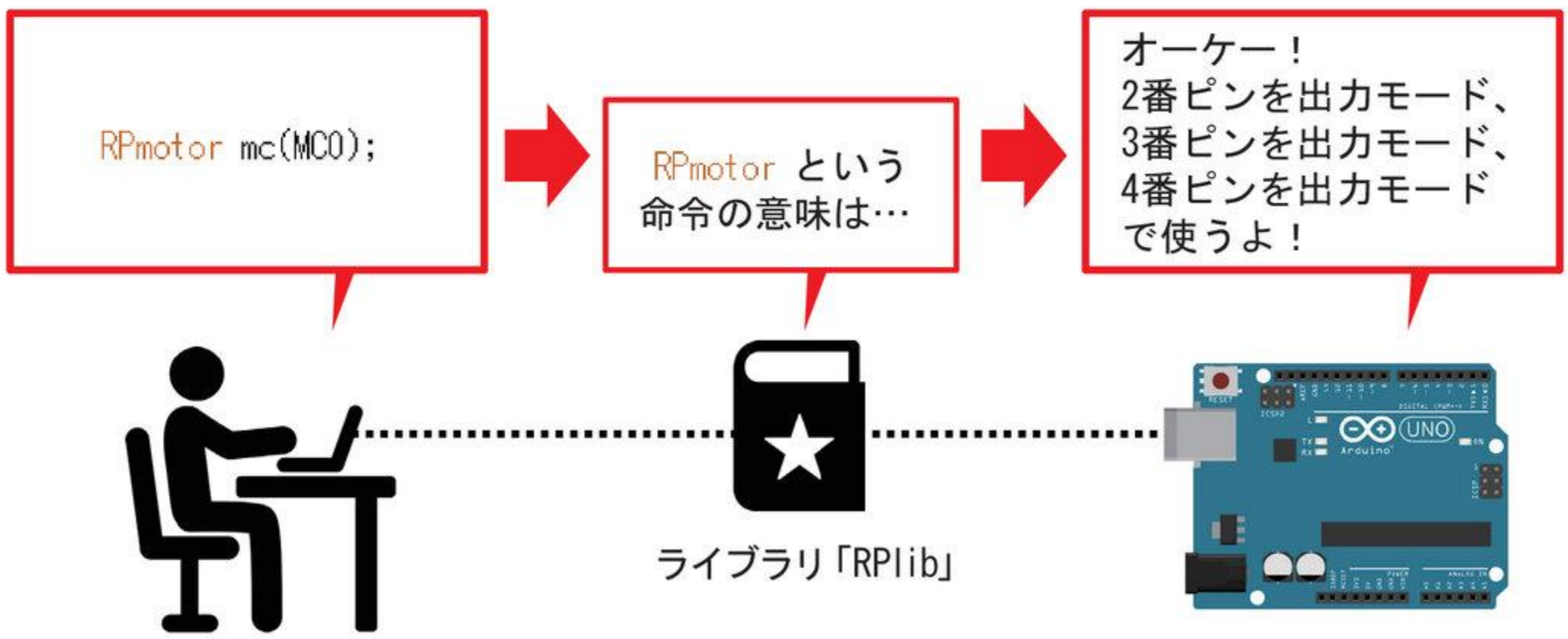


図1-3　ライブラリを活用した命令の出しかた

ロボプロで使用する主なライブラリとヘッダーファイル、そしてライブラリ内の主要なオマジナイ・命令の一例をまとめます。

表 1-3　ロボプロで使用するライブラリ（一例）

使用パーツ	ヘッダーファイル名	オマジナイ	各種命令
ギアドモーター 超音波距離センサー	RPlib.h	ギアドモーター： RPmotor 超音波距離センサー： なし	rotate ギアドモーター回転 ussRead 距離を計測
マトリクス LED	Matrix.h および Sprite.h	Matrix	write 指定のドット点灯 putch 指定の文字を表示 putd2 2 ケタの数を表示 clear マトリクス LDE を消灯
7 セグメント LED	LedControl.h	LedControl	setDigit 指定のケタを表示 setDec 指定の値を表示 setLed 指定のセグメントだけを点灯 clearDisplay 全てのセグメントを消灯
コントローラー	PS2X_lib.h	PS2X	read_gamepad コントローラーの状態を取り込む Button 指定のボタンの状態を見る Analog アナログスティックの傾き具合を見る
オムニホイール	RPomniDirect.h	RPomniDirect	move オムニホイール ロボットを移動させる
スピーカー	Tone.h	Tone	begin スピーカーを指定したコネクタで使う play 指定の周波数の音を鳴らす play_rtttl 指定のメロディを流す

チャレンジ課題

「MC1 ギアドモーターを回転させつつ、A2 のスピーカーからも音が出る」というプログラムを一からつくってみよう！

ヒント

ギアドモーター、スピーカーともに「ヘッダーファイルの取り込み」「オマジナイ」「動作命令」の3つが必要だね！
スピーカーはオマジナイ以外にも、void setup() 内にコネクタ指定の命令 begin が必要だよ！

講　解答例は巻末に記載します。

2. シリアルモニターを活用する（目安 30 分）

さて、最後に、前回のラストで少しだけ扱った「シリアルモニター」の機能をしっかり理解しておきましょう。
前回と同じプログラムをもう一度実行します。

プログラムの書き込み
RoboticsProfessorCourse2 > MagicItemA5 > Serialcom1

書き込めたら、USB ケーブルを抜かず、画面右上の虫メガネのアイコンをクリックし、右下の選択欄を「9600 baud」にしておきましょう。

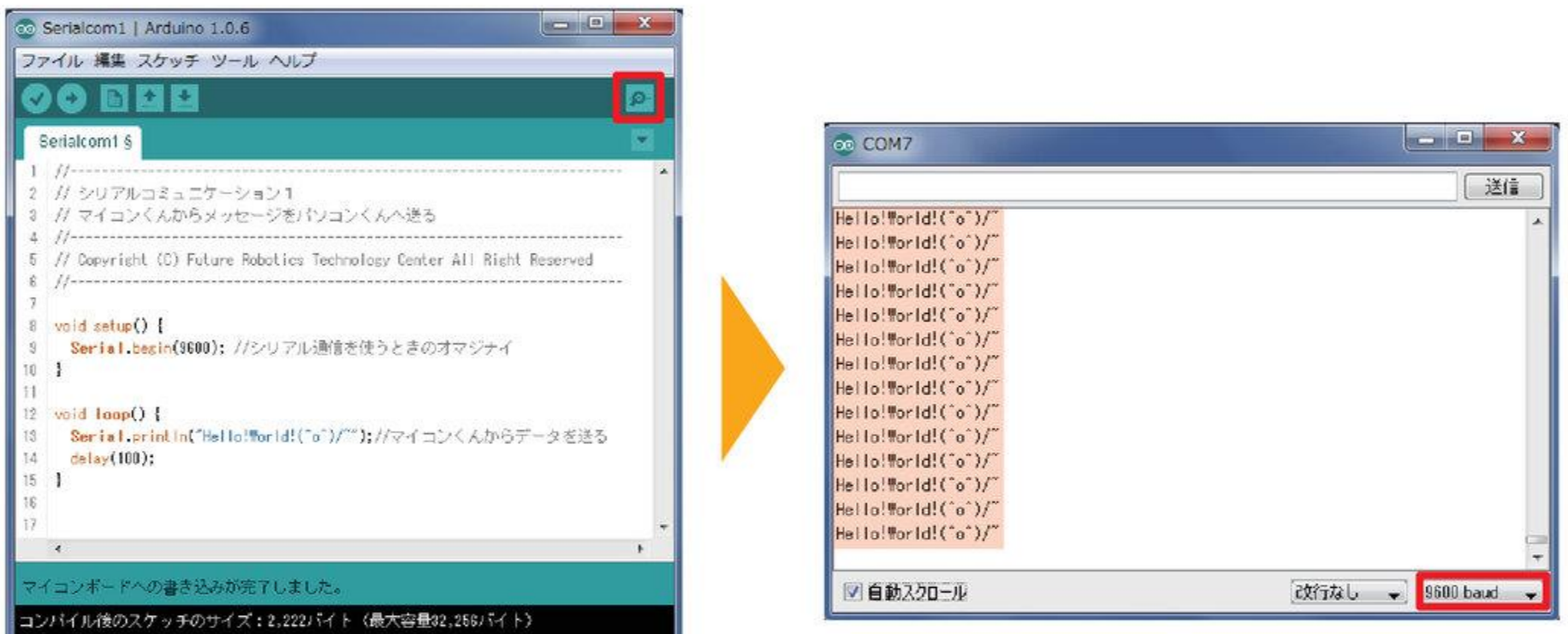


図 2-0　シリアルモニターの表示方法と表示画面

あらためて、このプログラムを見てみます。

プログラム「Serialcom1」より抜粋

```
void loop(){
    Serial.println("Hello!World!(^o^)/~");  // マイコンくんからデータを送る
    delay(100);
}
```

命　　令「println」
動作内容：シリアルモニターに指定の文字・数値を表示したあと、改行する
使 い 方：Serial.println("Robopro");
　　// シリアルモニターに「Robopro」という文字列を表示する

文字列だけでなく、センサーの状態などを表示することもできます。
ブレッドボードに回路(かいろ)をつくり、タクトスイッチからの信号が8番ピンに流れるようにしてみましょう。抵抗は100 Ωのものを使います。
その上で、以下のようにプログラムを書きかえます。

```
void loop(){
    Serial.println(digitalRead(8));  // マイコンくんからデータを送る
    delay(100);
}
```

POINT

8番ピンの使用宣言(せんげん)も書き足しましょう。

実行結果：シリアルモニターに「0」が連続して表示される。タクトスイッチを押している間は、「0」のかわりに「1」が表示される。

スイッチが押されているとき、つまり8番ピンに信号があるときだけ「1」、そうでないときは「0」が表示されます。
ちなみに、プログラムに書いた文字列をそのまま表示させたいときは文字列を `""` で囲みます。

println 命令には改行の処理が含まれるので、1行ずつしか表示されません。

さらに、以下のように変えてみましょう。

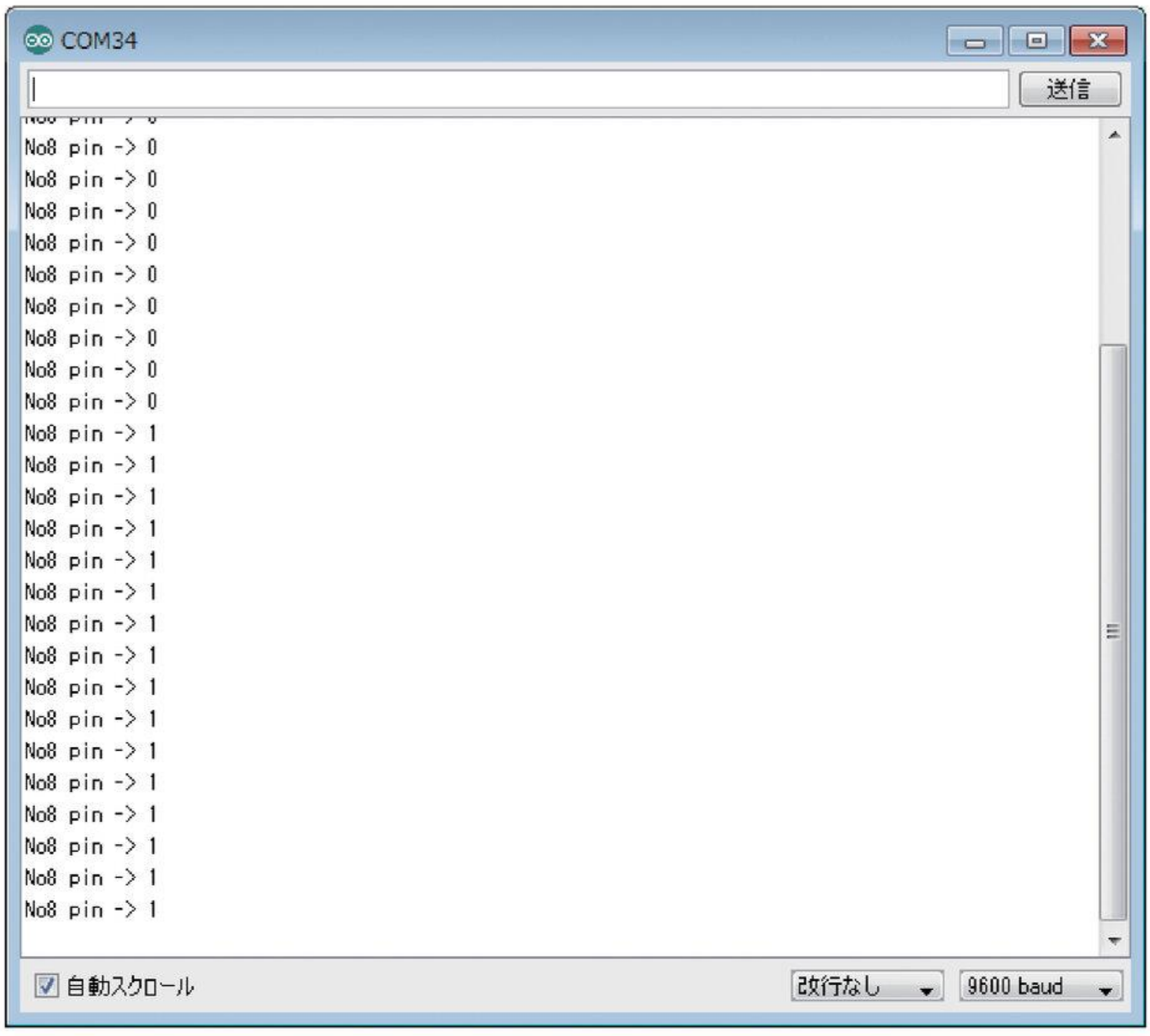
```
void loop(){
    Serial.print(digitalRead(8));  // マイコンくんからデータを送る
    delay(100);
}
```

実行結果：数字が同じ行に続けて表示される。

print 命令は、println 命令から改行の処理を除いたものです。続きの文を同じ行に表示させることができますね。

やってみよう！

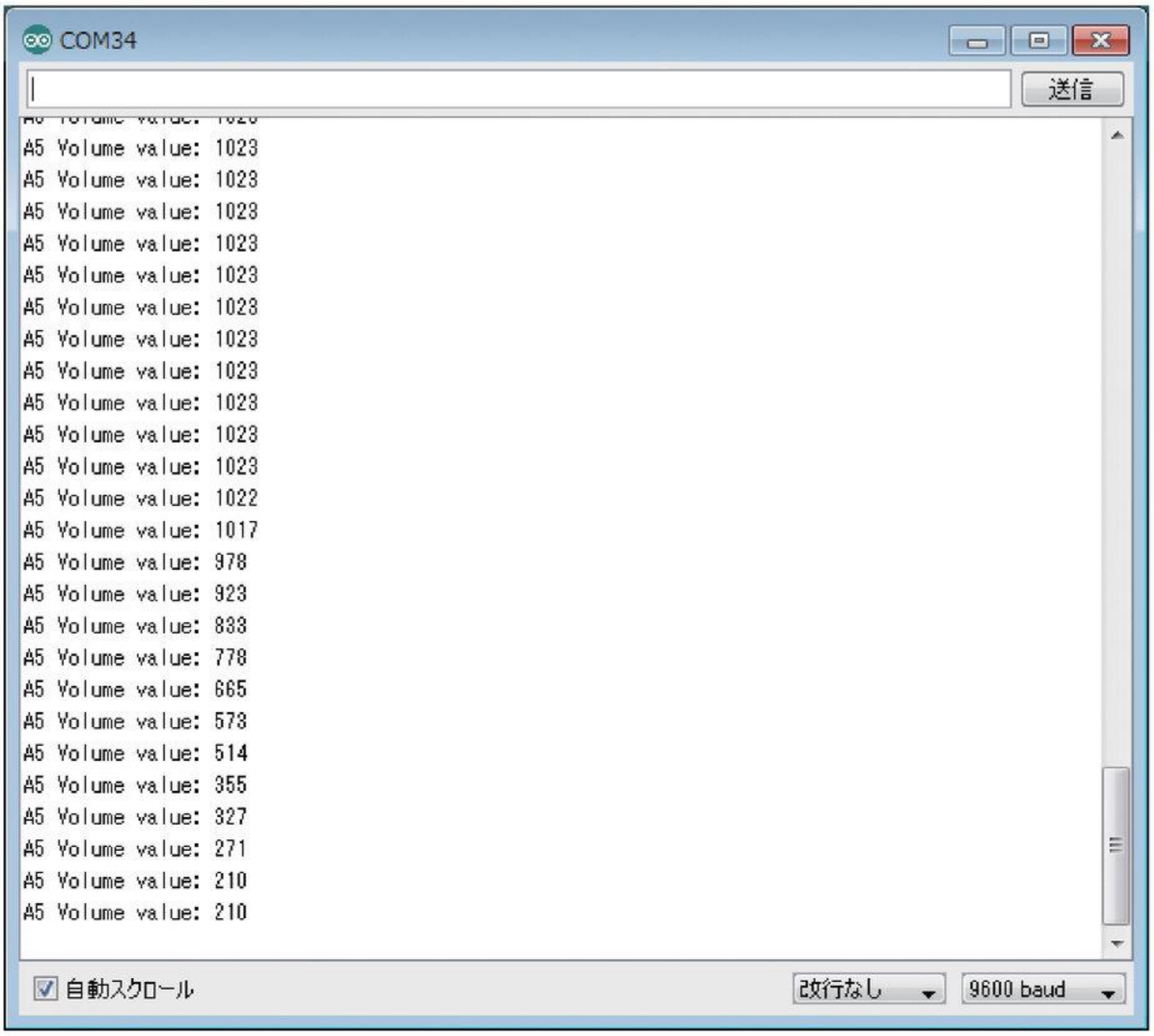
シリアルモニターに表示される文字列を、以下のようにできるかな？



講　解答例は巻末に記載します。

ステップアップ

A5 ピンに接続した可変抵抗ボリュームの値を、以下のように表示できるようにしよう！



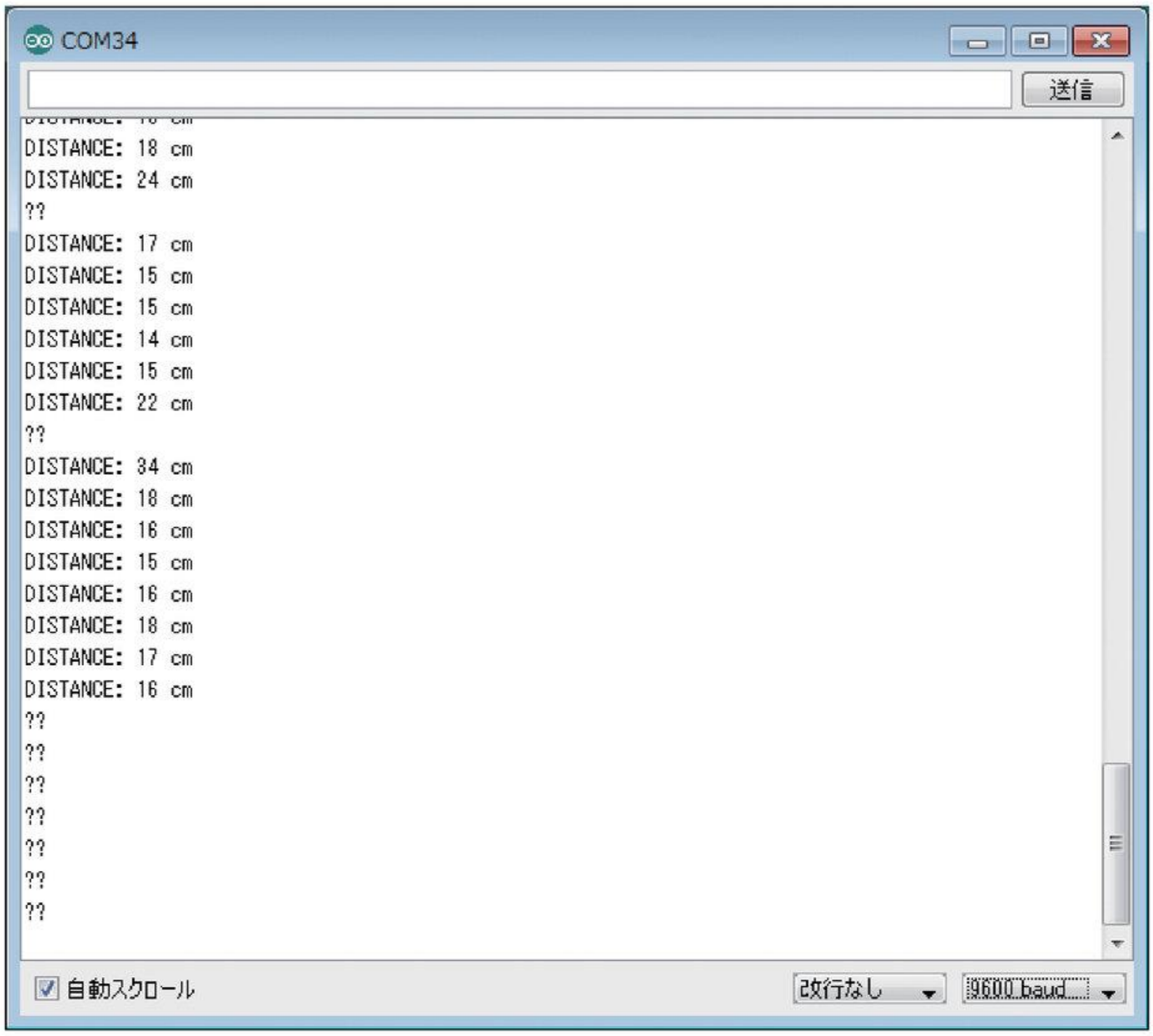
講

解答プログラムは以下となります。
RoboticsProfessorCourse2 > MagicItemA5 > Serialcom3

回路の解答例は巻末に記載します。

チャレンジ課題

マトリクス LED シールドと超音波距離センサーを接続し、計測した距離を図のようにシリアルモニターに表示できるようにしてみよう！
さらに、物体までの距離が遠い（1m より遠い）ときは、距離のかわりに「??」が表示されるようにしてみよう。



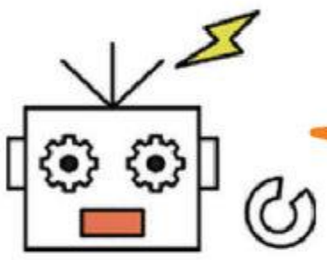
ヒント

超音波距離センサー用の命令を使うには、ライブラリが必要だったね！

講　解答例は巻末に記載します。

3．まとめ（目安 5 分）

さて、ライブラリの使い方がなんとなくわかってきたでしょうか？ 今回は、マイコンのこと、プログラミングのことをさらに深く勉強しました。これがわかれば、さまざまなセンサー、モーターを接続して、複雑な動きができるロボットをプログラミングしたり、自分でセンサーを買ってきて接続したりすることもできます！　オリジナルロボットをつくるには必須の技術になりますので、くり返し勉強していきましょう。



講

○　以下の授業の目標を再確認します。
　・マイコンについて理解を深める
　・ライブラリの使い方を学ぶ
　・複数のパーツを同時に使う
○　今回のターム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、2 年目コースは「アームロボット」、3 年目コースに進級する場合は「六脚ロボット」になります。

≪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	ドライバー 1	USB ケーブル 1	マイコンボード 1
			
ロボプロシールド 1	A-1 (アームロボットパーツ) 1	A-2 (アームロボットパーツ) 1	A-3 (アームロボットパーツ) 1
			
A-4 (アームロボットパーツ) 1	A-5 (アームロボットパーツ) 1	A-6 (アームロボットパーツ) 1	A-7 (アームロボットパーツ) 1
			
B-3 (アームロボットパーツ) 1	B-4 (アームロボットパーツ) 1	B-5 (アームロボットパーツ) 1	B-6 (アームロボットパーツ) 1
			
C-1 (アームロボットパーツ) 1	C-2 (アームロボットパーツ) 1	C-3 (アームロボットパーツ) 1	C-4 (アームロボットパーツ) 1
			
C-5 (アームロボットパーツ) 1	C-6 (アームロボットパーツ) 2	C-7 (アームロボットパーツ) 1	C-8 (アームロボットパーツ) 1
			

図 3-0　次回必要なもの①

ベアリング (S) 19	ベアリング (M) 1	MG995 サーボモーター 2	サーボモーター付属パーツ 2
			
SG90 マイクロサーボモーターセット 1	ジュラコンブッシュ 2	M3L12 タッピングネジ (B) 2	AC アダプター 1
			
M2L12 ネジ 2	M2 ナット 2	M3L12 タッピングネジ (B) 2	M2L6 タッピングネジ (B) 8
			
M3L10 タッピングネジ (B) 2	M3L6 フラットヘッドビス 15	M3L8 タッピングネジ (B) 8	
			

図 3-1　次回必要なもの②

P.13　チャレンジ課題　解答例

```
#include <RPlib.h>
#include <Tone.h>

RPmotor mc(MC1);
Tone tone1;

void setup(){
    tone1.begin(A2);
}

void loop(){
    mc.rotate(100);
    tone1.play(440);
}
```

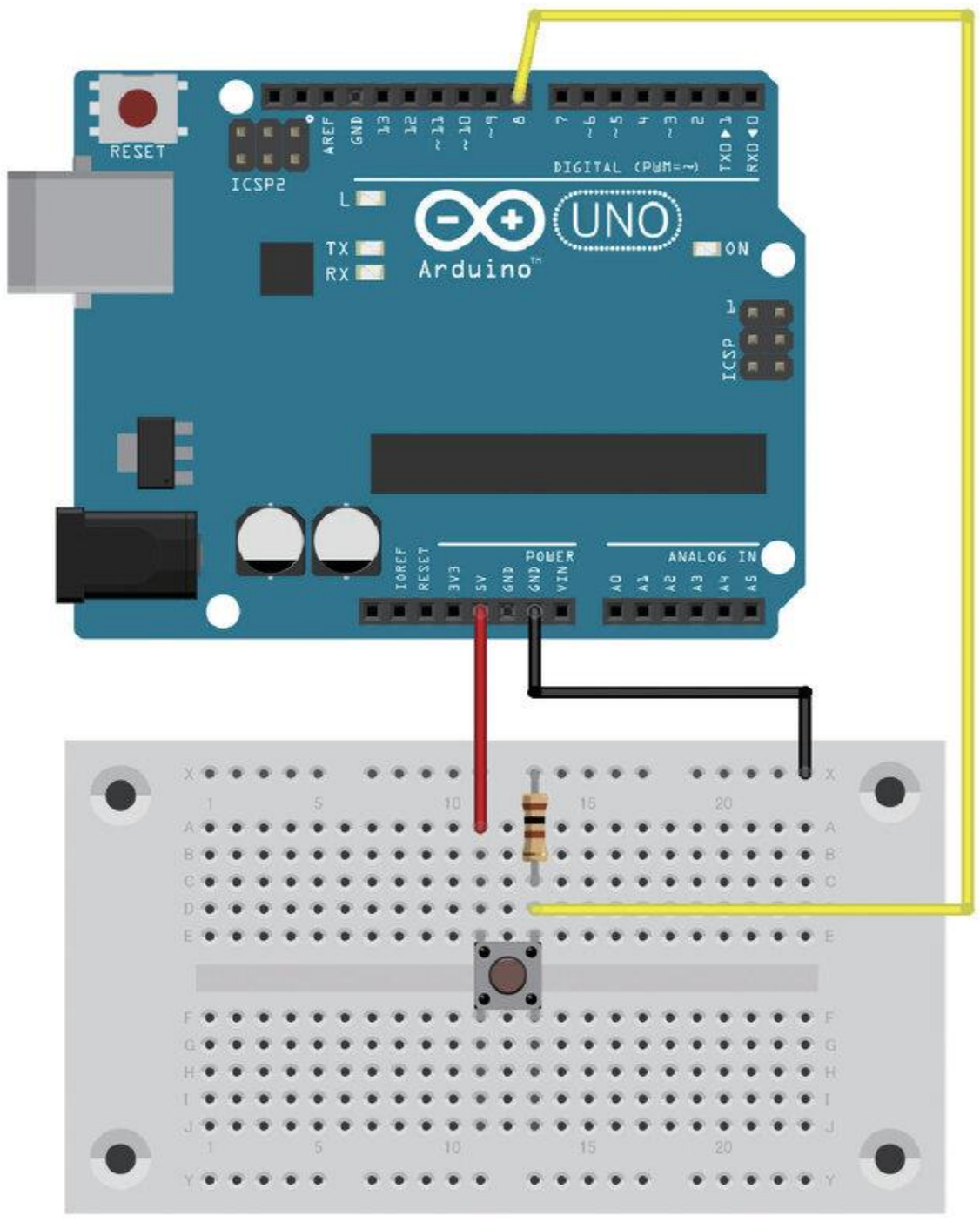
※ tone1.play の引数 440 は出す音の周波数です。440Hz の音は「ラ」を示します。

P.16　やってみよう！　解答例

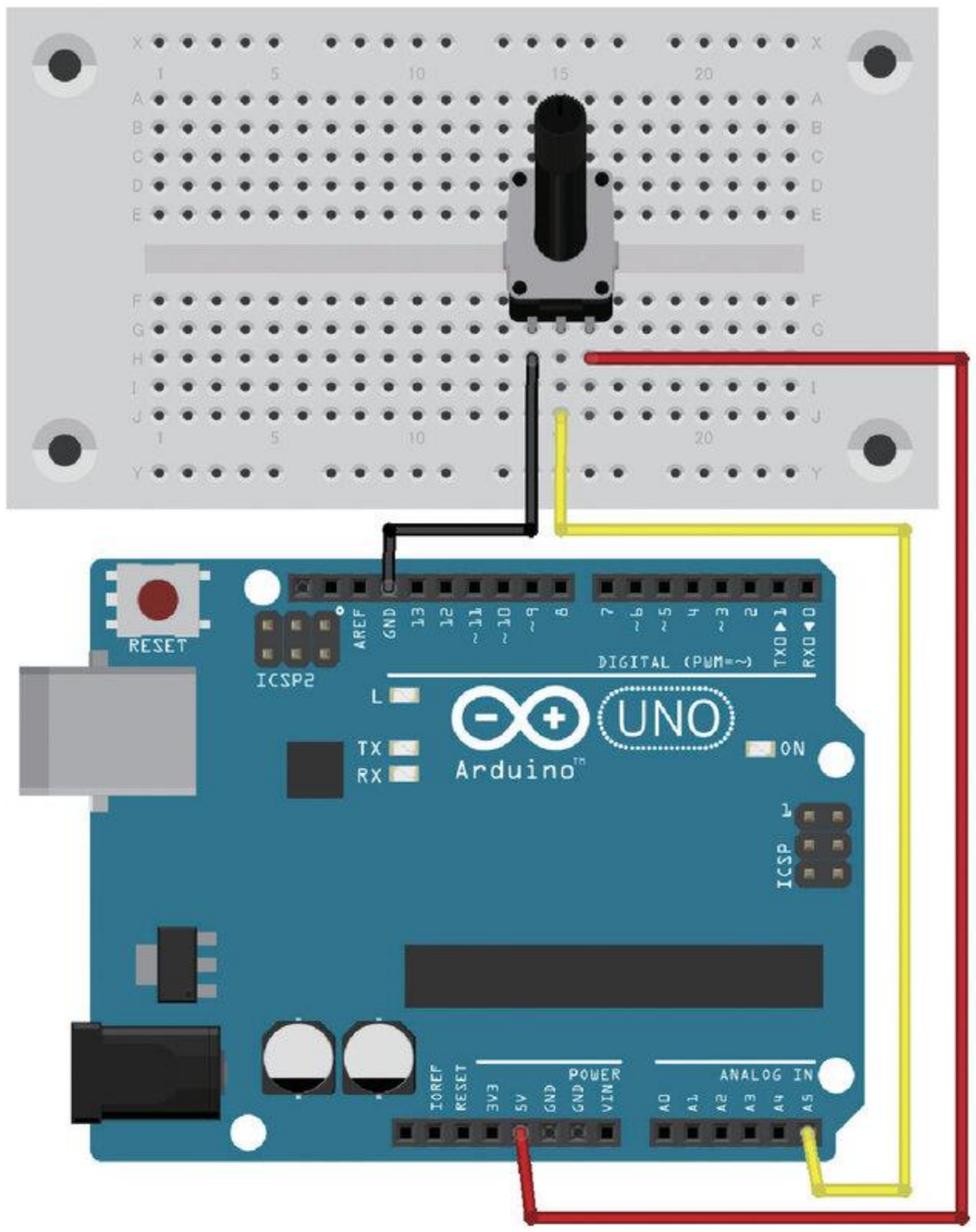
```
void setup(){
    Serial.begin(9600);
    pinMode(8, INPUT);
}

void loop(){
    Serial.print("No8 pin -> ");
    Serial.println(digitalRead(8));
    delay(100);
}
```

※回路図の解答例は以下となります。



P.17　ステップアップ　解答例



P.18　チャレンジ課題　解答例

```
#include<RPlib.h>


void setup(){
  Serial.begin(9600);
}


int d = 0;
void loop(){
  d = ussRead(US1);
  if(d > 100){
    Serial.println("??");
  }


  else{
    Serial.print("DISTANCE:");
    Serial.print(d);
    Serial.println("cm");
  }
  delay(100);
}
```