

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

センサーロボット③

第6回

エンコーダーを使ったロボット

講師用

目 次

0. エンコーダーを使ったロボット

0.0. 「エンコーダーを使ったロボット」でやること

0.1. 必要なもの

0.2. 動作確認と調整

1. エンコーダーロボットの組み立て

1.0. 完成形の確認

1.1. オムニキャストターの組み立て

1.2. エンコーダーロボット本体の組み立て

1.3. エンコーダーロボットを動かしてみよう

2. 赤外線ライトレーサー

2.0. 赤外線ライトレーサーへ改造

2.1. 赤外線ラインセンサーの調整

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。

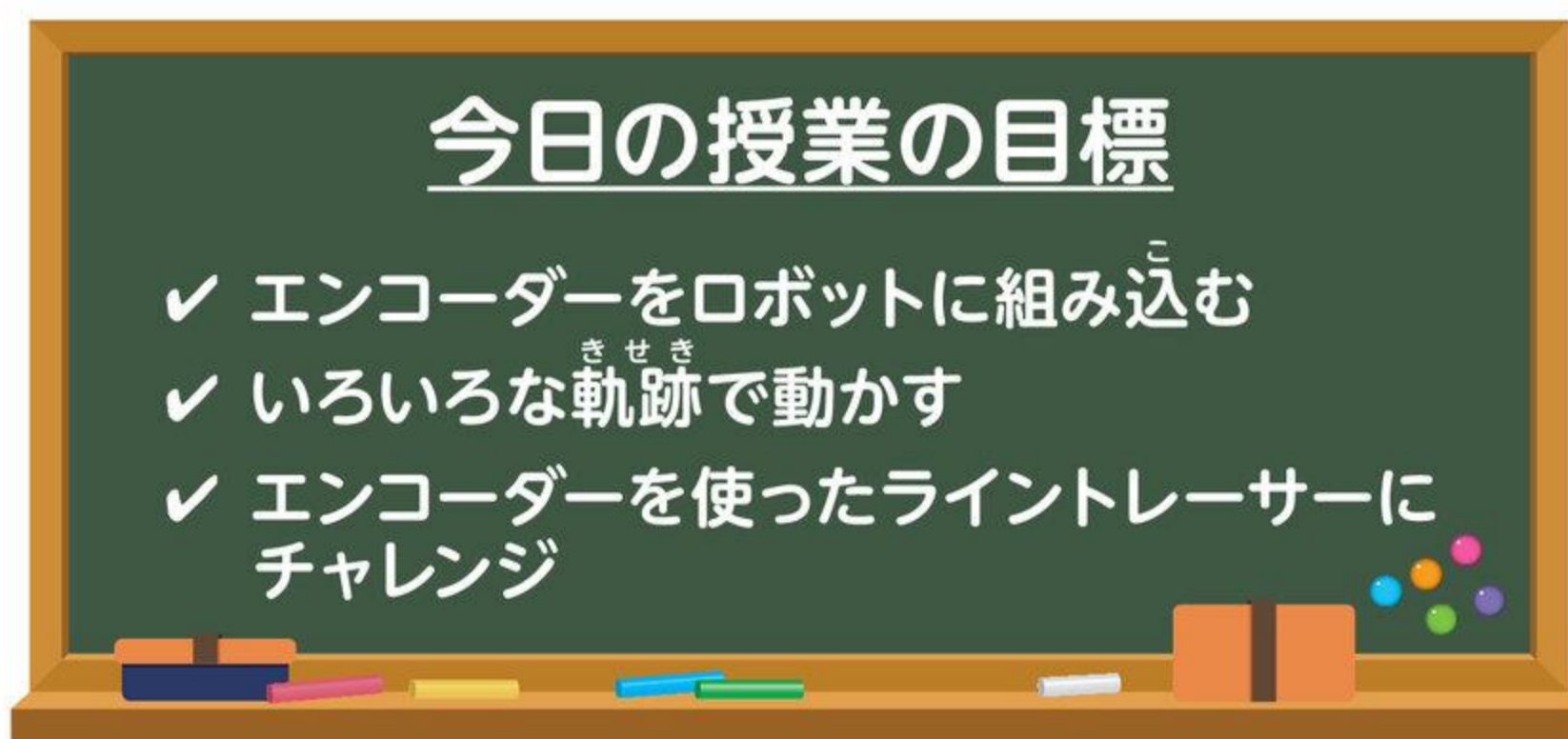
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(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. エンコーダーを使ったロボット（目安10分）

0.0. 「エンコーダーを使ったロボット」でやること

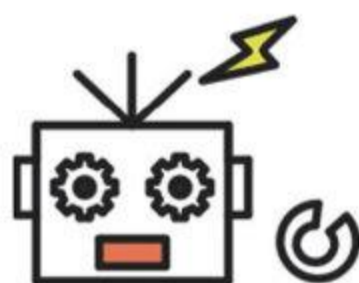


第5回では、赤外線反射によって白黒のシマを検出する「エンコーダー」という装置の使い方を学びました。

このエンコーダーを活用することで、モーターの回転方向と速度の情報を得ることができました。

今回は、このエンコーダーの機能をロボットに組み込んで、「いろいろな軌跡で動かす」とこと、「ライトレーサー」にチャレンジしていきましょう。

少し面倒に感じる計算も必要ですが、数学を応用すればロボットを思った通りに動かすことができますよ。良いロボットを作るまでの道はなかなか険しく、うまくいかないこともあるかもしれませんが、トライ&エラーで、高精度のロボットを目指しましょう！



赤外線センサーの利用方法は、いろいろとあるんだよ！

0.1. 必要なもの

今回は、1年目の「不思議アイテム I-2」で製作した、二輪駆動の「ベースロボット」にエンコーダーを組み込みます。まずは、ベースロボットを完成させて、前回使ったエンコーダー付きのモーターを組み付けます。以下のものを準備してください。

ラジオペンチ	1	ドライバー	1	USB ケーブル	1	電池ボックス	1
							
センサーL字ステイ	3	オムニホイール	1	M3L30 ネジ	1	白円形ボード	1
							
M3 ナット	19	M3L8 ネジ	22	8mm 角スペーサー	4	30mm 角スペーサー	3
							
ユニバーサルボード	1						
							

図 0-0 必要なもの

0.2. 動作確認と調整

まずは前回モーターに組み込んだエンコーダーの精度の再チェックを行いましょう。テストプログラムは、それぞれ以下となります。配線は前回のテキストを確認しましょう。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > CombRobot5 > encoderTest

RoboticsProfessorCourse2 > CombRobot5 > encoderMotorTest

また、オムニホイールの組み付けに使用されているネジの締めつけ具合なども今一度調整しておきましょう。

1. エンコーダーロボットの組み立て（目安 55 分）

1.0. 完成形の確認

図 1-0 は今回組み立てるエンコーダー付きロボットの完成形です。前回組み立てをしたエンコーダー付きのギアドモーターを左右に組み付けて、モーターの回転数を読み取りながらロボットの移動量をコントロールすることができます。

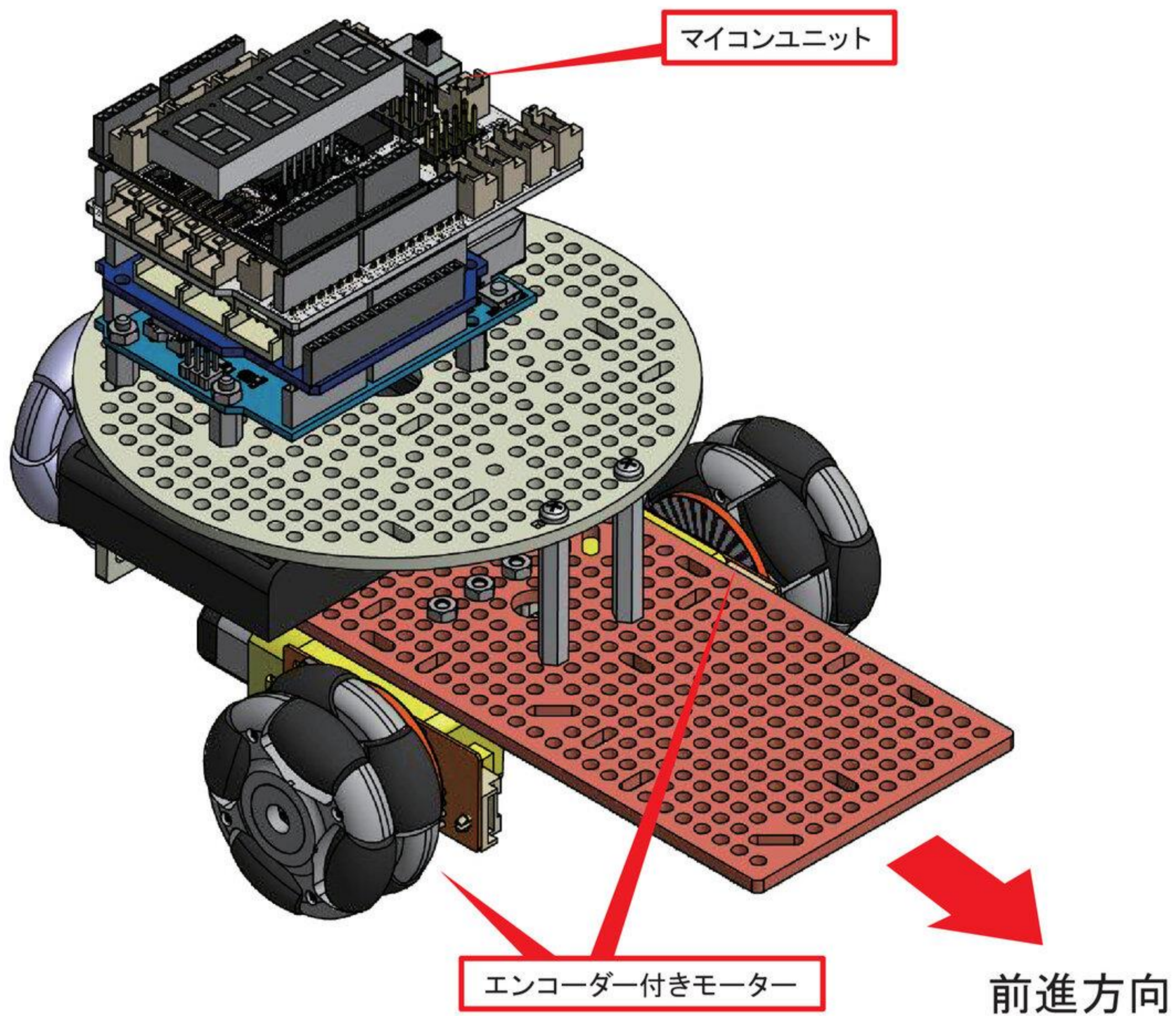


図 1-0 エンコーダーロボットの完成形

講

マイコンユニットの接続は、前回と同様です。下からマイコンボード、姿勢検出シールド、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の順です。（ただし、姿勢検出シールドは、ロボプロシールドの上下どちらに接続しても動きに影響はありません。）

なお、電池残量により、ロボットの動きに影響が出る場合がありますので、予備の電池の準備を推奨してください。

1.1. オムニキャスターの組み立て

まず、はじめにエンコーダーロボット後方に取り付ける、オムニキャスターの組み立てから行いましょう。

M3L30 ネジをオムニホイールに通し、M3 ナットで固定します。さらにセンサーL字ステイを通し、M3 ナットで固定します。

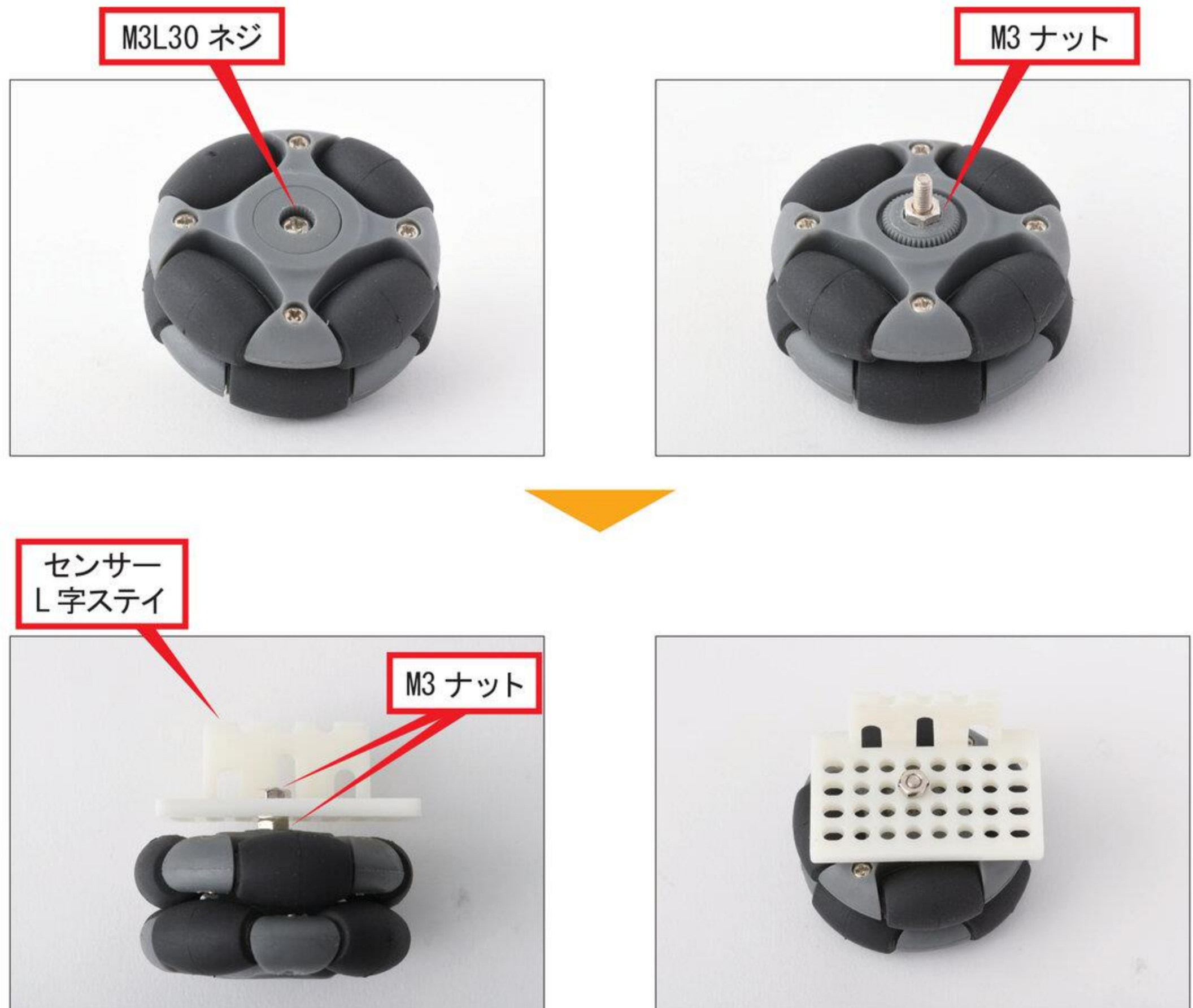


図 1-1 オムニキャスターの組み立て

講

ロボットの走行中に、オムニキャスターのナットが緩むことがあります。緩んだ際には、固定し直してください。

1.2. エンコーダーロボット本体の組み立て

やってみよう！

さあ、腕試しだ！ 図1-2～図1-4をヒントに、エンコーダーロボットを組み立てよう。

💡 ヒント

各パーツの取り付けのネジは、すべてM3L8ネジを使うよ。また、ネジの取り付け位置や使用するパーツなどは図と多少異な^{こと}っていても、ロボットが動けば大丈夫だよ。

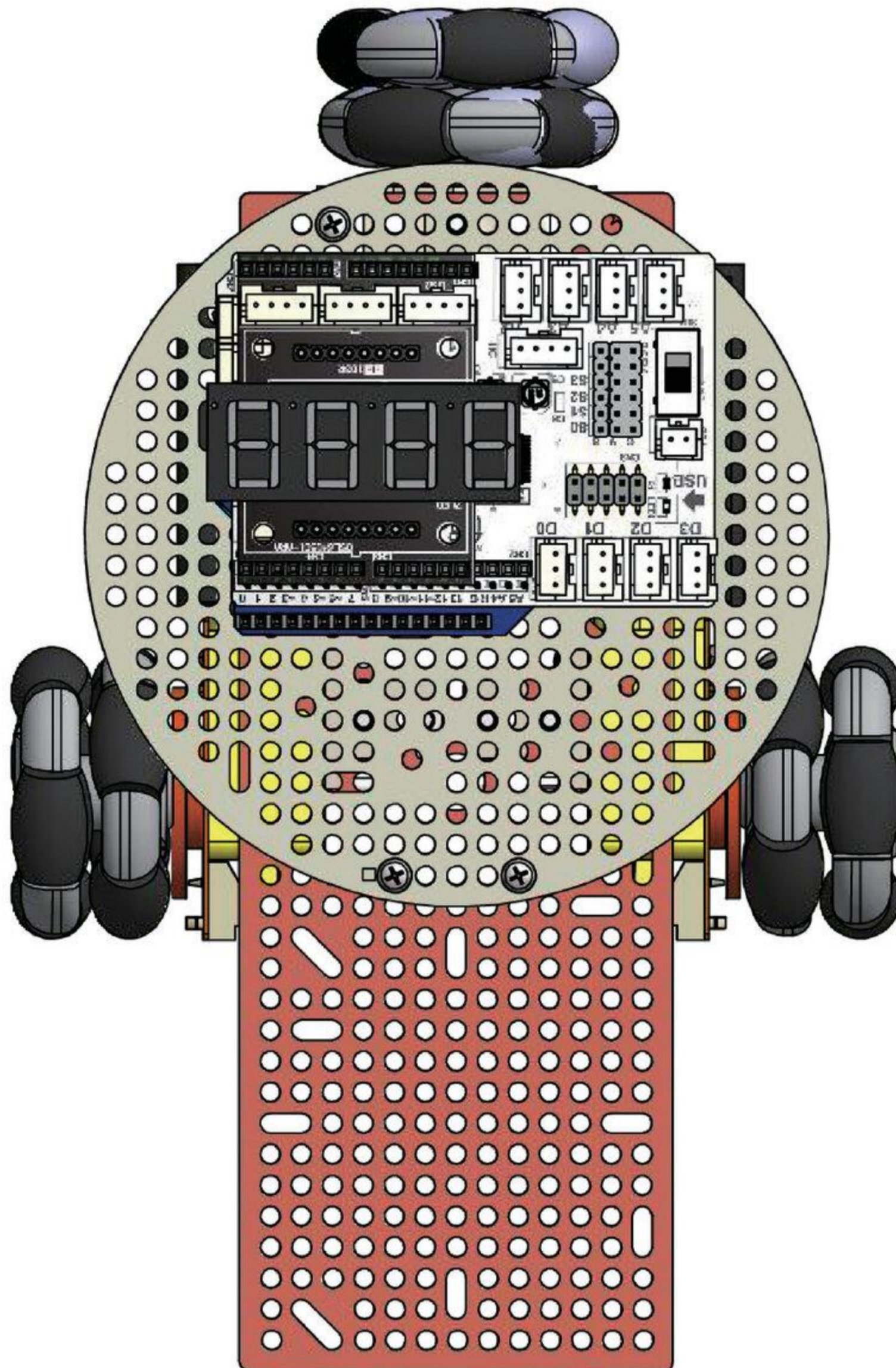


図1-2 エンコーダーロボット（上面の図）

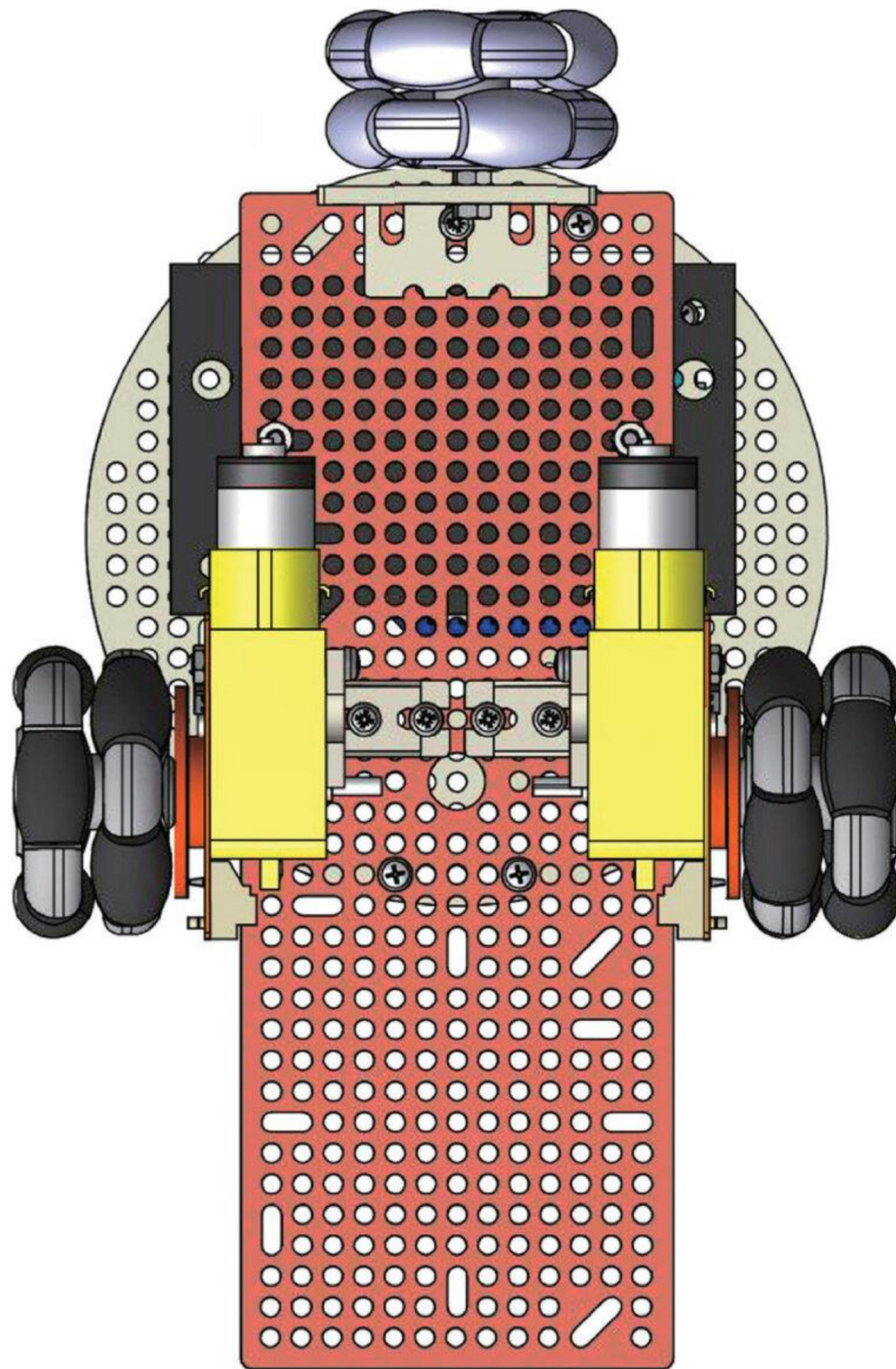


図 1-3 エンコーダーロボット (下面の図)

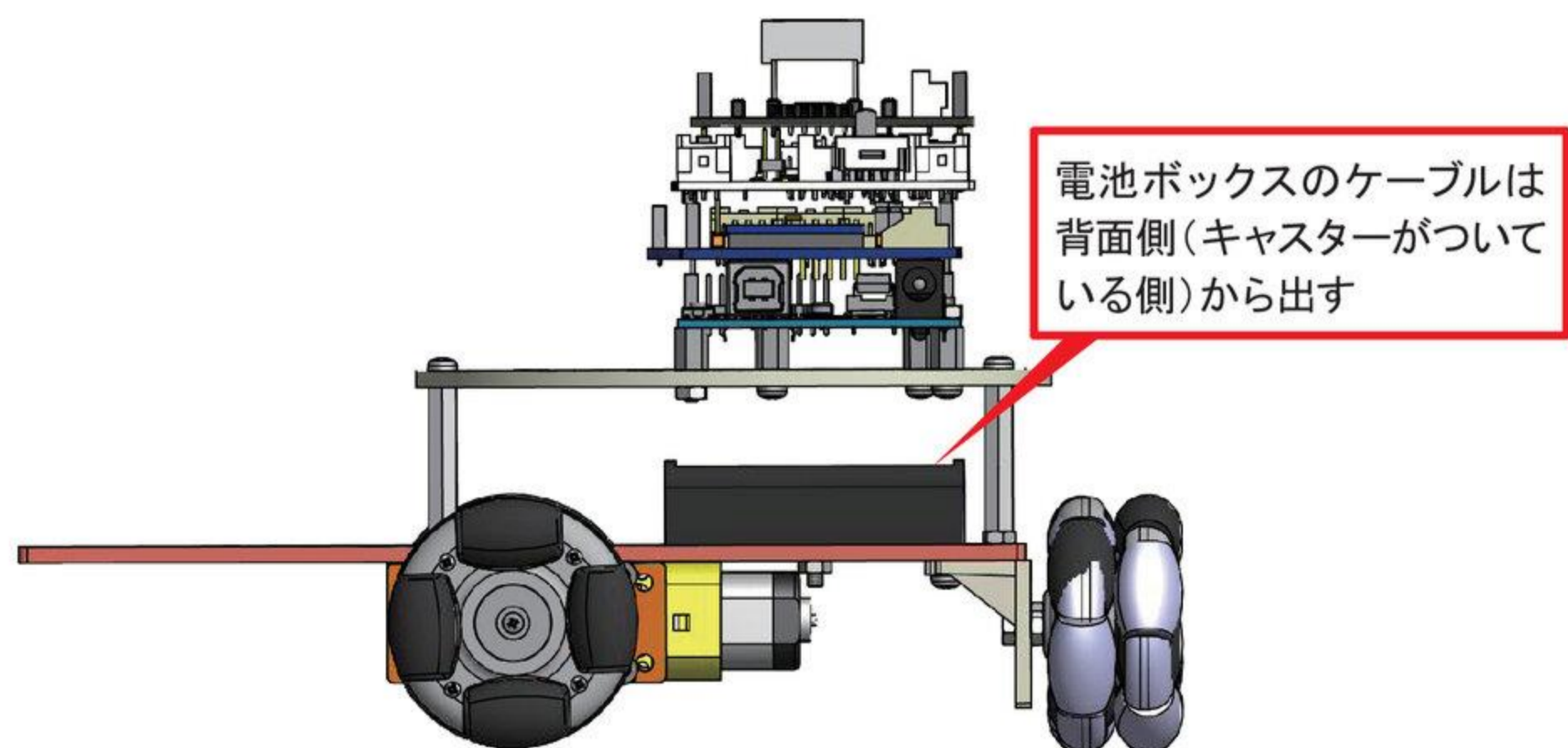


図 1-4 エンコーダーロボット (側面左の図)

1.3. エンコーダーロボットを動かしてみよう

1) 配線

エンコーダーロボットの組み立てが完了したら、以下のように配線をしましょう。



POINT

各部の配線

- ・姿勢検出シールド `ENC1` : 右モーターのエンコーダー
- ・姿勢検出シールド `ENC2` : 左モーターのエンコーダー
- ・ロボプロシールド `MC1` : 右モーター
- ・ロボプロシールド `MC2` : 左モーター
- ・ロボプロシールド `D2` : タッチセンサー (動作スタートで使用します)

2) エンコーダーを使った動作プログラム

それではエンコーダーロボットを動かしていくため、まずは前回少しだけ扱ったエンコーダー用の動作命令について、改めて学んでおきましょう。前回のプログラムを再び開いてみましょう。

プログラムの書き込み

RoboticsProfessorCourse2 > CombRobot5 > encoderMotorTest

プログラム「encoderMotorTest」より抜粋

```
rotate_enc(-SPEED_L, 1); // 引数1、速度、引数2、チャンネル
```

`rotate_enc();` という、新たな命令が登場していましたね。

命令「rotate_enc(x, y);」

実行結果：エンコーダー番号と速度を指定し、白黒検知を行いながらモーターを回転させる

使い方：rotate_enc([モーターの回転速度] , [使用するモーター]);

第4回で勉強した「引数」が2つありますね。第1引数（黄色の部分）はモーターの回転速度を指定しますが、RPMの値をそのままかくわけではありません。たとえば50RPMの速度で回転させるとして、`rotate_enc(50, 2);` とかいてもうまくいかないということです。30、50、70RPMを指定する場合、第1引数には次のような値を入れる必要があります。

表 1-0 RPM と第 1 引数^{ひきすう}の関係

目標速度 [RPM]	第 1 引数 ^{ひきすう} の値
30	186
50	235
70	253

複雑^{ふくざつ}な計算式を用いて第 1 引数^{ひきすう}を求めているため、キリの良い値ではなく覚えづらいです。しかしこのままでは「速度を 70RPM に切り替えたいけど、値はいくつだったっけ…」などと、毎回悩むことになってしまいます。何とかしたいですね。

ここで、「encoderMotorTest」を開いた画面の上部にある「enc.h」というタブを見てみましょう。

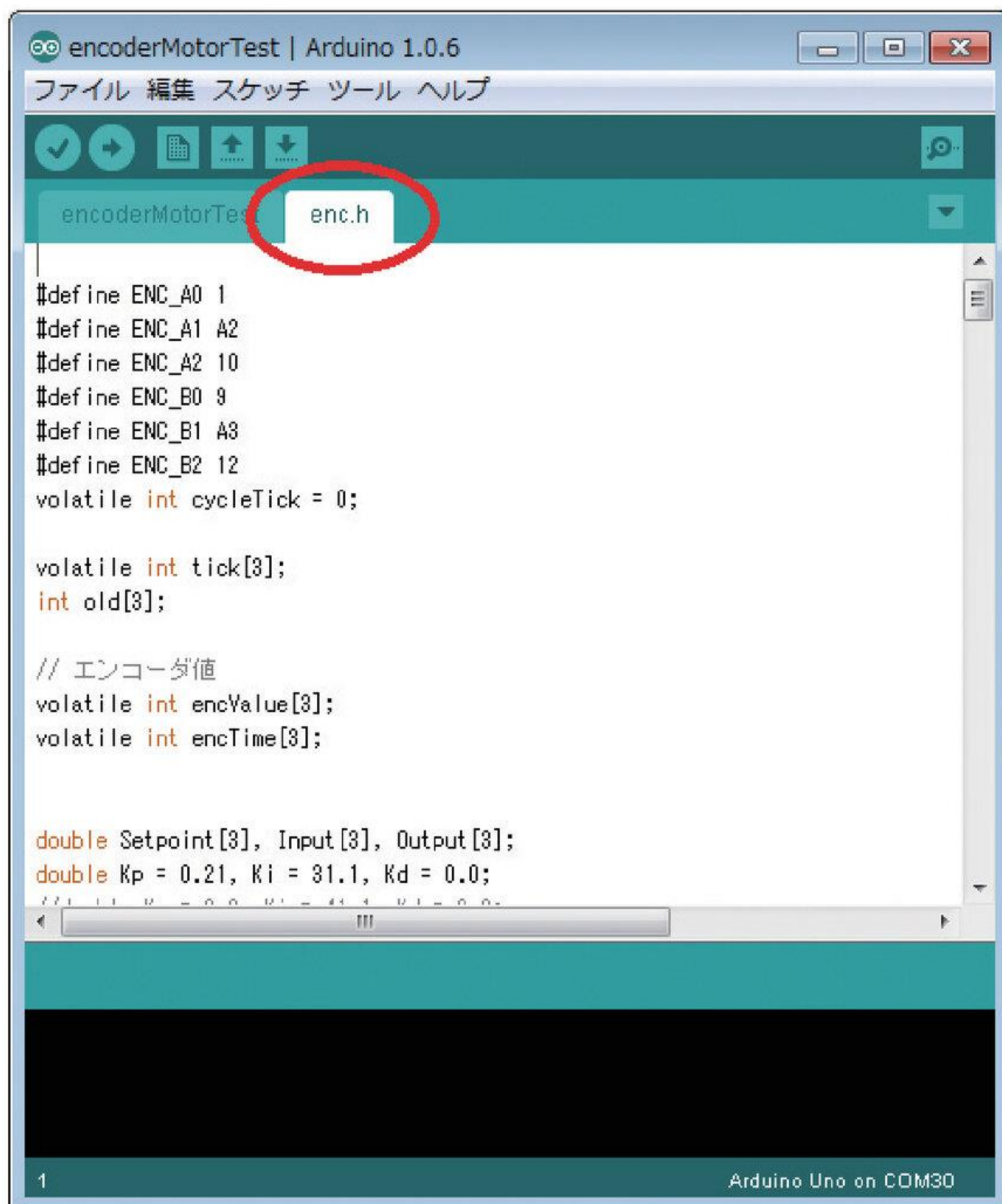


図 1-5 「enc.h」を確認する

以下のような文がかかれています。確認してみましょう。

□ ヘッダーファイル「enc.h」より^{ぼっすい}抜粋

```
#define SPEED_L 186
#define SPEED_M 235
#define SPEED_H 253
#define SPEED_STOP 0
```

この命令は他のプログラムでもたびたび登場しますので、覚えておきましょう。

命 令 [#define x y]

実行結果：プログラム内に「x」とかかれた部分は、実行時「y」に置きかえられる

使 い 方：#define [プログラムにかくときの文字列] [実行時に置きかえたい文字列]

これで、`186` という値をかくかわりに `SPEED_L` とかいても同じ動作をするようになりました。もし回転速度を 50RPM に^{へんこう}変更したければ、いちいち値を求めることなく `SPEED_L` を `SPEED_M` にかきかえるだけですみます。

このプログラムを見ると、3段階の速度が指定されているのが分かりますね。表にまとめましょう。

表 1-1 RPM とスピードの設定

文字列	目標速度 [RPM]
SPEED_H (高速)	70RPM
SPEED_M (中速)	50RPM
SPEED_L (低速)	30RPM
SPEED_STOP (停止)	0PRM

`#define` は今回のように数値が^{ふくざつ}複雑で覚えづらいときに便利です。他にも、同じ数値をプログラム内で何度も使う際にも役立ちます。たとえば 100 という数値を 120 に^{へんこう}変更したいとき、普通なら 100 とかかかれている場所をすべて打ち直さなければなりません。はじめに `#define ●● 100` と^{せんげん}宣言しておけばここを `#define ●● 120` と修正するだけで済むわけです。

他にも、たとえばタッチセンサーを使うときに `HIGH`、`LOW` よりも `ON`、`OFF` とかいたほうが分かりやすいな、と思ったら、`#define ON HIGH` としておくという手もあります。

3) 移動距離^{きょり}を求める

さて、これでエンコーダーロボットの動作プログラムを学びましたが、プログラム中に必要なのはあくまでRPMと `delay` の値、つまり「1分間に何回転するか」と「モーターを何秒間回転させるか」の2つであることがわかりました。

これまでは「30cm 前進させたいけど何秒モーターを回転させればいいんだろう…」というときは、ひとまず適当な時間を指定して動作させてみて、値を直してまた動作させて…というのをくり返していましたね。

しかし、エンコーダーでRPMが指定できるようになった今なら、この作業を行うことなく計算だけで予想できるようになるのです。

つまり、「RPM」をヒントに「モーターを回転させる時間(秒)」を求めたいわけですが、どのように計算式をたてていけばよいのでしょうか。いったんロボットを持っている手をペンに持ちかえ、パソコンではなくテキストに向かって考えてみましょう。

やってみよう!

モーターを `SPEED L` (30RPM) で動作させ、ホイールを3回転半させるのに、モーターは何秒間回転させればよいかを以下の2つの式で求めたよ!

それぞれの式で求めているのは何の値かな? 式に使われた数値を読みとって、言葉にしてみよう!

$$30 \div 60 = 0.5 \quad 0.5 \text{ は何の値かな?}$$

 1秒あたりのホイールの回転数

$$3.5 \div 0.5 = 7 \quad 7 \text{ は何の値かな?}$$

 3回転半させるのに必要な時間 [秒]

ステップアップ

もう少しレベルを上げてみよう！






[SPEED L] (30RPM) でエンコーダーロボットを動かし 300mm 前進させたければ、モーターは何秒間回転させればいいかな？

先ほどの問題も参考にして途中式をかき、それぞれの式で求めているのが何の値か言葉にしつつ計算してみよう！

 ヒント

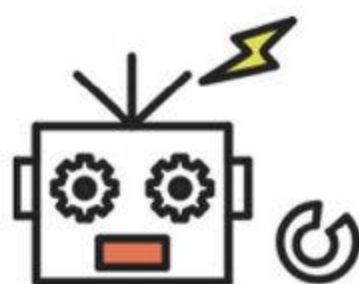
計算には以下の条件を使おう。

- ① オムニホイールの直径は 48mm とする。
- ② 円周率は 3.14 とする。
- ③ わり算をするときにわり切れなかった場合、答えは小数第 3 位を四捨五入し、小数第 2 位までの数とする。

 $48 \times 3.14 = 150.72$	求めたのは	ホイール 1 回転で進む距離 [mm]
 $300 \div 150.72 = 1.99$	求めたのは	300mm 進むのに必要な回転数
 $30 \div 60 = 0.5$	求めたのは	1 秒あたりのホイールの回転数
 $1.99 \div 0.5 = 3.98$	求めたのは	300mm 進むのに必要な時間 [秒]
	求めたのは	

講

計算順序は解答例以外のパターンもありますが、記述部分は概ね「答えの単位がある程度特定できるかどうか」を基準として判定してあげてください。「距離」「時間」「回転数」などといったフレーズが入っていることが望ましいです。



ここまで計算できれば動作のプログラムに使えるだね！

4) ロボットの動き

では、具体的にロボットの動きを考えてみましょう。ここでは300mm角の正方形の軌跡^{きせき}をえがくように、エンコーダーロボットを動かすことを目指しましょう。

図1-6のように右回りに正方形の軌跡^{きせき}をえがく場合は、「直進」と「右に回転」の2つが基本的な動きとなりますね。

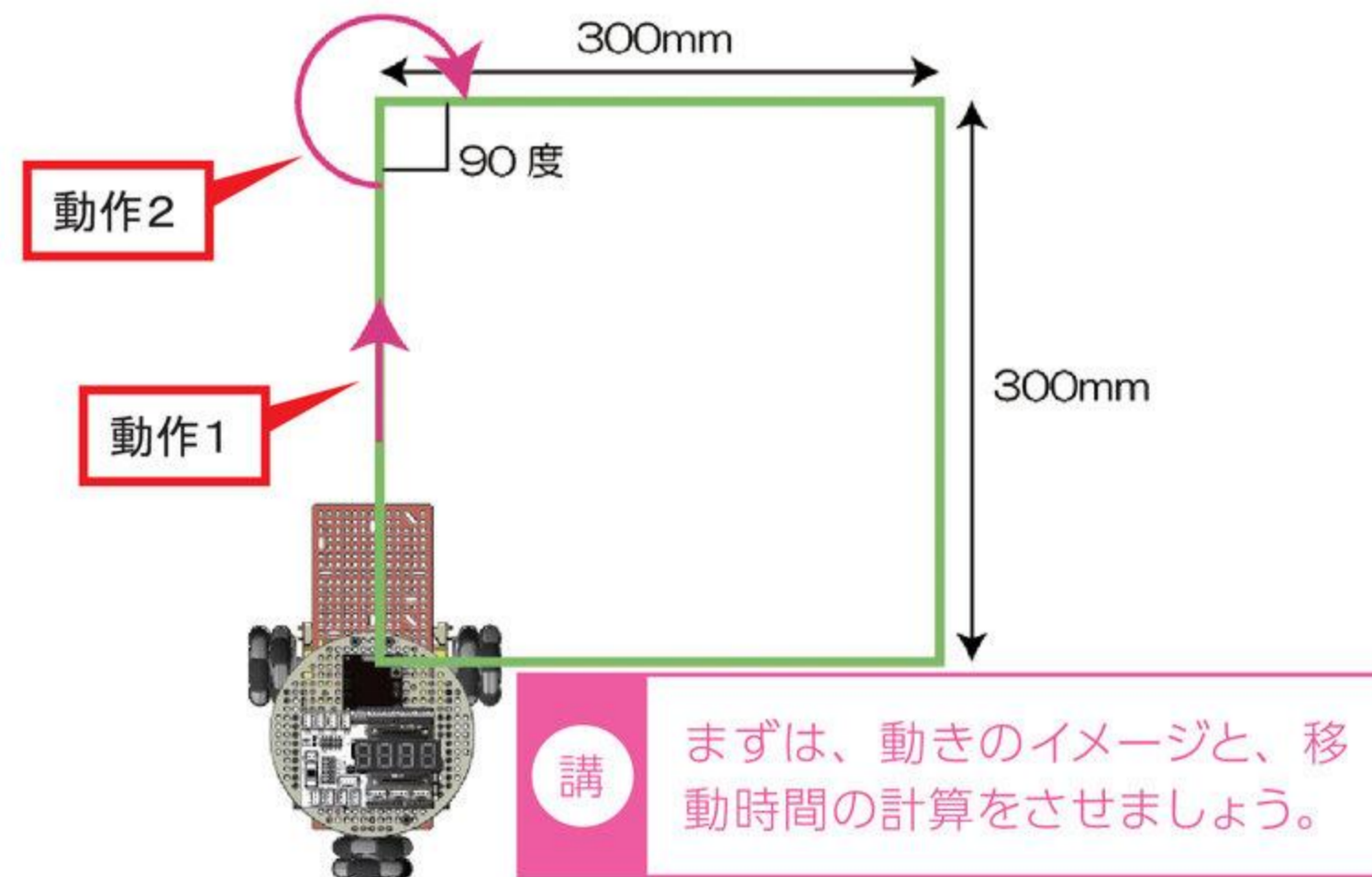


図1-6 エンコーダーロボットを正方形に動かす

2つの動きをより具体的に表現すると以下ようになります。

POINT

直進：300mm 直進
 右に回転：右に90度回転

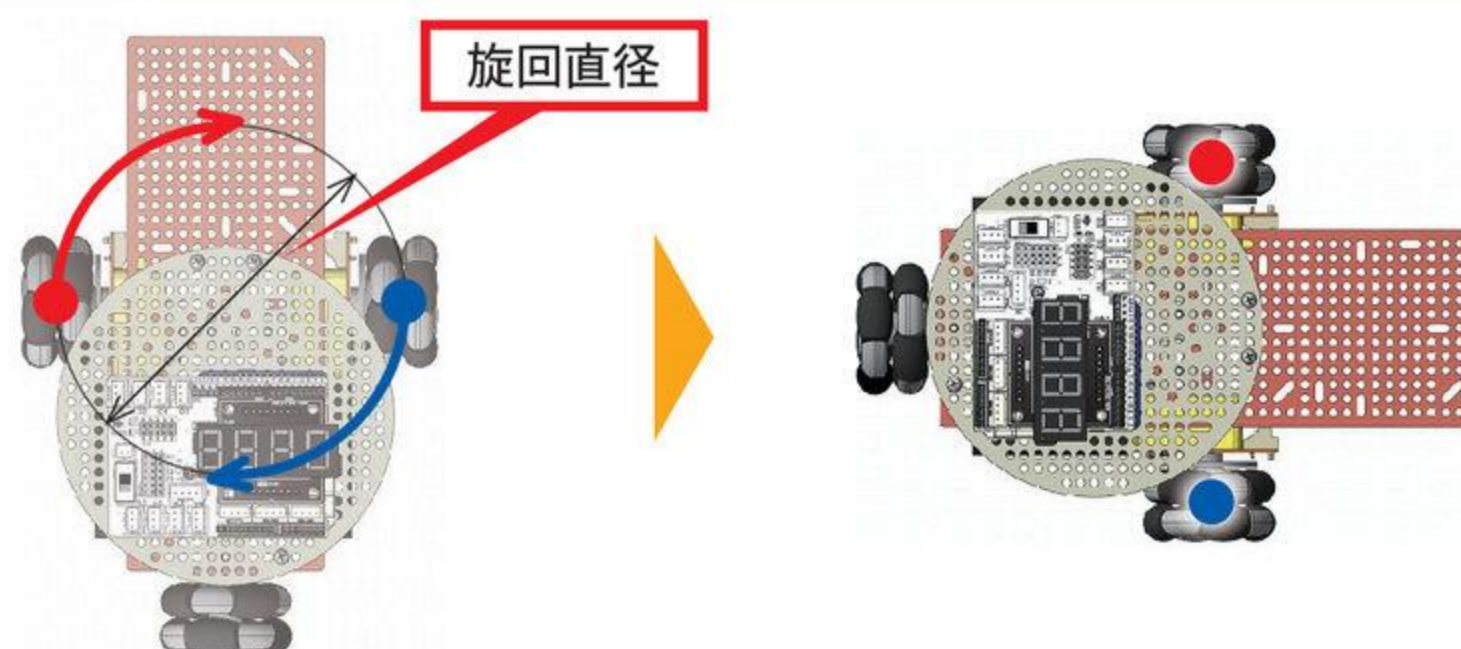
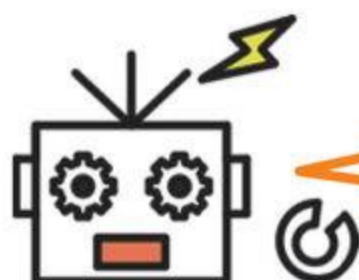


図 1-7 右回転



「300mm 直進」はすでに計算したネ。「右に 90 度回転」は難しそうダ……。

90 度右を向くとき、左右のホイールは図 1-7 の通り、円周の 4 分の 1 の弧をえがくように移動します。この直径をはかれば移動距離を求めることができ、モーターを回転させる時間も分かりますね。

講

ロボットの組み立て精度で若干前後しますが、旋回直径が概ね 110 ～ 120mm 程度になります。実際に走行させる際の床の材質等にも影響を受けるため、正確な距離を測るというよりは大まかな目安を求めるのが目的です。ちなみに解答例プログラムでは旋回直径を 125mm として計算しています。

5) ロボットを動かす

さて、これでエンコーダーロボットに正方形をえがかせるための材料はそろいました。身についた知識を駆使して、正方形の軌跡をえがくエンコーダーロボットを作成しましょう！

ステップアップ

プログラム「encoderMotorTest」をかきかえ、正方形の軌跡をえがくロボットのプログラムを完成させよう！

💡 ヒント

「encoderMotorTest」のメインループは `while (!digitalRead(D2));`、つまり `D2` タッチセンサーが押されたらスタートしているね！ これを利用して、`D2` タッチセンサーが押されたらロボットが動きはじめ、正方形の軌跡を 1 周えがいたら停止するというプログラムにできるかな？

講

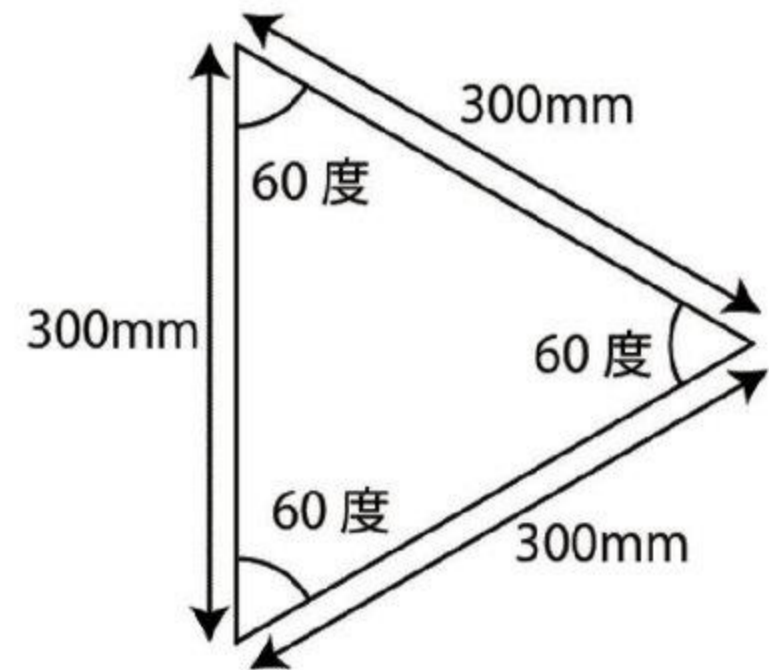
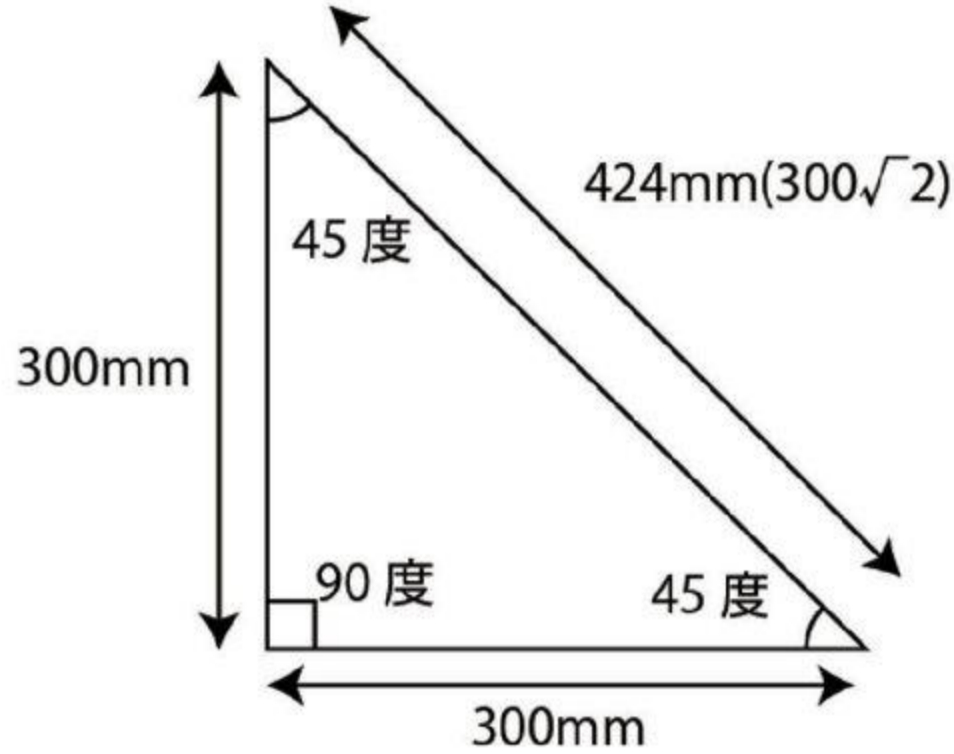
解答例は以下のプログラムです。
RoboticsProfessorCourse2 > CombRobot6 > Square

ヒントの通り `while (!digitalRead(D2));` を使用する場合、1 回のループ内で正方形 1 周を終える必要があります。直進→回転を 4 セットくり返せばちょうど 1 周分になるため、解答例プログラムは for 文を用いて軽量化を図っています。こういった工夫がひとりで行えるようになると、思考力がぐっと伸びます。必要に応じてアドバイス等してあげてください。

チャレンジ課題

難易度中級

四角をえがくプログラム「Square」を^{へんこう}変更して、直角三角形や正三角形の軌跡^{きせき}をえがくロボットにしてみよう。計算のもとになる値は図を参考にしてね。



講

解答プログラムはそれぞれ以下となります。

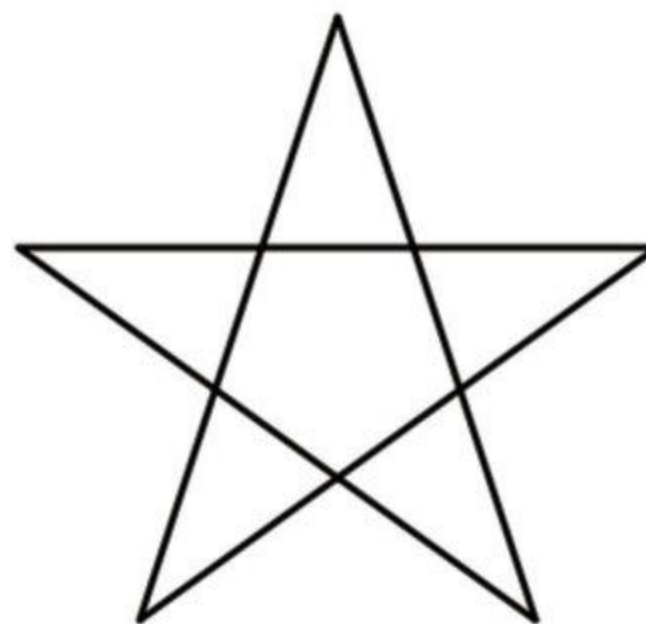
直角三角形：RoboticsProfessorCourse2 > CombRobot6 > Triangle1

正三角形：RoboticsProfessorCourse2 > CombRobot6 > Triangle2

チャレンジ課題

難易度上級

四角をえがくプログラム「Square」を^{へんこう}変更して、星型の軌跡^{きせき}をえがくロボットにしてみよう。各辺は 300mm、頂点の角度は 36 度として考えよう。



講

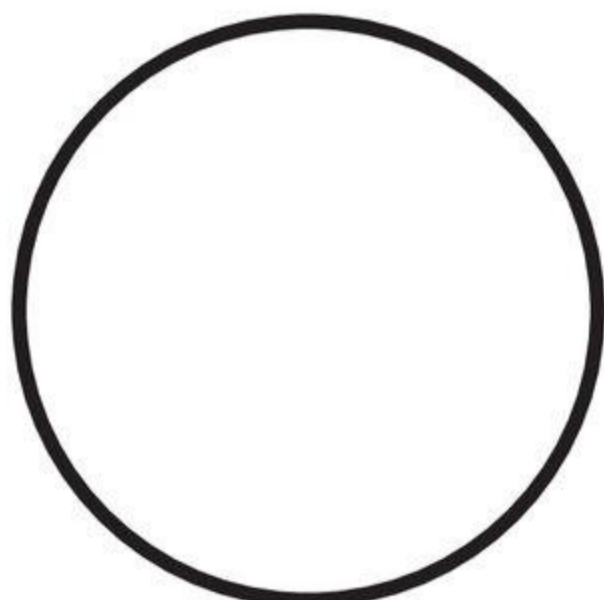
解答プログラムは以下となります。

星型：RoboticsProfessorCourse2 > CombRobot6 > Pentagon

チャレンジ課題

難易度上級

円の軌跡^{きせき}をえがくロボットにしてみよう。



講

解答プログラムは以下となります。

円 : [RoboticsProfessorCourse2](#) > [CombRobot6](#) > [Circle](#)

なお、円をえがくプログラムの移動時間は、今回はトライアル&エラーで設定するようにしてください。サンプルプログラムでは、右モーターを「SPEED_L」左モーターを「SPEED_M」として、ロボットが弧をえがくようにプログラムしています。

2. 赤外線ライトレーサー (目安 40 分)

2.0. 赤外線ライトレーサーへ改造

続いて、赤外線ライトレーサーに改造します。エンコーダー基板の片側もしくはすべてを取り外します。エンコーダー基板から取り外したオムニホイールは再びモーターに取り付け直します。取り外したエンコーダー基板は図 2-0 のように前方に取り付けます。こうすると、エンコーダーの機能でライトレースができるようになります。

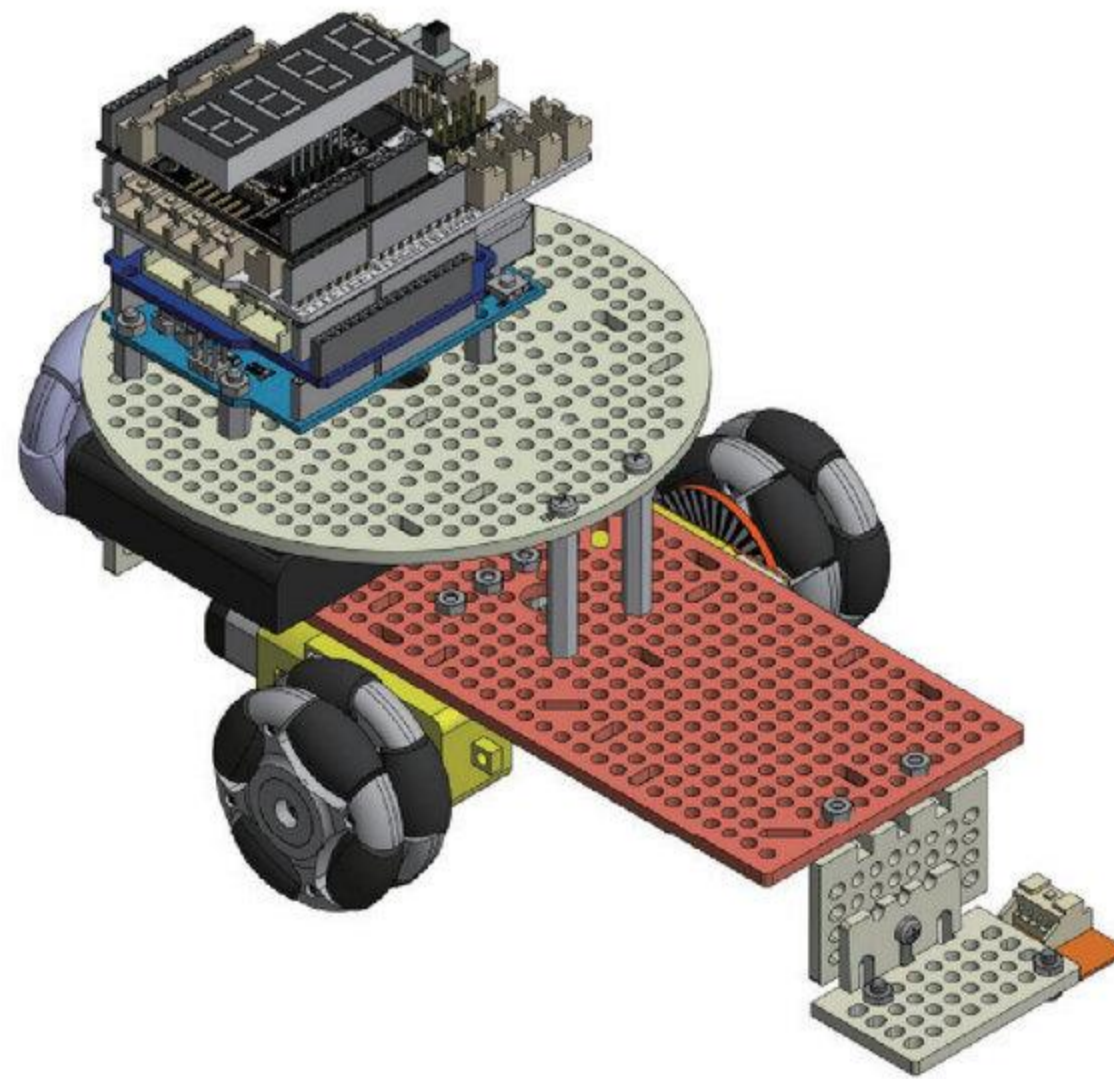


図 2-0 赤外線ライトレーサー

図 2-1 を参考に組み立てましょう。なお今回もパーツやネジ穴の位置などはテキストと多少異なっても大丈夫です。なおテキストの場合は、使用しているパーツは、M3L8 ネジ (× 5)、M3 ナット (× 5)、センサー L 字ステイ (× 2) です。

エンコーダーの位置は、地面からの距離の調整が肝心です。エンコーダーは、姿勢検出シールドの **ENC1** にセンサーケーブルを使って接続します。

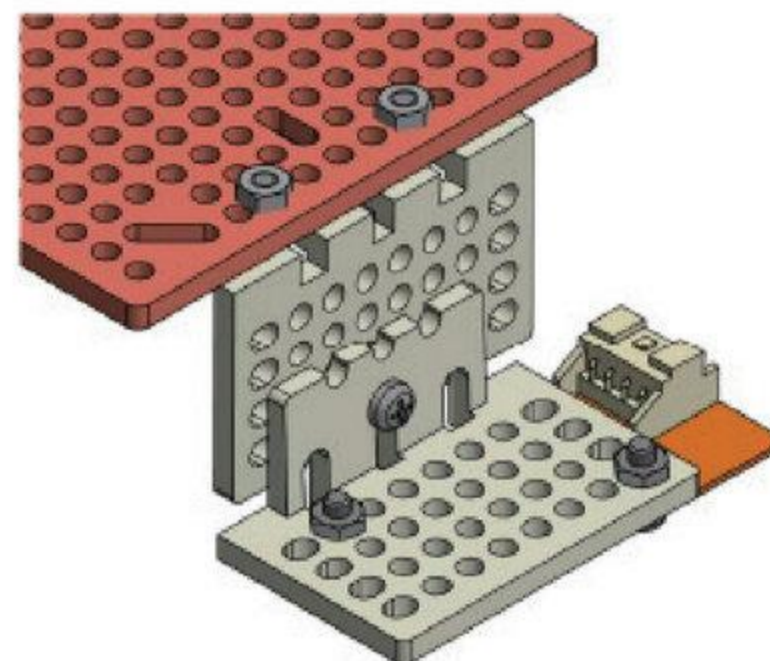


図 2-1 エンコーダー基板の取り付け例

2.1. 赤外線ラインセンサーの調整

センサーとラインとの高さを調整し、黒いラインに反応するようにします。
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 > CombRobot6 > LineSensorTest

実行結果：センサーが黒に反応すれば、7セグメントLEDに「YES」と表示され、反応しない場合は、「no」が表示される。

うまくいかない場合は、**図2-2**を参考にネジを使って高さの調整をしてください。また、赤外線ラインセンサーは、周りの環境（明るさなど）にも影響えいぎょうされますので、場所を変えるたびに調整を行ってください。

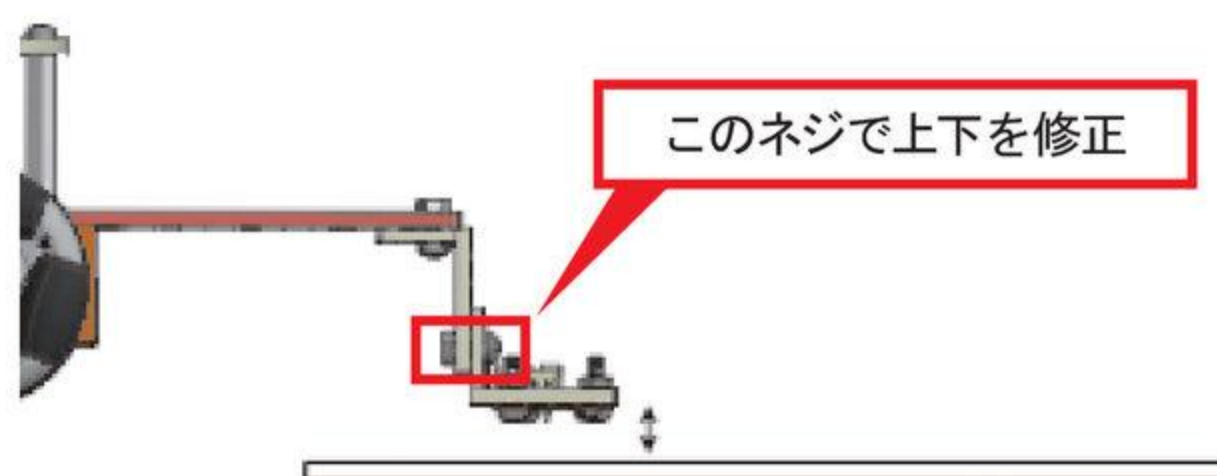


図2-2 赤外線ラインセンサーの調整

ちなみに、アルゴリズムはいたって簡単です。一番シンプルなルール（ロボットがラインの左端ひだりはしに沿って進む場合）は以下ようになります。

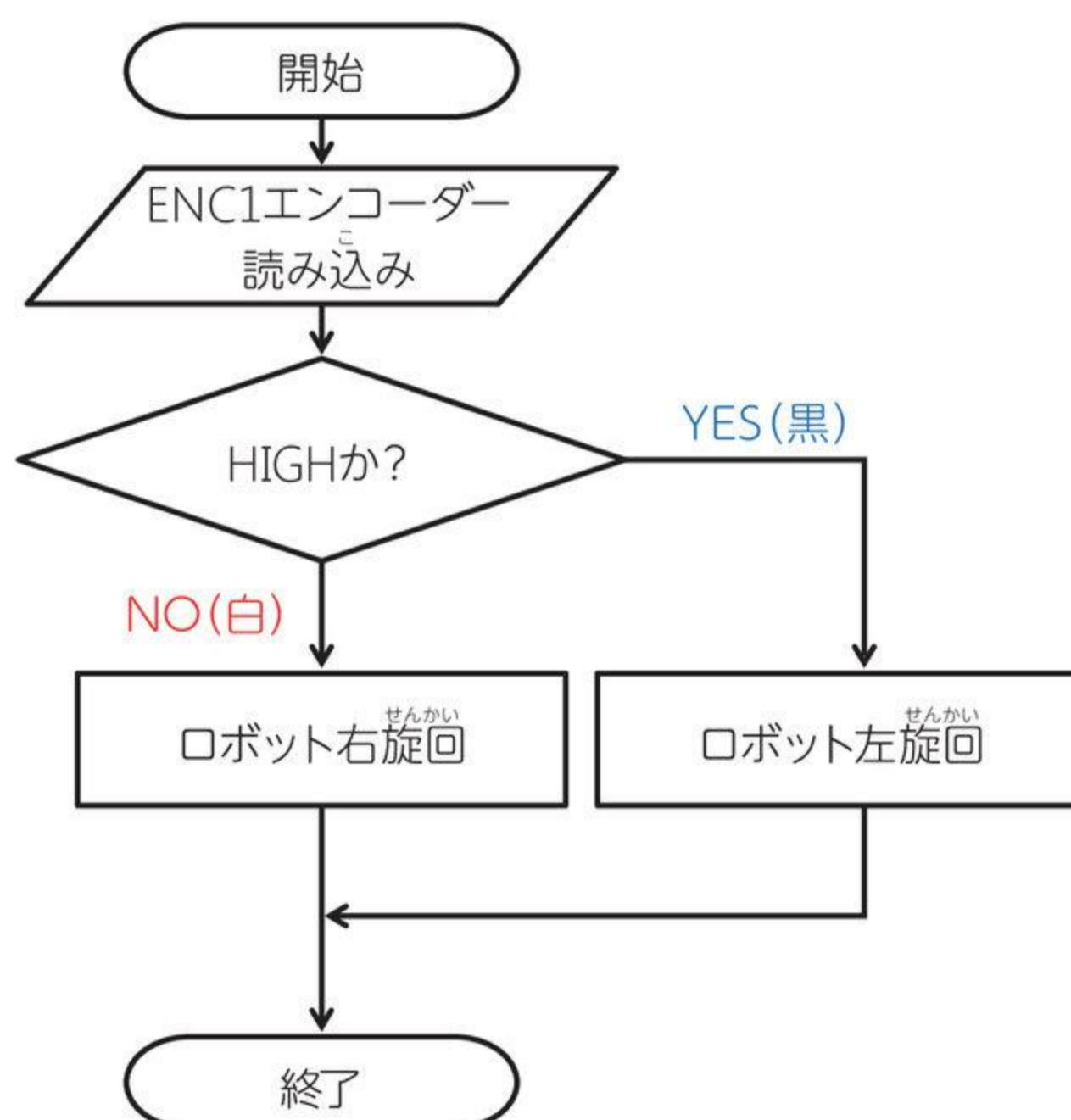


図2-3 ライントレースのフローチャート

それでは、ラインレースをしましょう！
次のページのコースを使ってください。

ステップアップ

前のページのフローチャートを参考に、エンコーダー基板を用いた赤外線ライントレーサーのプログラムを作成してみよう！

今までこのテーマで登場したプログラムの中で、どれをベースにしていけばいいかも考えてみよう。

余裕のある人は、先ほど調整で使った「LineSensorTest」の7セグメントLED表示機能も追加してみよう！

解答例は以下のプログラムです。

RoboticsProfessorCourse2 > CombRobot6 > LineTracer

【プログラムについて】

解答例は、7セグメントLEDの動作も組み込んだプログラムになっています。

自作する場合検知結果が `HIGH` か `LOW` かで動作を分けていた第5回プログラム「switchCount1」、エンコーダーを用いてモーターを制御していた「Square」などが、ベースとして使いやすいです。

自分で構文をかいていくのが難しいようであれば、それぞれの必要部分を切り貼りしていくイメージで作業させてもよいでしょう。なお、解答例プログラムは「switchCount1」等と違いエンコーダーに `ENC1_U1` 等と名付けていますが、従来通り `ENC1_A` 等と名付けていても何ら問題はありません。

【改造のポイントとコースについて】

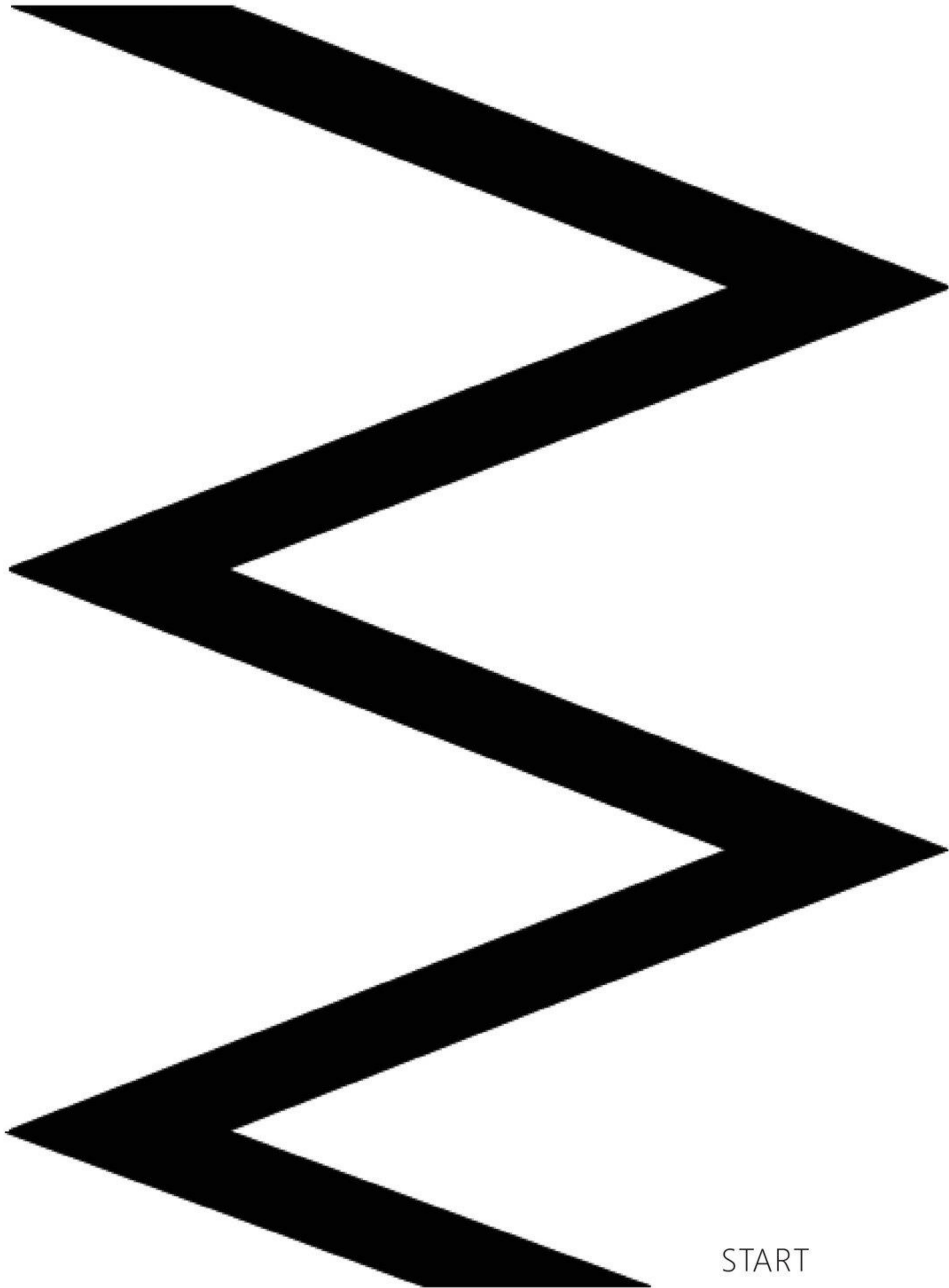
「センサーの取り付け位置を、ロボットの回転中心から遠くしたり近くしたりする」「モーターの回転スピードを、速くしたり遅くしたりする」ことなどによって性能も変わりますので、試行錯誤させてください。

テキストの巻末にも、ラインレースのコース見本がありますが、小さい場合は、A3サイズに拡大コピーしてください。

また、床に黒いビニールテープなどを貼ってもオリジナルのコースが作れます。

講

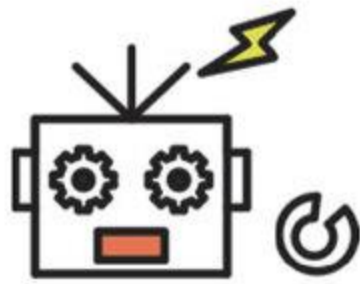
【テストコース】 ※ A4 で小さい場合は、A3 に拡大コピーしてください。



3. まとめ（目安5分）

これで、センサーロボットの回は終わりになります。プログラムについて、より理解が深まったでしょうか？ ここまでくれば、自分でイチからプログラムをかける人もでてきているかもしれません。ロボットはハードウェアを如何にシンプルに作って、如何にそれを有効に使うって複雑なことをさせるか、そこに性能がかかってきます。そして複雑なプログラムをかけることが大切なわけではなく、むしろシンプルなプログラムの方が素晴らしいのです。

自分の思った通りのことを、シンプルにロボットに組み込むということを考えてプログラムをかいていきましょう！



次回からは、「姿勢検出シールド」について勉強スルヨ！

講

- 以下の授業の目標を再確認します。
 - ・エンコーダーをロボットに組み込む
 - ・いろいろな軌跡で動かす
 - ・エンコーダーを使ったライトレーサーにチャレンジ
- 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「倒立振り子ロボット」、3年目コースに進級する場合は「二足歩行ロボット」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。




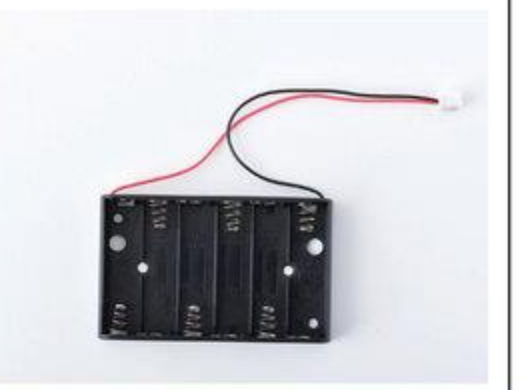





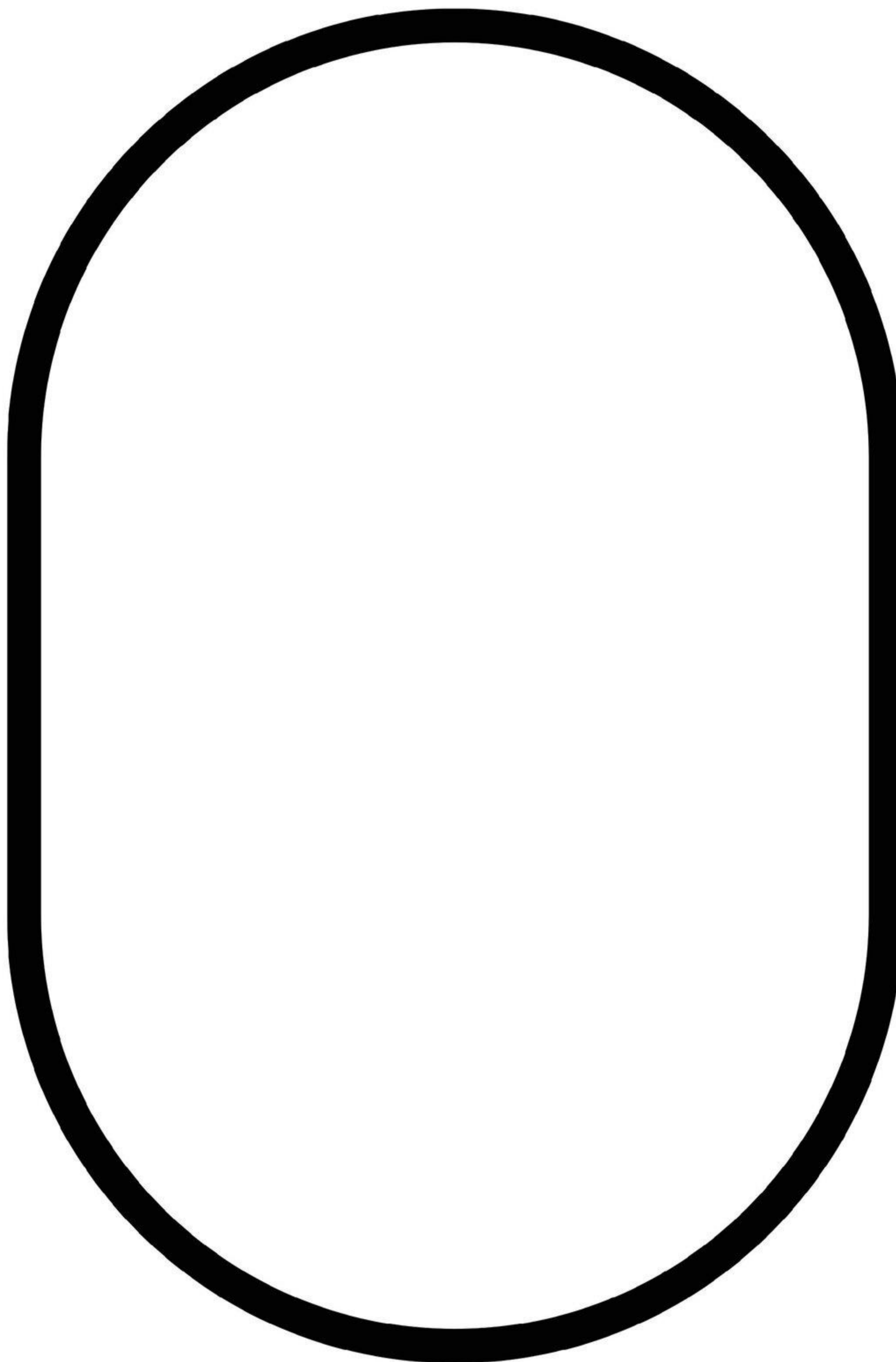
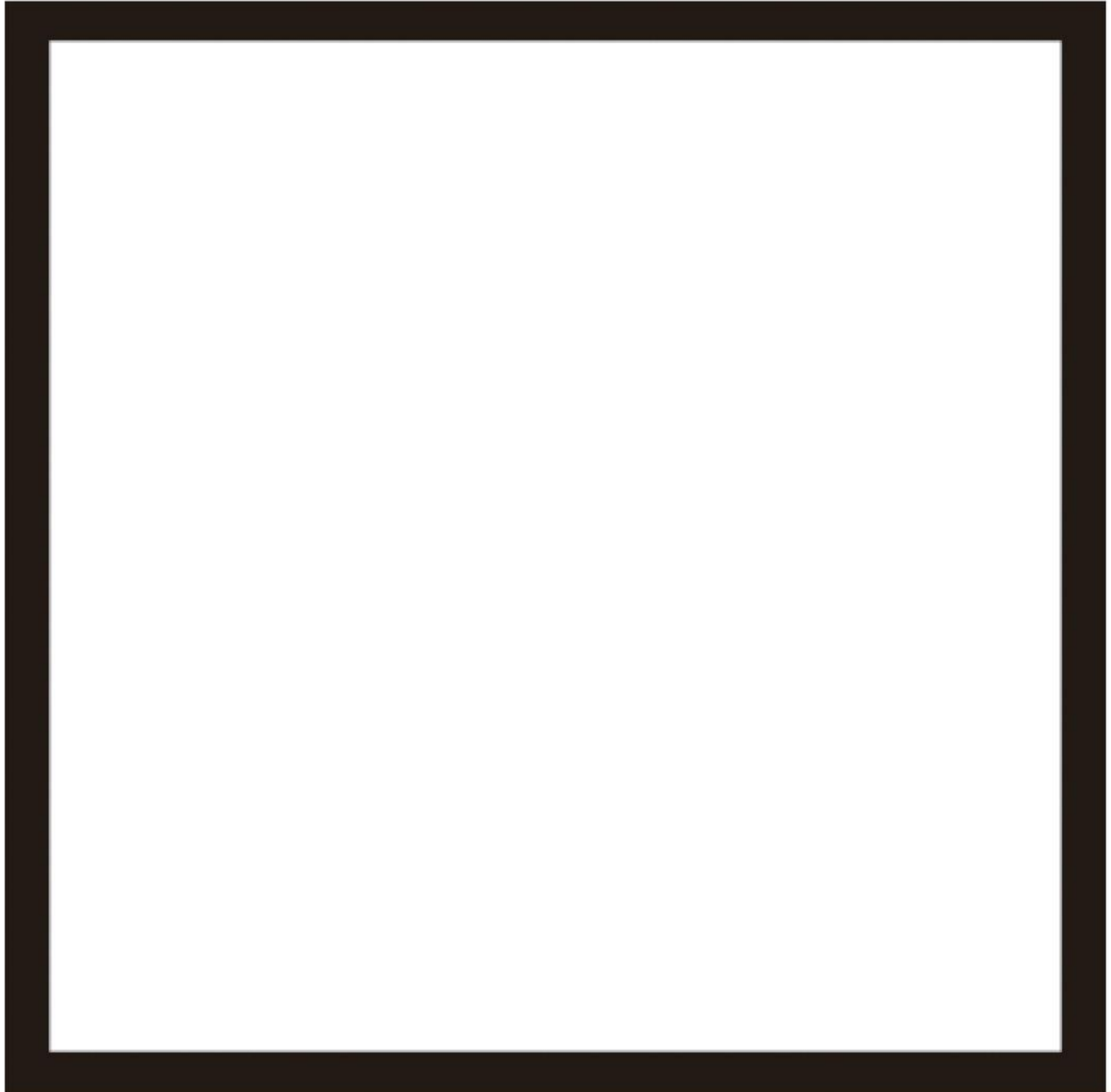
USB ケーブル 1	マイコンボード 1	ロボプロシールド 1	電池ボックス 1
			
マトリクスLEDシールド 1	マトリクスLED 1	スピーカー 1	7セグメントLED 1
			
姿勢検出シールド 1			
			

図 3-0 次回必要なもの

※ A4 サイズで小さい場合は、A3 サイズに拡大コピーしてください。



※ A4サイズで小さい場合は、A3サイズに拡大コピーしてください。



※ A4 サイズで小さい場合は、A3 サイズに拡大コピーしてください。

