

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

とう りつ しん し
倒立振子ロボット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3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

とうりつしんし

倒立振子ロボット③

第5回

とうりつしんし

倒立振子ロボットの製作

講師用

目 次

0. 倒立振子ロボットの製作

- 0.0. 「倒立振子ロボットの製作」でやること
- 0.1. 倒立振子ロボットとは
- 0.2. 必要なもの

1. 倒立振子ロボットの仕組み

- 1.0. 倒立と振り子
- 1.1. 倒立振り子とは
- 1.2. 姿勢を保つ仕組み

2. 倒立振子ロボットの組み立て

- 2.0. 上段部とモーターの組み立て
- 2.1. ベース部の組み立て
- 2.2. 中段部の組み立て
- 2.3. ベース部と中段部の接続
- 2.4. 回路と上段部の取り付け
- 2.5. 動作確認

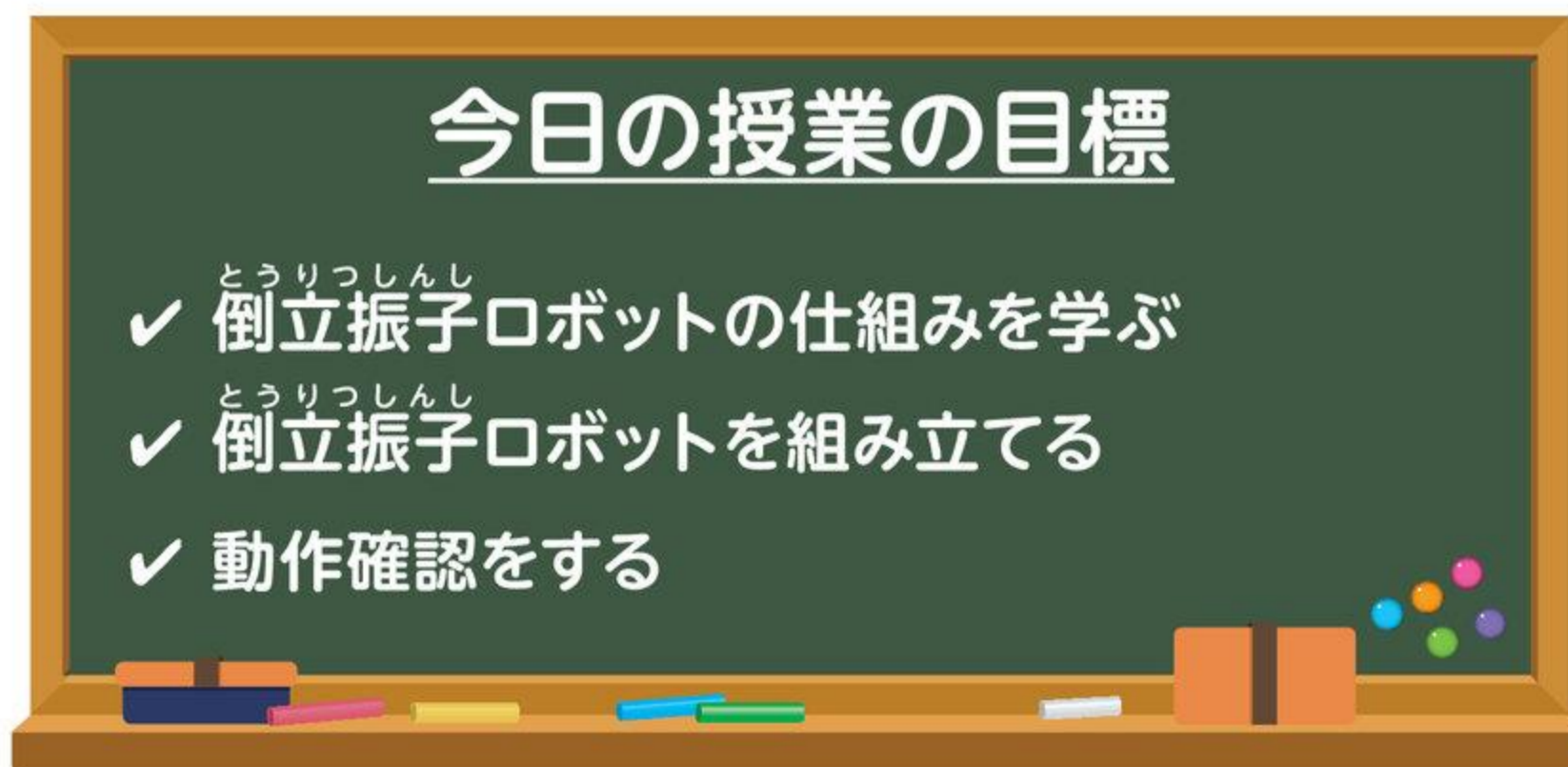
3. まとめ

- 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. 倒立振り子ロボットの製作（目安10分）

0.0. 「倒立振り子ロボットの製作」でやること

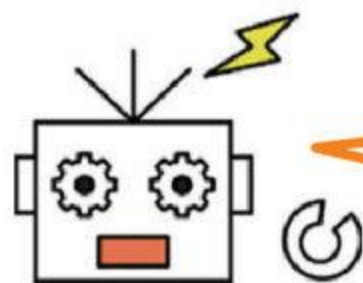


これまで、姿勢検出シールドのそれぞれのセンサーの機能と使い方や、「フィードバック」という制御方法について学びました。第5・6回では、いよいよこれまで学んだ知識をもとに「倒立振り子ロボット」を作って、動かしていきましょう！

「倒立振り子ロボット」については、この後詳しく説明しますが、文字通り「振り子を逆立ちさせたもの」をイメージするとわかりやすいでしょう。

振り子は逆立ちさせると不安定になりますよね。そのため、「逆立ち状態をより安定させるにはどうしたらよいのか？」を考えるのがポイントになります。

姿勢検出シールドの機能をおさらいしながら、ロボットをしっかりと倒立させられるようにチャレンジしていきましょう。



それぞれのセンサーの機能をフルに使うぜー！！

0.1. 倒立振り子ロボットとは

図 0-0 が、今回作る「倒立振り子ロボット」の完成形です。

「倒立振り子ロボット」とは、ロボット自体の姿勢を検出してバランスを保つロボットのことで、バランスを保つために必要となるのが、第4回までに使用した姿勢検出シールドの各センサーです。普通に考えれば、2個のタイヤではロボットは転んでしまいますよね？ しかしこのロボットは、姿勢検出シールドの各センサーの機能を活用することで、バランスを保って倒立するわけです。

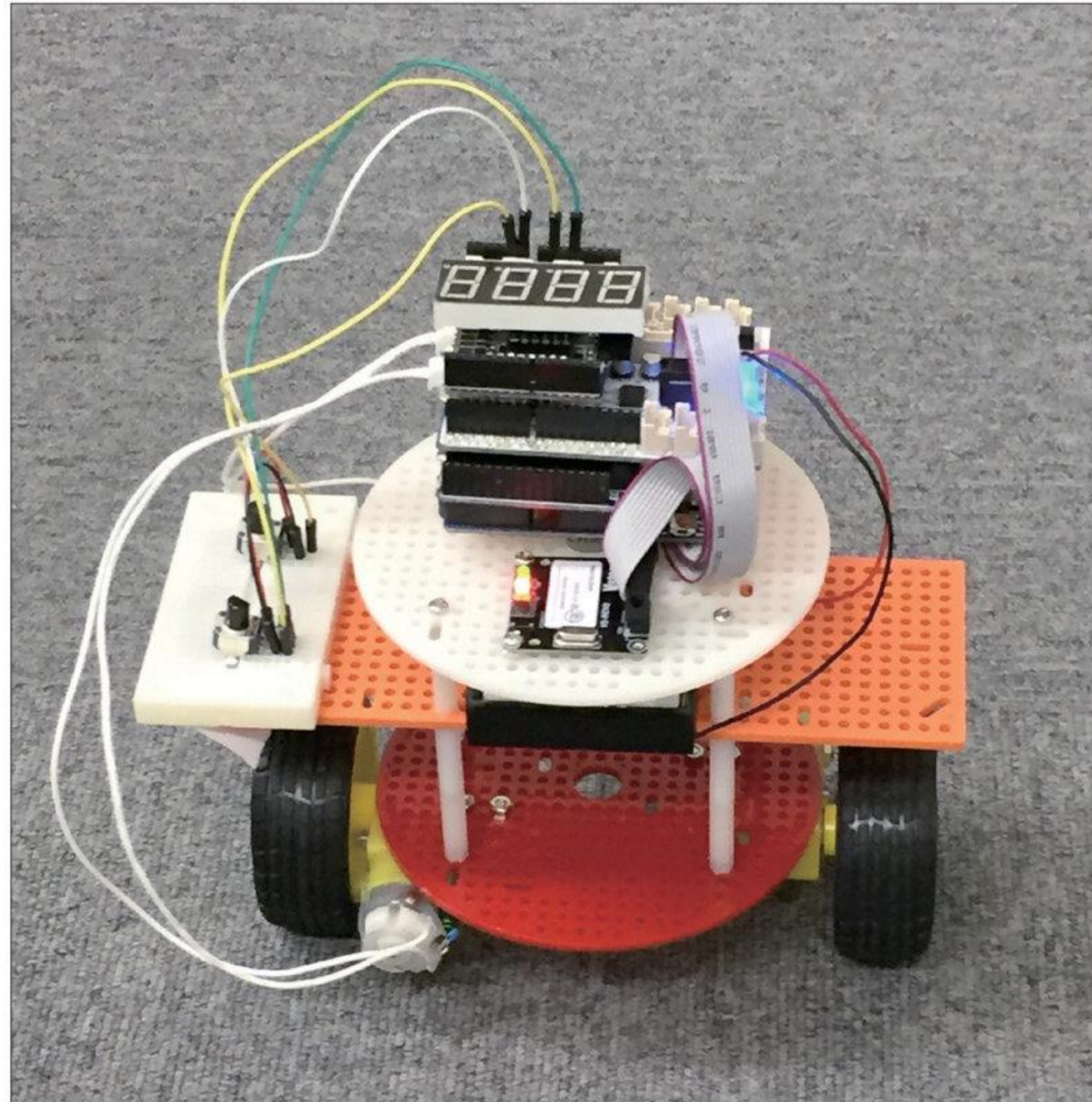


図 0-0 倒立振り子ロボット

0.2. 必要なもの

前回使ったロボットと、以下のパーツを準備しておきましょう。また、前回使ったロボットにタッチセンサーがついている人は、取り外しておきましょう。

ラジオペンチ 1	ドライバー 1	USB ケーブル 1	リボンケーブル 1
			
無線受信モジュール 1	ユニバーサルボード 1	301 ブレッドボード 1	ジャンパー線 65
			
可変抵抗ボリューム 2	タイヤ 2	M3L8 ネジ 5	25mm 角スペーサー 8
			

図 0-1 必要なもの

1. 倒立振り子ロボットの仕組み (目安 35 分)

1.0. 倒立と振り子

ここでは、倒立振り子ロボットの仕組みを学びます。まずは「倒立」と「振り子」について考えてみましょう。

1) 倒立

「倒立」とは「手を地面に置いて足を上にして、通常の直立状態とは上下反対に立つ行為」のことです。簡単にいうと、逆さまに立つということです。

そもそも人間が何気なくやっている「立つ」という動作も実は難しく、われわれが無意識のうちにバランスを保っているからこそできているのです。

皆さんも将来、二足歩行のロボットを作ることがあれば、「立つ」という単純な動作をさせることの難しさを実感するはずです。そして、これが自然界の中での一般的な感覚です。自然界で、二本足で立っている動物は非常に少ないのです。

そう考えるとロボットに倒立させることの難しさがわかりますね。

2) 振り子

「振り子」とは、支点から吊るされ、重力の作用により、揺れを繰り返す物体のことです。重力によって下に引かれると加速し、一番下の位置で最高速になります。そして、反対側に揺れるときは、減速した後一旦停止します。また、支点での摩擦や空気抵抗の無い理想の環境では永久に揺れ続けます。

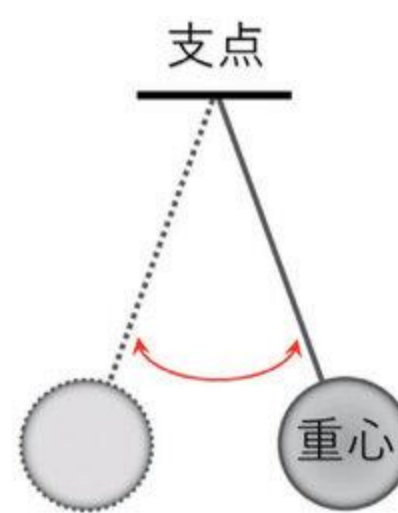


図 1-0 振り子



豆知識

振り子の等時性

振り子には、「ひもの長さが同じなら、振り子の揺れが大きくても小さくても、往復にかかる時間は同じ」という性質があります。これを「振り子の等時性」といいます。

1.1. ^{とうりつしんし}倒立振り子とは

1) ^{とうりつしんし}倒立振り子のイメージ

さて、「倒立」^{とうりつ}と「振り子」^{ふりこ}について、確認を終えたところで、「倒立振り子」^{とうりつしんし}とは何かを考えていきましょう。

ここでは、「振り子時計」をイメージしてみましよう。「倒立振り子」^{とうりつしんし}とは時計に入っている振り子^{ふりこ}がひっくり返って立ったものだと考えてみてください。普通の振り子^{ふりこ}は支点が重心の上の方にあるのに対して、倒立振り子^{とうりつしんし}は下の方にあります。これでは、すぐに倒れそうですね。

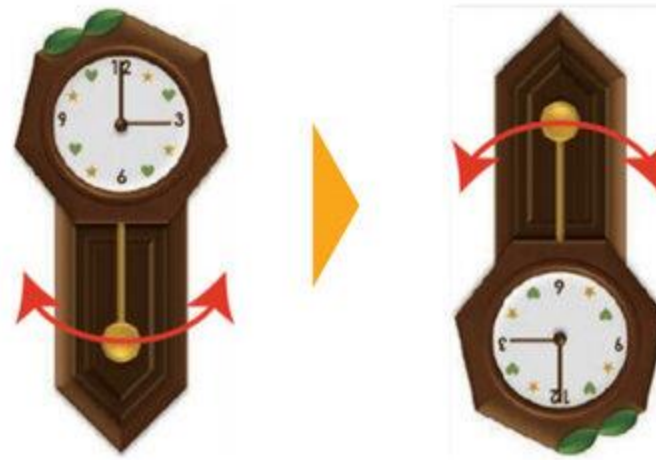


図 1-1 逆さ振り子^{ふりこ}（倒立振り子^{とうりつしんし}）

2) ^{たお}倒れないためのポイント


では、倒れない^{たお}ようにするためのポイントを考えてみましょう。

やってみよう！

ほうきのような長いもの、鉛筆のような短いもの、の2つを用意して手の平のうえで立たせてみよう。どちらが安定しやすいかな？ 試^{ため}してみて、結果とその理由を書いてみよう。なお、長いものは1mくらい、短いものは10cmくらいのものであればなんでもOKだよ。



 結果：長いものの方が立てやすい。

 理由：（例）長いものの方が、倒れるまでに時間がかかるため、その分姿勢を立て直しやすいから。

講

実験に使用するものは、身近にあるもので結構です。落下や衝突に注意して進めてください。パソコンなどの電子機器から離れた場所で行ってください。

多くの人は、長いものの方が立てやすいと感じたはずですが、なぜでしょうか？
結論からいえば、短いものよりも長いものの方が、「倒れるまでに時間がかかるから」です。
時間がかかると、その分姿勢を立て直すチャンスが多くもらえるわけです。
また、長いものの方が「反応速度が遅いから」でもあります。手をちょっと動かしても、長いものの方は、そんなに大きく反応しません。一方、短いものは飛んでいってしまいそうなくらい反応してしまいます。反応速度が遅いものは、少しいい加減な動きをしてもうまく立てることができるということです。
つまり、こちらが正確に素早く反応できれば、たとえ短いものでも、しっかりと立ちますが、それはなかなか難しいので、長いものの方が立てやすいのです。
少し難しく表現すれば、「重心位置」と「重さによる慣性力」によって差が出るといえます。
重心位置とは、物体の釣り合う位置のことです。



図 1-2 重心位置の違い

やや難しく感じるかもしれませんが、ここでは以下のポイントをおさえておけば大丈夫です。

**POINT**

支点から重心位置が離れていて、かつ重いものほど立てやすい。

1.2. 姿勢を保つ仕組み

1) 慣性

では今度は、姿勢を保つための仕組みを学びましょう。具体的にどんな仕組みにすればいいのかわかれば、プログラムによって実現できます。

実は、倒れないためのポイントでもある「慣性」が関係します。

まずは、長いものを立てているときの動作を思い出しましょう。どんな動きで倒れないようにしましたか？ ここで思い出してほしいのは、「慣性」です。「慣性」とは、「物に外から力がはたらかないとき、止まっている物は止まったままになり、動いている物はそのまま動き続ける」などといった性質でしたね。

では身のまわりの例で「慣性」について考えてみましょう。

やってみよう！

文章を読んで、カッコ内の正しい方を選び、○で囲もう。

1. 自動車の場合

(1) 走っている車が急ブレーキをかけたなら、乗車している人は、

(前のめりになる ・ 後ろに引っ張られる)。

(2) 走っている車が急発進をしたなら、乗車している人は、

(前のめりになる ・ 後ろに引っ張られる)。

2. ほうき立ての場合

(1) 手の平に置いたほうきが、前の方に倒れそうになったら、

(手を前に出す ・ 手を後ろ側に引く)。

(2) 手の平に置いたほうきが、後ろの方に倒れそうになったら、

(手を前に出す ・ 手を後ろ側に引く)。

このように普段あたり前のように経験していることでも、「慣性」と関係しているのです。

2) 倒立振り子への応用

では、そんな「慣性」をどう「倒立振り子」で応用しているのでしょうか？ 手の平にのつたものを安定させて倒立させるには、手の平（支点）の位置を動かす必要がありますね。そこで、考えられるのが、**図1-3**のように重心の倒れ具合や方向によって、支点の位置を移動させる台車のような仕組みです。

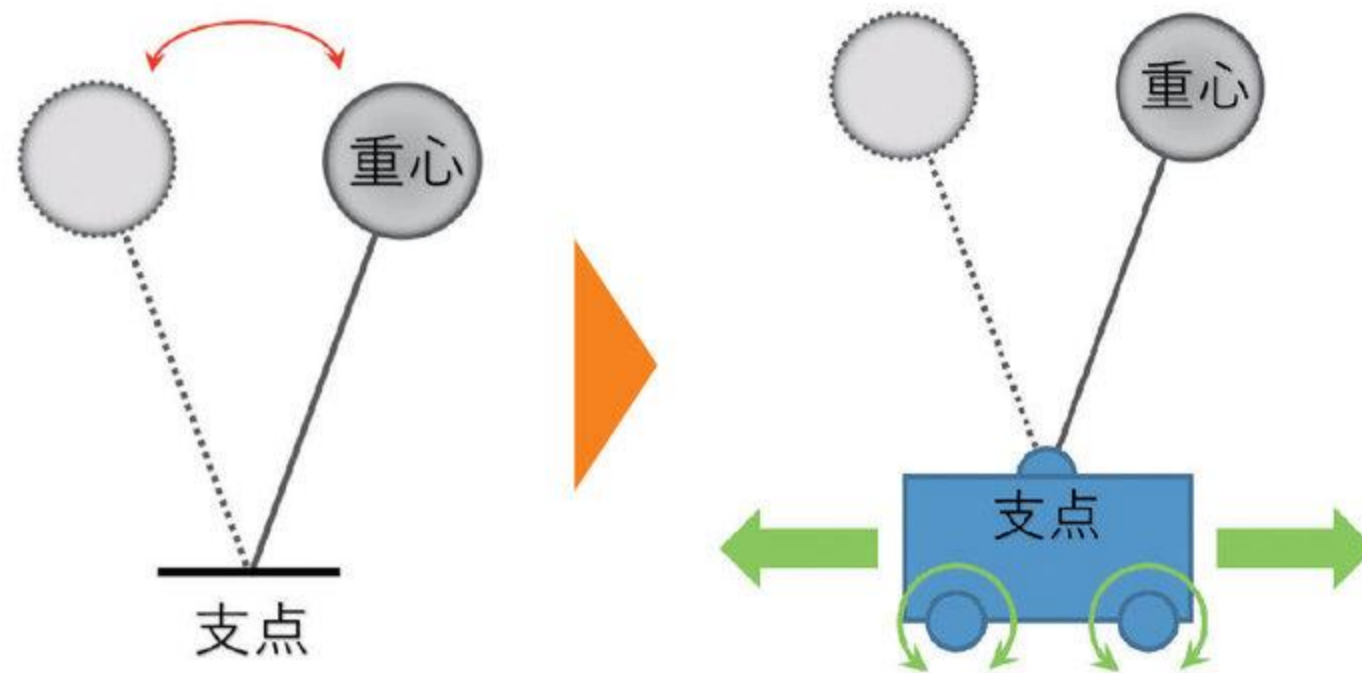


図1-3 倒立振り子

この方法をロボットに適用すると、**図1-4**のようなイメージになります。ちなみに、「倒立振り子ロボット」は、**図1-3**の台車より不安定な二輪です。横から見ると、一輪車をイメージできますね。

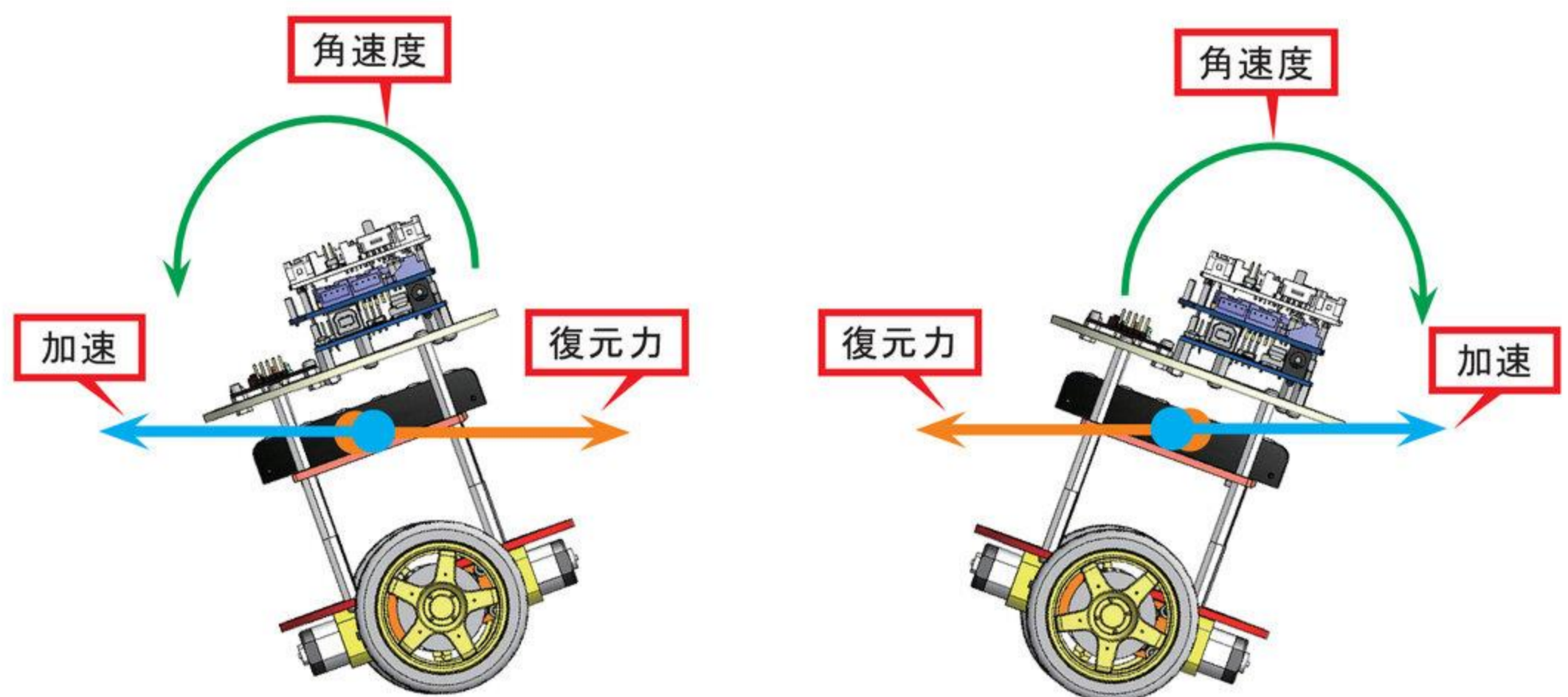


図1-4 倒立振り子ロボットのメカニズム

やってみよう！

文章を読んで、カッコ内の正しい方を選び、○で囲もう。

1. ロボットが前向きに^{たお}倒れそうになったら、

(前向きに加速する ・ 後ろ向きに加速する)。

2. ロボットが後ろ向きに^{たお}倒れそうになったら、

(前向きに加速する ・ 後ろ向きに加速する)。

だんだんイメージができてきたでしょうか？ 逆さになった振り子の運動を制御することによって、姿勢を保ち、倒立させることが今回のロボットのミッションです。

また、人間が感覚的にやっていることをロボットにやらせる場合、必ず使われるのがセンサーです。まず、今回はロボットの姿勢が「今現在どうなっているのか？」をセンサーで検出して、ロボットが倒れそうになったら、倒立姿勢に戻すように目標位置を入力します。これは、すでに勉強した、「フィードバック制御」を利用します。

2. 倒立振り子ロボットの組み立て (目安 60 分)

2.0. 上段部とモーターの組み立て

倒立振り子ロボットのさらなる制御^{せいぎよ}に関しては、次回くわしく勉強するので、今回は組み立てを中心に行います。

<組み立て手順①>

姿勢検出ユニットの付いた白円形ボード^{しせいけんしゅつ}を前回使ったロボットから取り外し、無線受信モジュールを取り付けます。M3L8 ネジ (× 2) と M3 ナット (× 4) を使います。図 2-0 の赤●の部分が穴位置です。必要なパーツはオムニホイールロボットを分解して使います。ちなみに3つあるギアドモーターのうち1つは今回使用しないため完全に分解して大丈夫です。

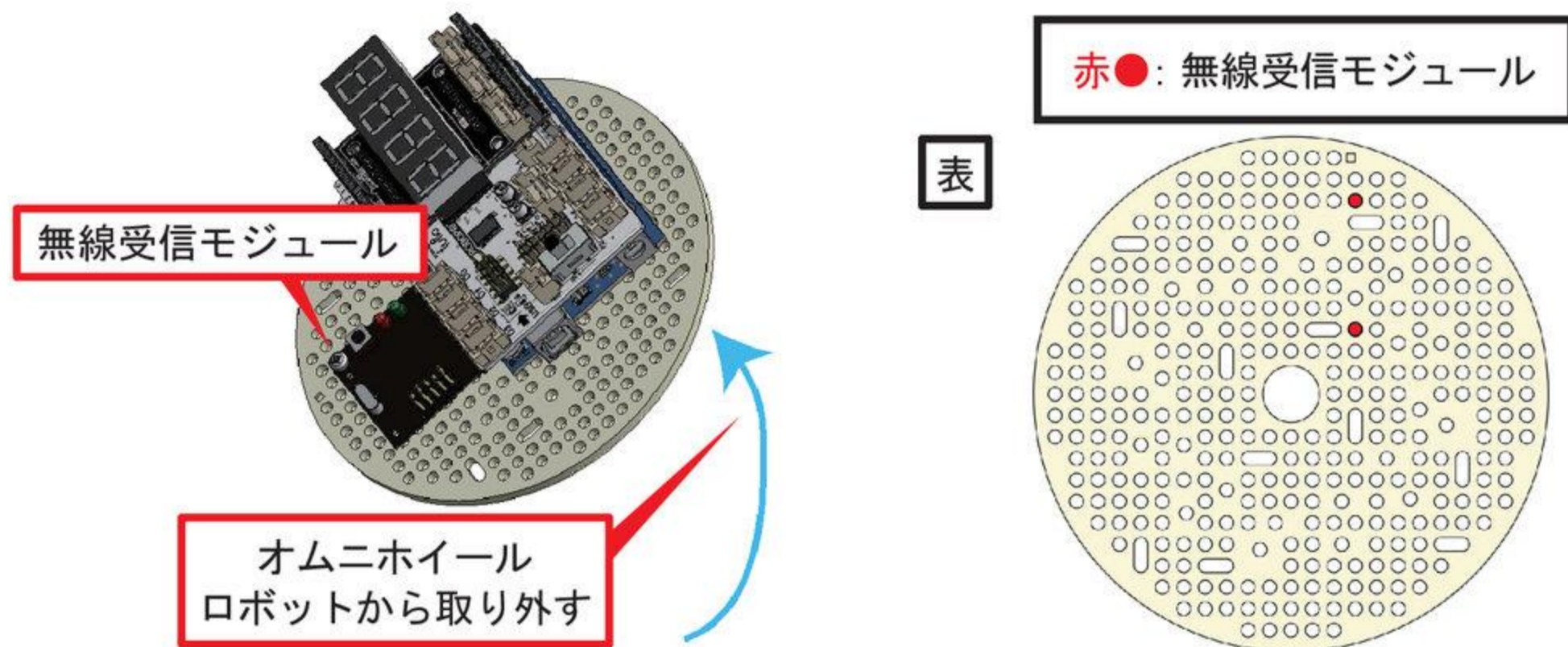


図 2-0 上段部の組み立て

<組み立て手順②>

モーターL字ステイ付きのギアドモーター (× 2) を赤円形ボードから取り外します。次に各モーターのオムニホイールを取り外し、かわりにモーター軸^{じく}にタイヤ (× 2) を取り付けます。なお、タイヤはネジで固定する必要はありません。

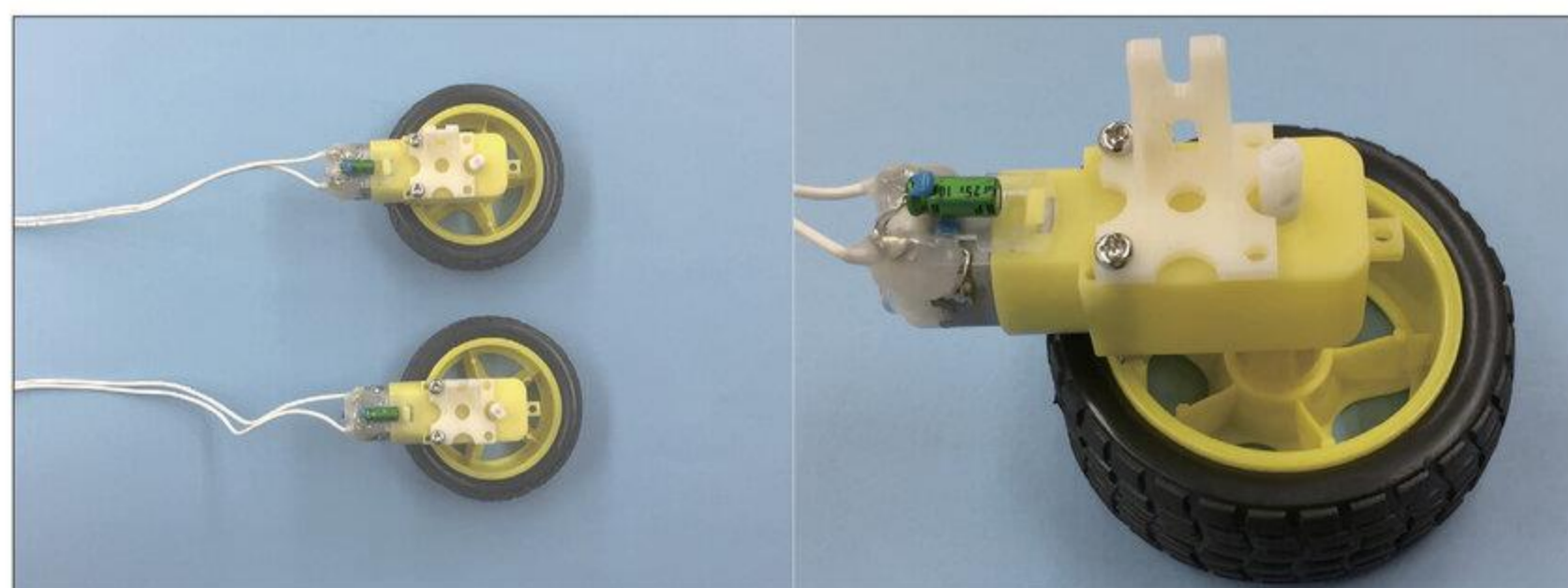


図 2-1 タイヤの取り付け

2.1. ベース部の組み立て

<組み立て手順①>

モーターを取り付けるベース部を組み立てます。25mm 角スペーサーを2段重ねたものを4セット作り、赤円形ボードに M3L8 ネジ (×4) で取り付けます。

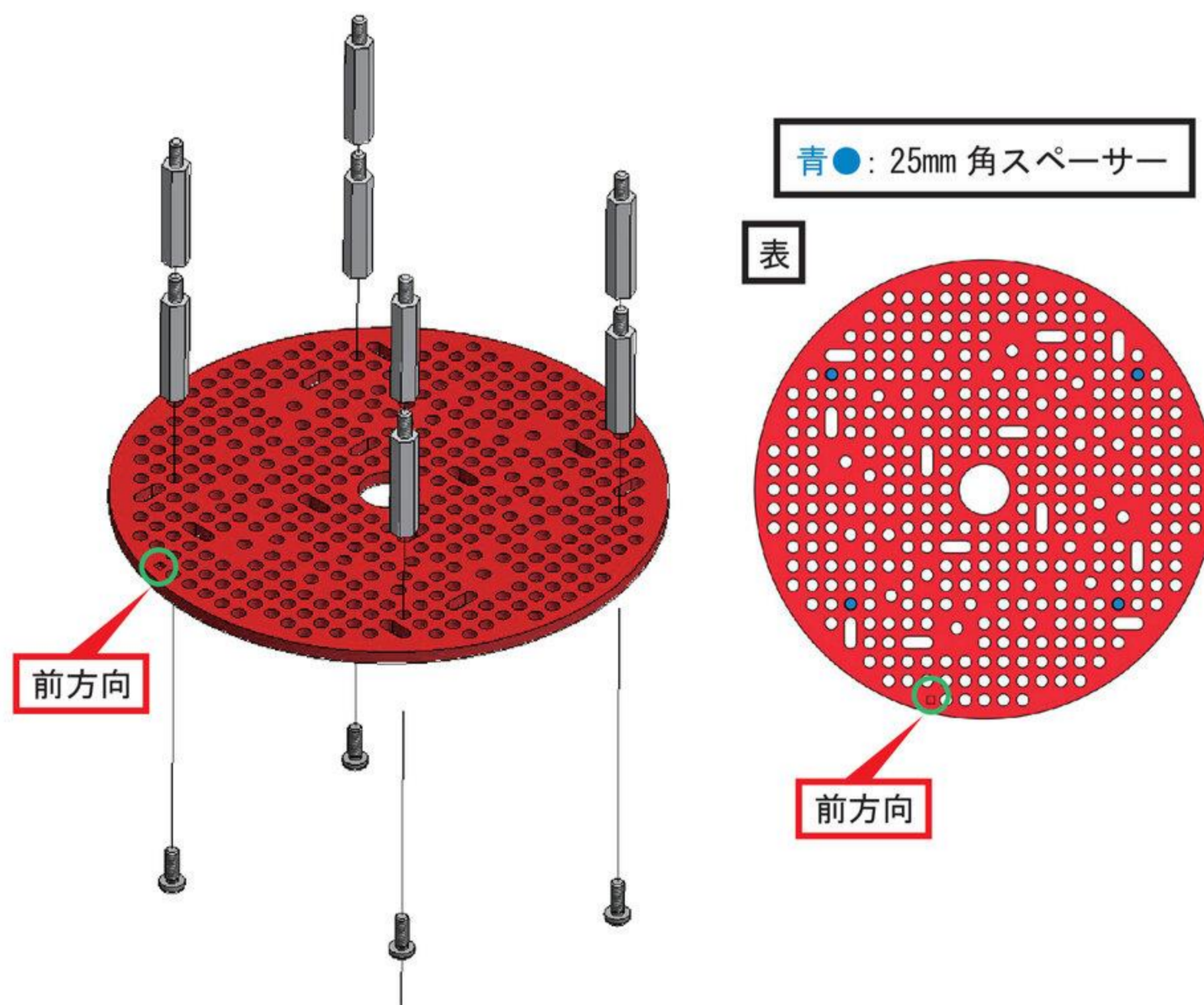


図 2-2 ベース部の組み立て

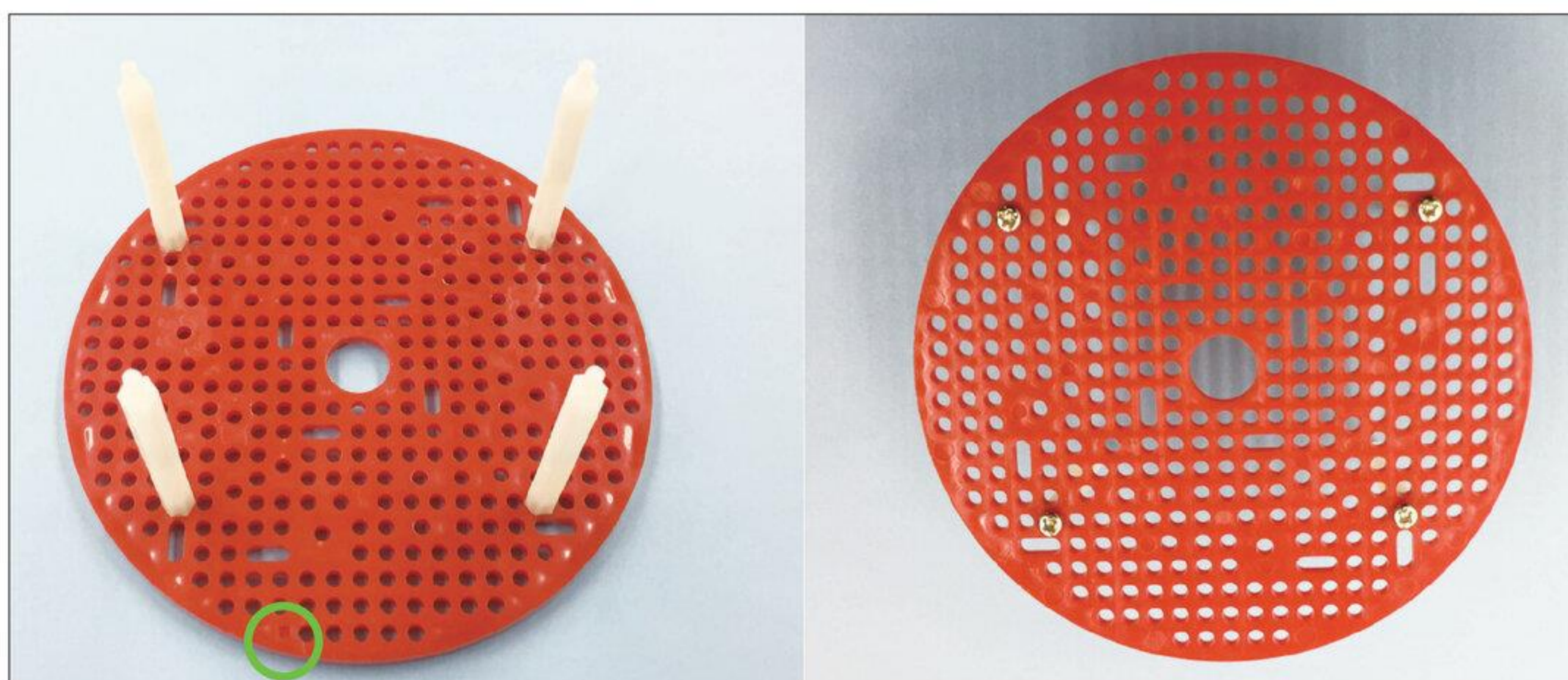


図 2-3 25mm 角スペーサーの取り付け

講

ベース部の前方方向となる□マークに注目させてください。

<組み立て手順②>

モーターL字ステー付きのギアドモーター (×2) を赤円形ボードに取り付けます。
M3L8 ネジ (×4) と M3 ナット (×4) を使用します。

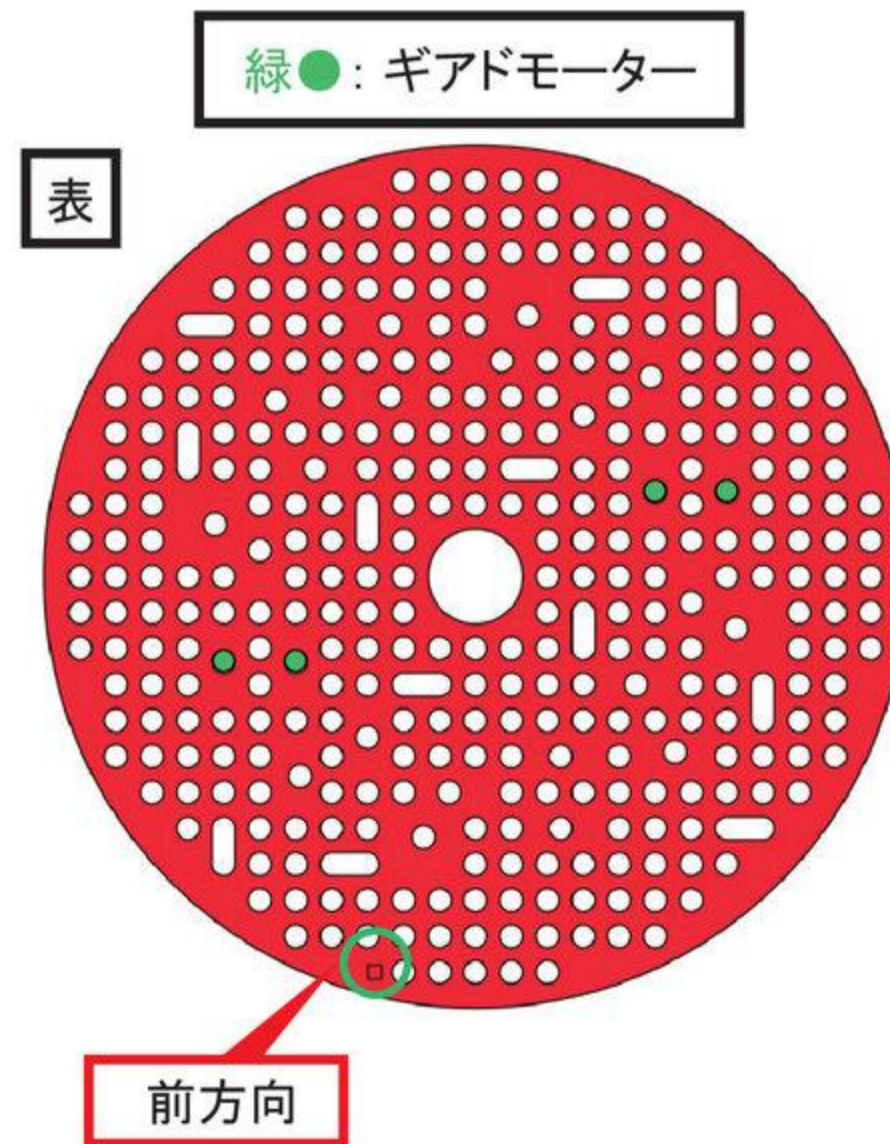


図 2-4 ギアドモーターの穴位置

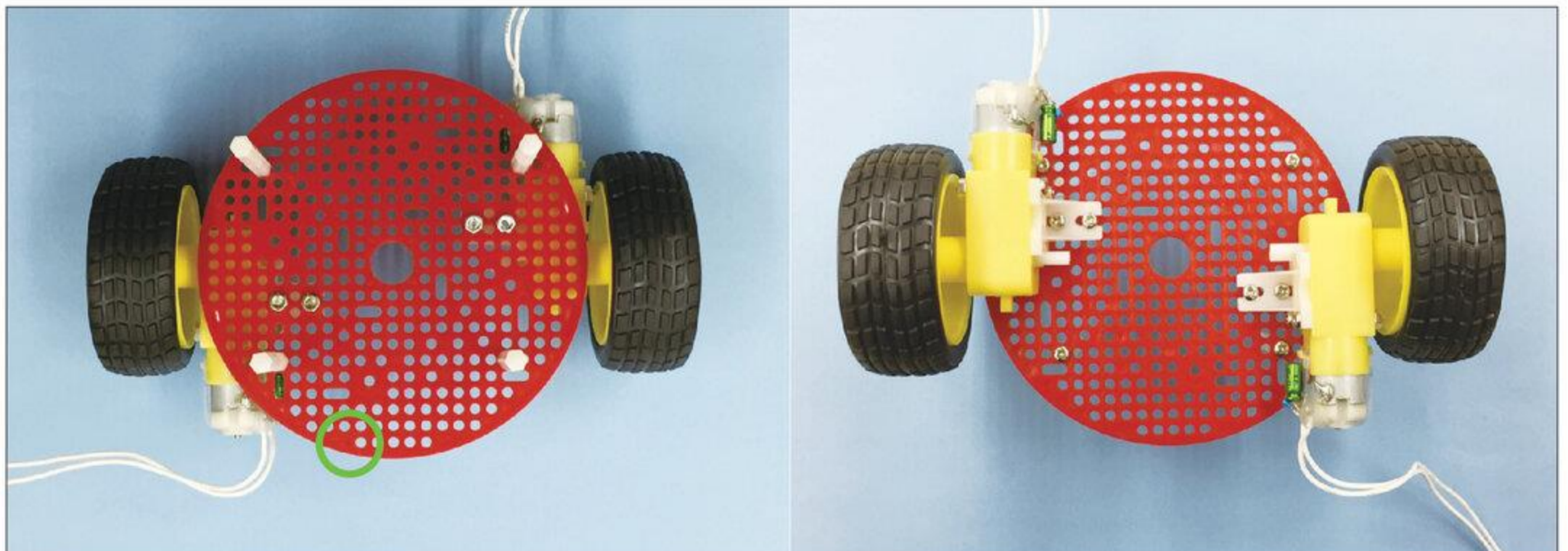


図 2-5 ギアドモーターの取り付け

2.2. 中段部の組み立て

ユニバーサルボードに電池ボックスを取り付けます。電池ボックスのコネクターが星印側に出るようにしましょう。M3L8 ネジ (× 2) と M3 ナット (× 4) を使用します。

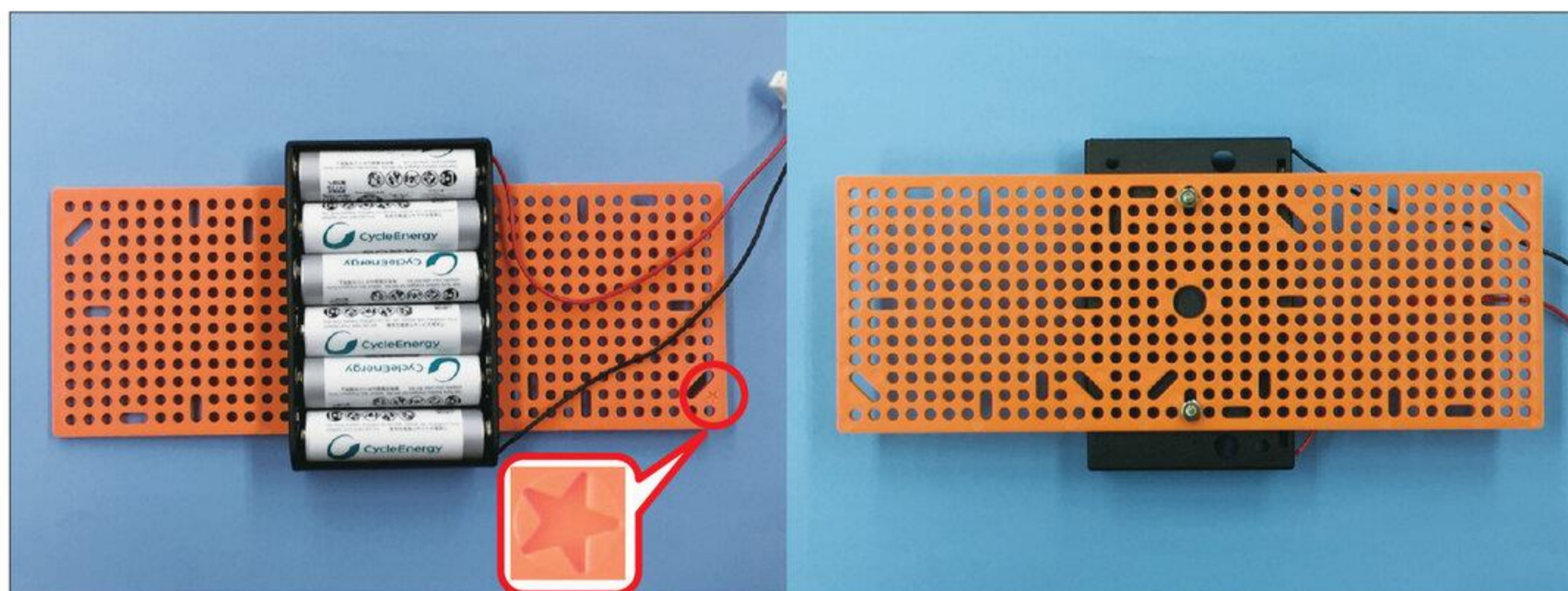


図 2-6 中段部の組み立て

2.3. ベース部と中段部の接続

ベース部に取り付けた 25mm 角スペーサー (× 4) にユニバーサルボードをはめ込み、かんつう貫通させた 25mm 角スペーサー 4 つのうち、3 つに 30mm 角スペーサー (× 3) を取り付け固定します。

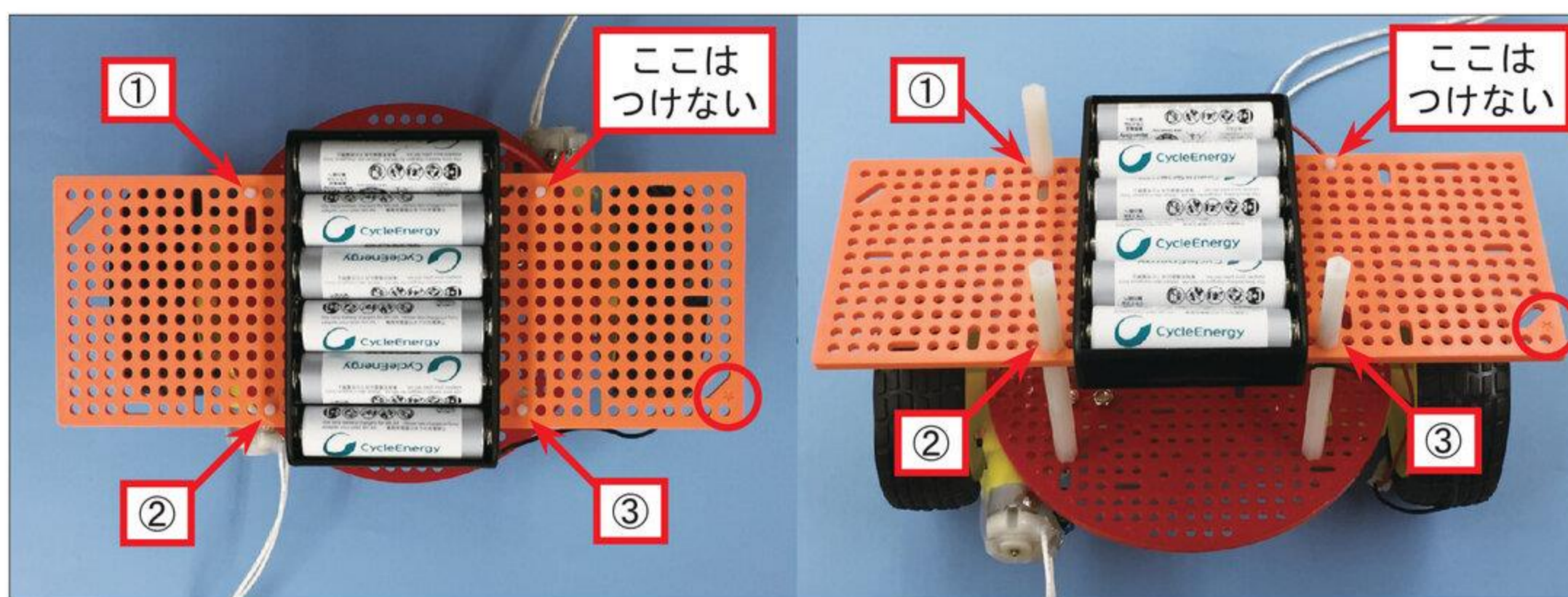


図 2-7 ベース部と中段部の接続

講

中段部のユニバーサルボードには、後ほど、星印側とは逆側に 301 ブレッドボード (回路) を取り付けます。

2.4. 回路と上段部の取り付け

次に、倒立振り子ロボットの制御を行うために非常に重要な回路を組み立てます。

<組み立て手順①>

301ブレッドボード、ジャンパー線（×6）、可変抵抗ボリューム（×2）を用意して、**図 2-8** を参考に回路を作りましょう。姿勢検出ユニットへの配線は後の手順で行うので、ここでは301ブレッドボードの配線のみを行いましょう。

なお、ジャンパー線はテキストと同じ色でなくてもかまいませんが、なるべく長いものを使うようにしましょう。また、可変抵抗ボリュームは、**図**と同じく手前側にピンがくるように取り付け、穴から抜けないようしっかりとさし込みましょう。

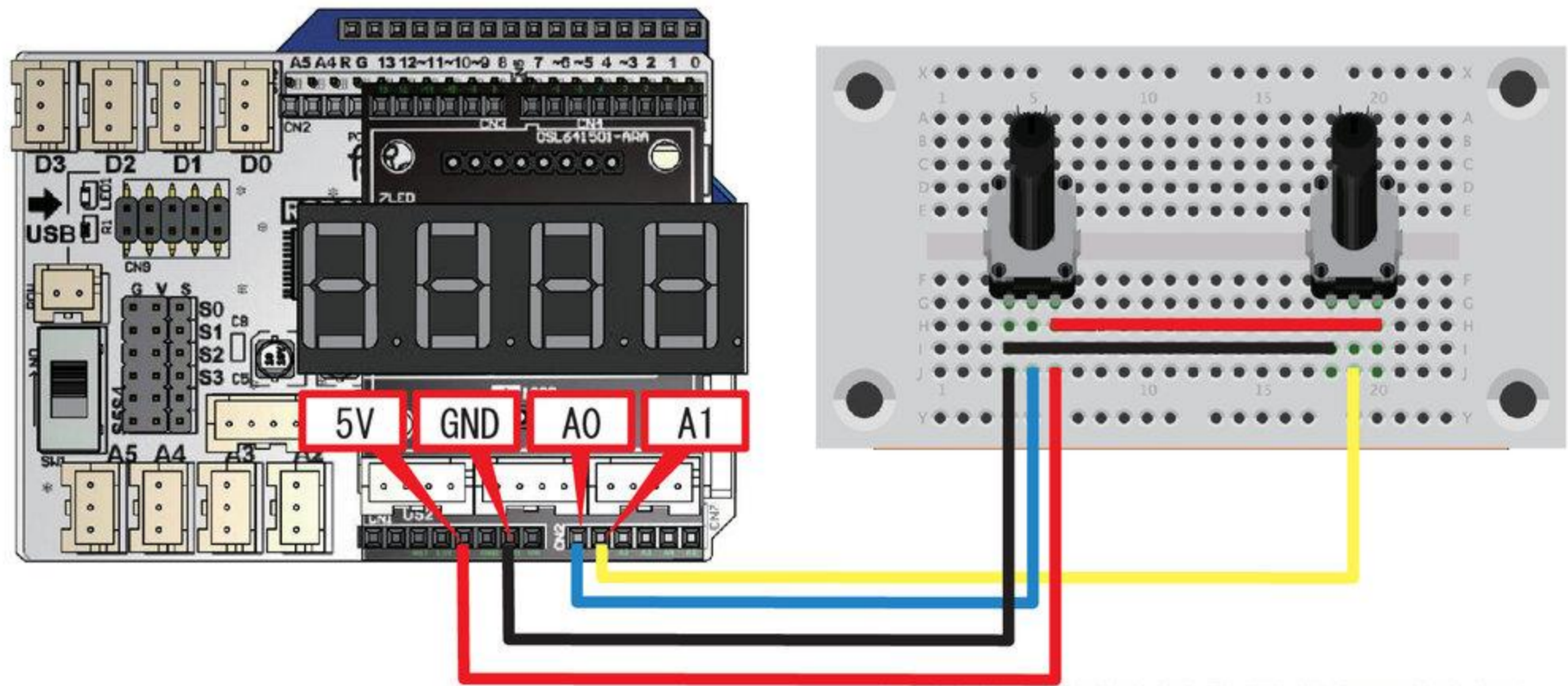


図 2-8 回路図

<組み立て手順②>

中段部に、上段部と301ブレッドボードの回路を取り付けます。上段部は**図 2-9**の位置に、M3L8ネジ（×3）で3点をとめ、取り付けます。前方方向の2点をとめた後に、位置合わせをしながら後方の1点もとめましょう。

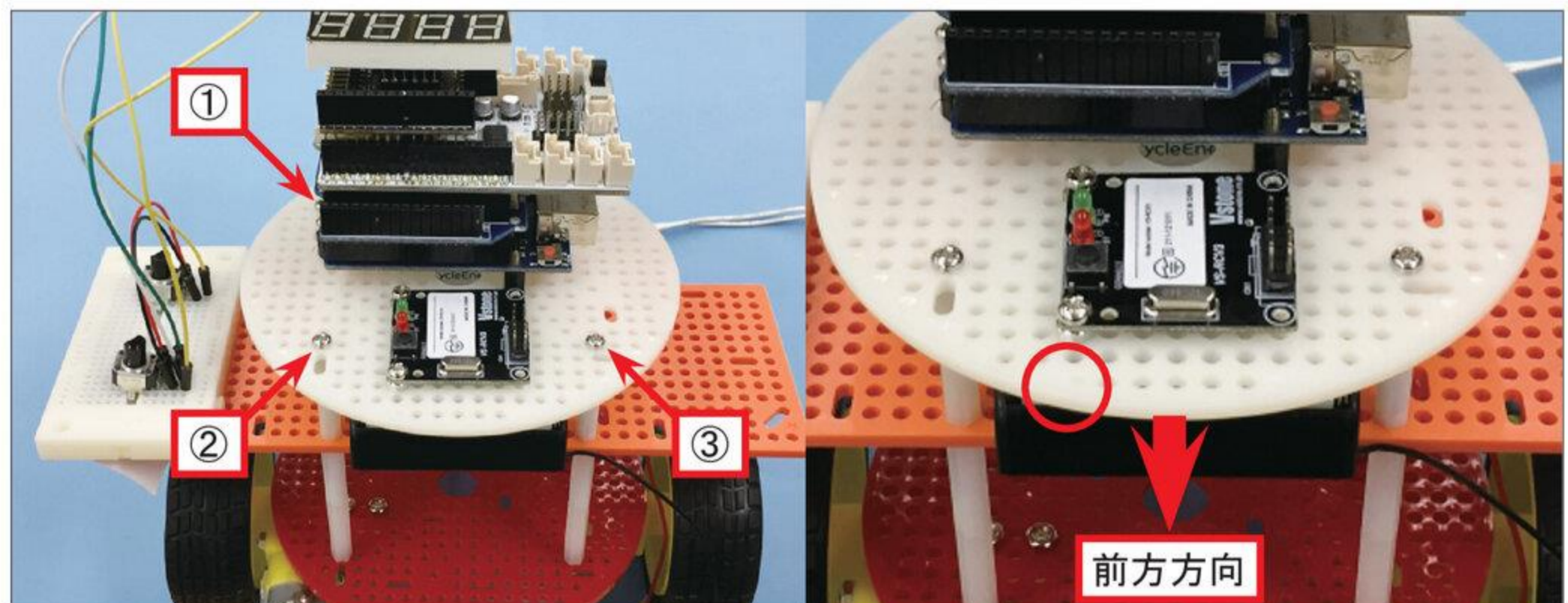
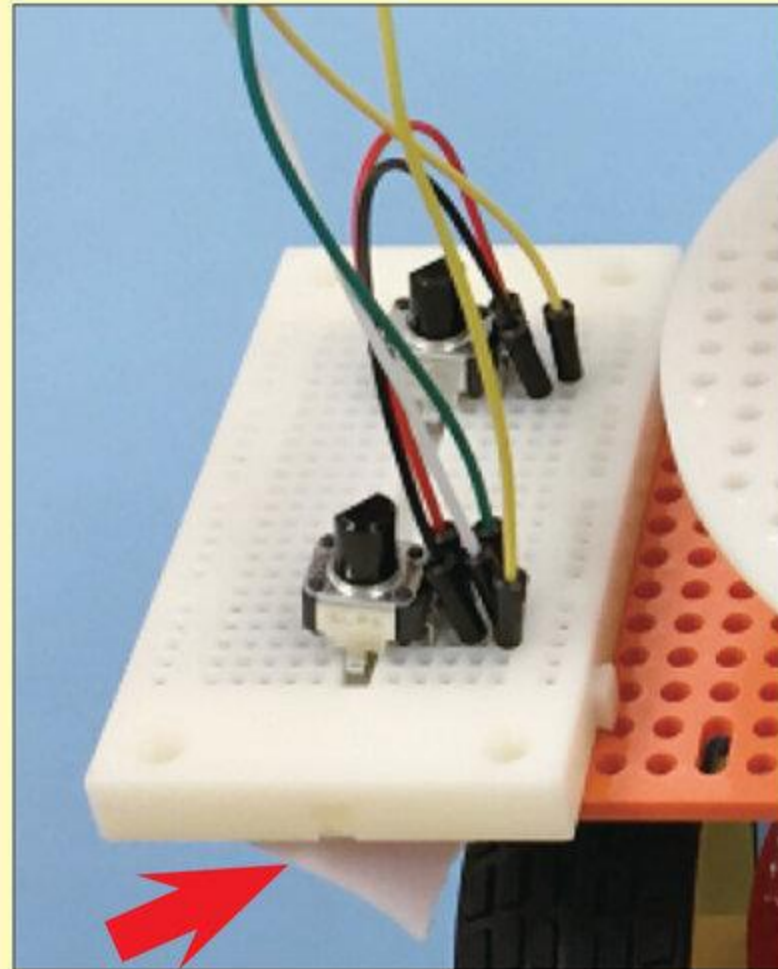


図 2-9 上段部と301ブレッドボードの取り付け



POINT

301ブレッドボードの裏面は両面テープになっています。フィルム面を1/3～1/2ほどはがしてから、接着させる部分に軽くはり付けましょう。粘着力が強力なため、全面をはり付けるとはがれなくなるので、注意しましょう。



<組み立て手順③>

図2-8の回路図を参考に、姿勢検出ユニットへのジャンパー線の配線をして、無線受信モジュールとロボプロシールドをリボンケーブルで接続したら完成です。

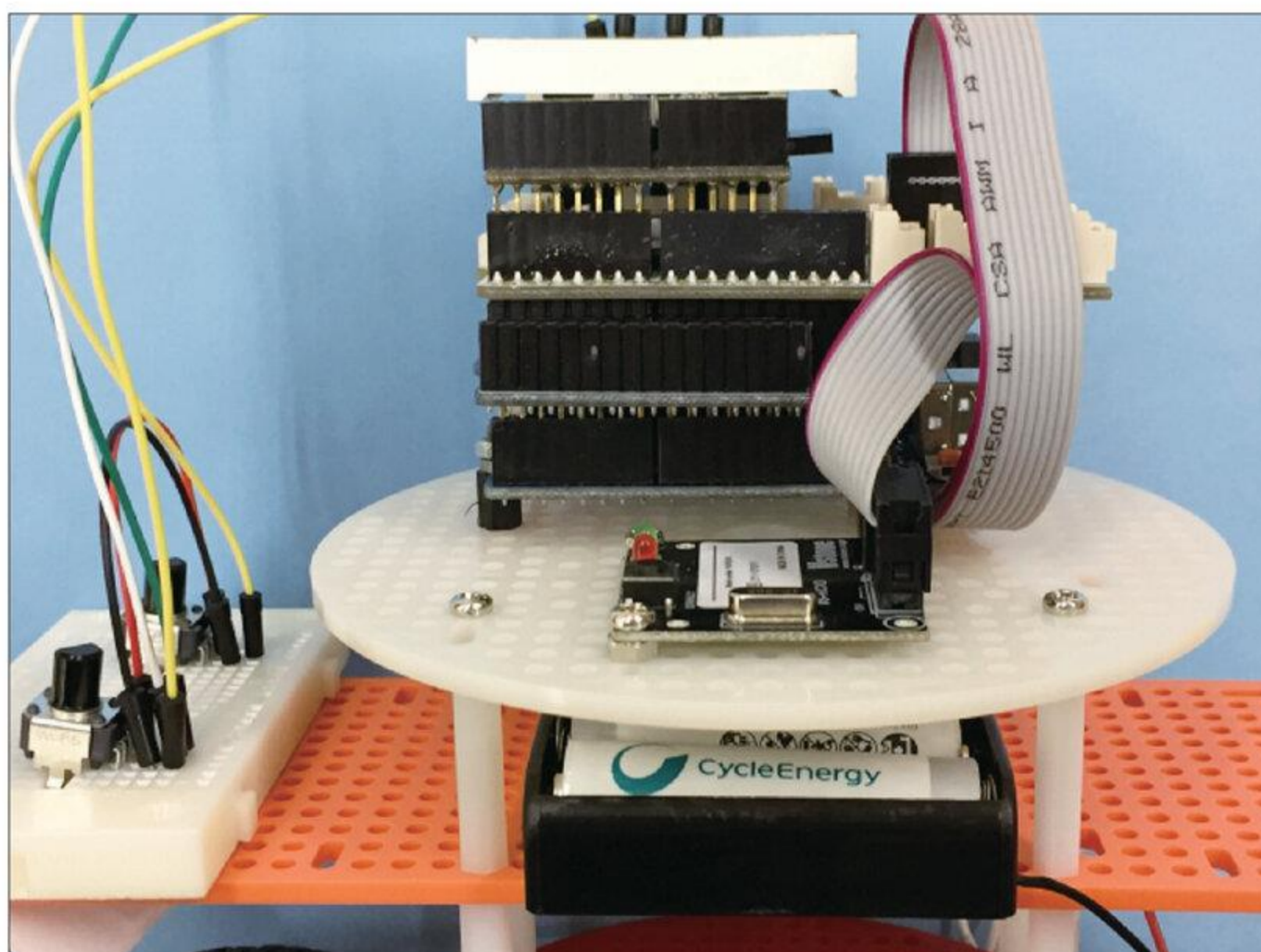


図2-10 リボンケーブルの接続

2.5. 動作確認

今回は、モーターと可変抵抗ボリュームの動作確認を行い終了します。

1) モーターの確認

モーターの配線は、右モーターをロボプロシールドの [MC0] に、左モーターを [MC2] に接続してください。

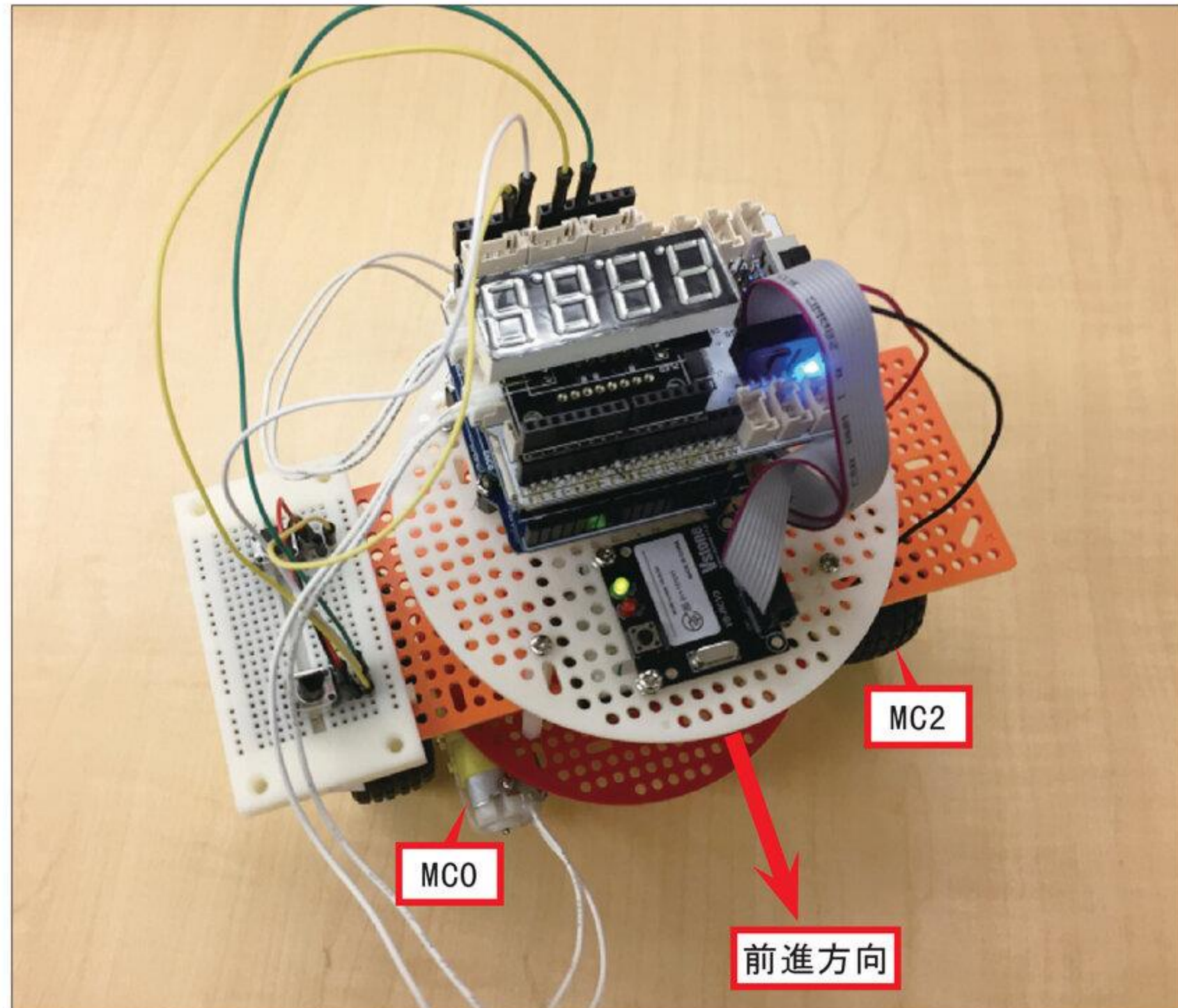


図 2-11 モーターの配線

以下のプログラムをそれぞれ実行しましょう。

🔄 プログラムの書き込み

RoboticsProfessorCourse 2 > Inverted5 > MC0Test

RoboticsProfessorCourse 2 > Inverted5 > MC2Test

実行結果：それぞれ [MC0] と [MC2] に接続されたモーターが、1秒ごとに正回転と逆回転をくり返す。

[MC0] と [MC2] をそれぞれ確認してください。指定したモーターがきちんと動いているかを確認めます。そして、もし動かなかった場合は、シールドへの接続がうまくいっていない可能性があります。シールドの接点などの確認をしましょう。

2) ^{かへんていこう}可変抵抗ボリュームの確認

続いて、301ブレッドボード上の^{かへんていこう}可変抵抗ボリュームの動作確認です。
まず、以下のプログラムをそれぞれ実行しましょう。

∞ プログラムの書き込み

RoboticsProfessorCourse 2 > Inverted5 > Volume0Test

RoboticsProfessorCourse 2 > Inverted5 > Volume1Test

実行結果：301ブレッドボード上の各^{かへんていこう}可変抵抗ボリュームを回すと、それぞれおよそ「0
～1023」の^{はんい}範囲で7セグメントLEDの値が変化します。

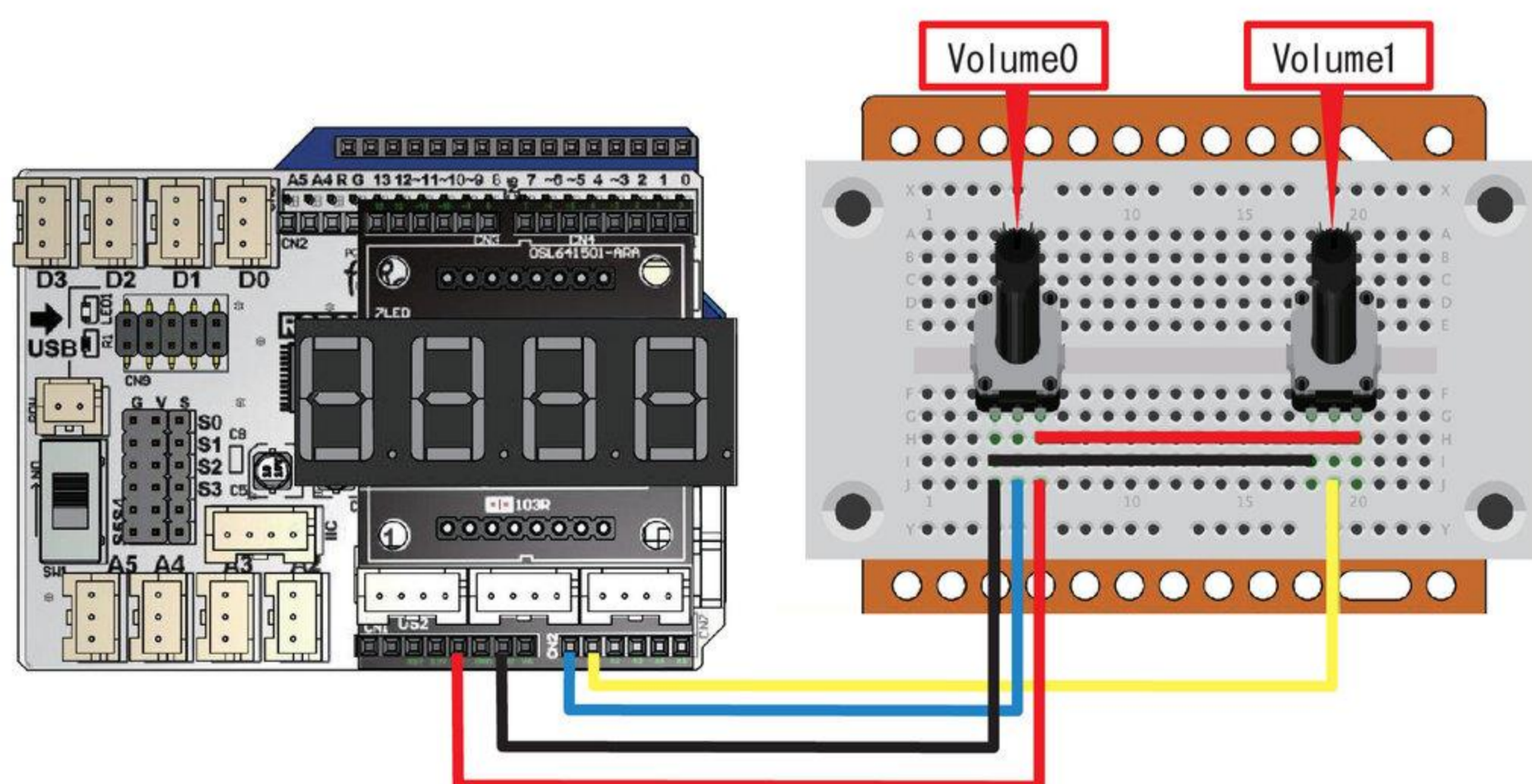


図 2-12 ポリュームテスト

正しく表示されなかった場合は301ブレッドボードの配線の再確認を行いましょう。また、ジャンパー線が断線などしている場合は交換しましょう。

講

ボリュームテストでは、可変抵抗ボリュームの回された量（角度）に応じて、アナログ入力値 0 ～ 1023 が7セグメントLEDに表示されます。

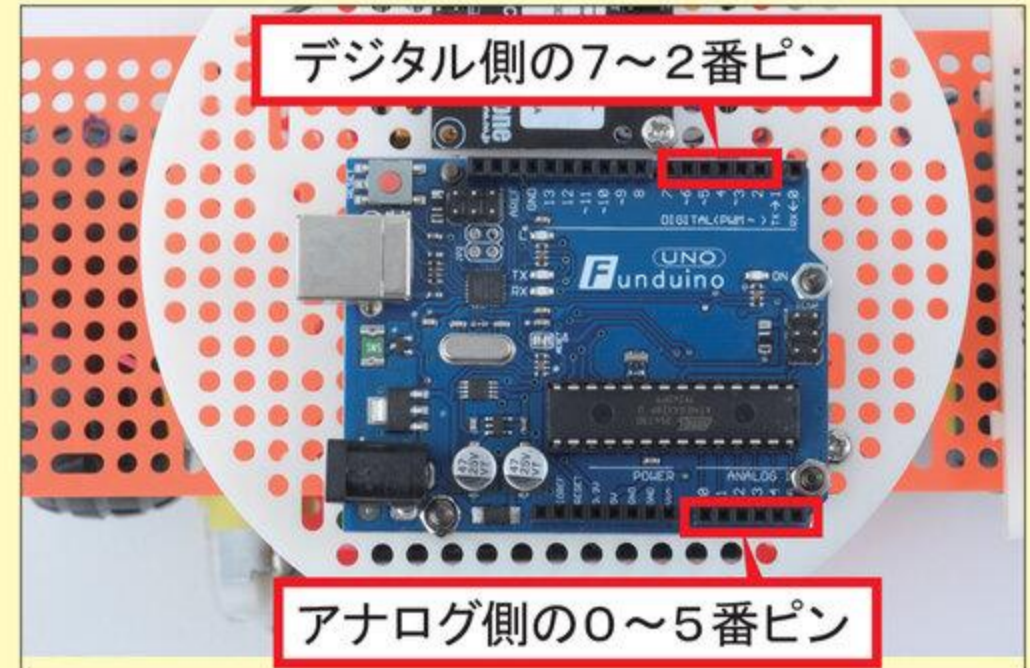
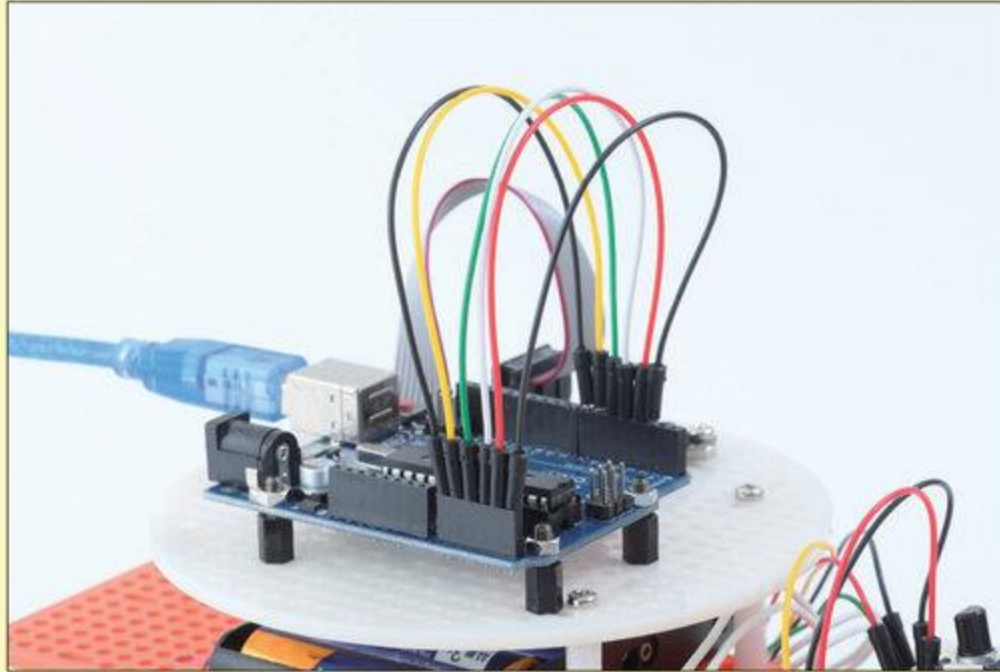


POINT

●ジャンパー線の導通チェック

うまく動かない場合はジャンパー線が断線している可能性があります。その場合は、チェックしましょう。

まず、以下の図のように、マイコンボードより上の全てのシールドを外し「デジタル側の7～2番ピン」を「アナログ側の0～5番ピン」にそれぞれ対応するように接続します。

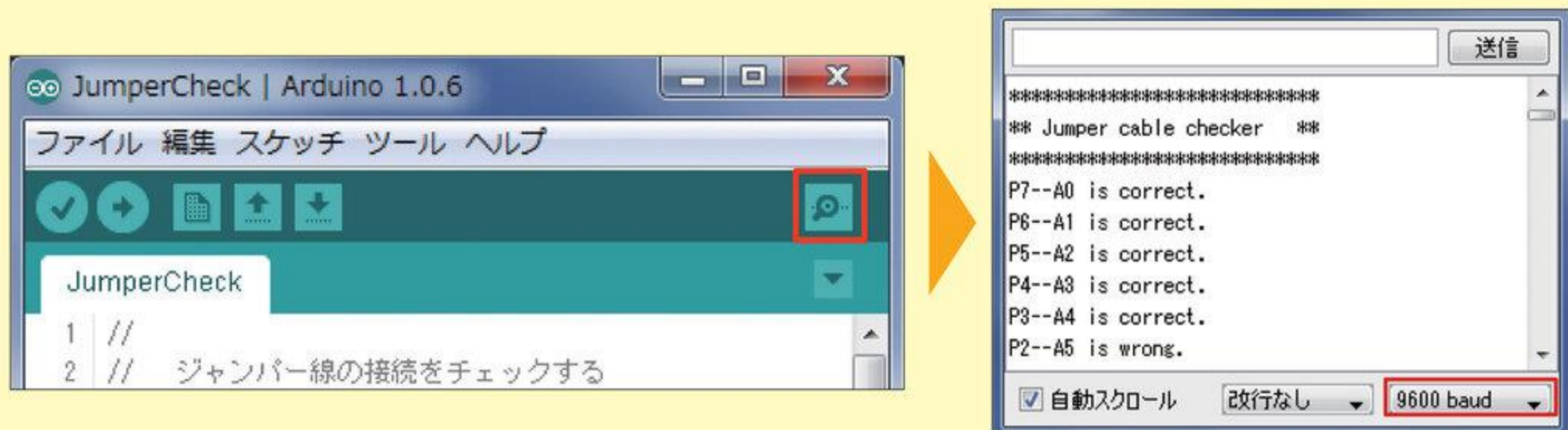


次に以下のプログラムを実行します。

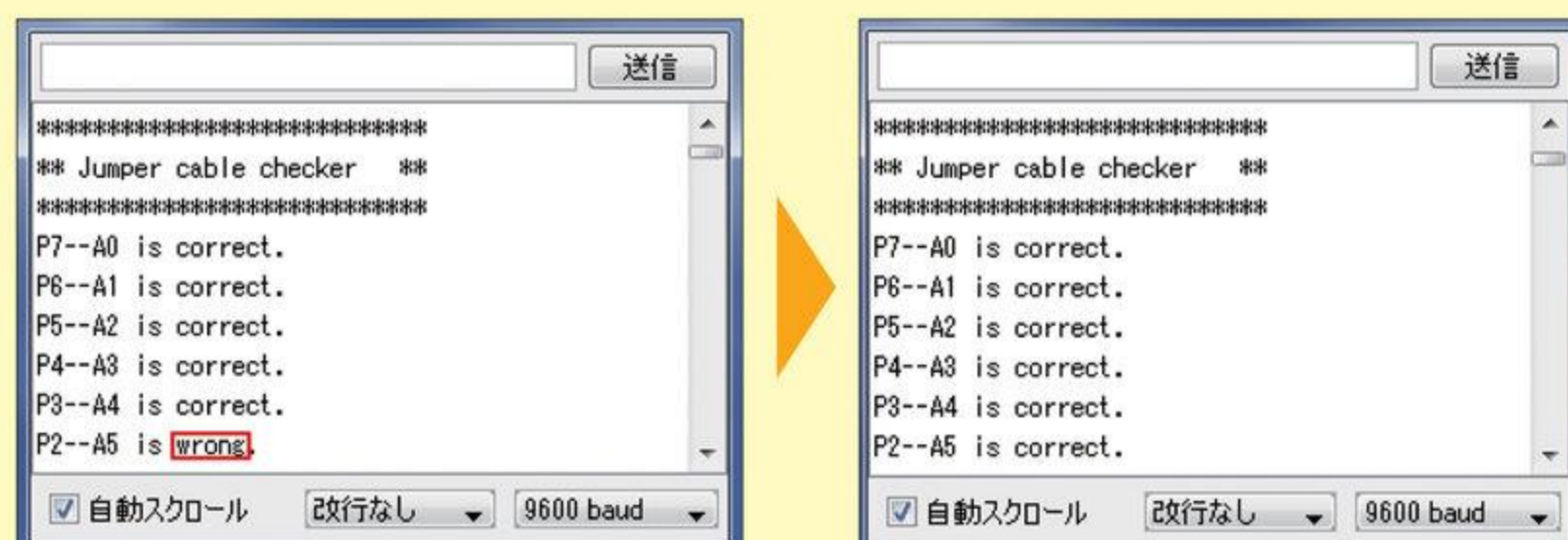
∞プログラムの書き込み

RoboticsProfessorCourse2 > MagicItemA1 > JumperCheck

画面右上の端の虫眼鏡マークをクリックし、シリアルモニターを開きます。「9600baud」に設定しましょう。



シリアルモニターに以下のように各ピンの導通状況が表示されます。つながっていれば「correct」、断線の場合は「wrong」と表示されます。「wrong」と表示されているものは入れかえて「correct」と表示されるものを使用するようにしましょう。



あとは、前のページで考えておいた動きを、このロボットに「実装」すれば倒立振り子ロボットの完成です。



POINT

- ① ロボットが前向きに倒れそうになったら、前向きに加速する。
- ② ロボットが後ろ向きに倒れそうになったら、後ろ向きに加速する。

もちろん「倒れないように姿勢を維持する」ところまで実現しようとする、非常に細かい設定や複雑な処理が必要です。

ただ、「ひとまず立たなくてもよいので、上の動きが最低限できるようにする」だけなら、モーターに回転命令を出すだけなのでもうプログラムをつくることのできるはずですね。

ステップアップ

これまでに登場したサンプルプログラムを活用し、「傾き具合に応じて、タイヤを回転させる」というロボットにしてみよう！



ヒント

まずは「傾き具合に応じて動作が変わる」というプログラムを探す必要があるね！

講

たとえば、第1回で扱った「AccelDrop」は、「傾き具合に応じてマトリクスLED上のドットが移動する」という処理がかけられています。ここにモーターの回転命令を付け加えれば、目的の動作をさせられることになります。

必要なのは以下の3要素です。

- ① モーター関連の命令を使うためのライブラリ「RPLib.h」を取り込む。
- ② MC0とMC2のモーターの名前を宣言する。
- ③ それぞれのモーターに、傾き具合に応じた速度で回転命令を出す。

このプログラムでは、前後方向の傾きから変数 y の値を算出し、点灯するLEDの y 座標を決定しています。

y の値は、最も前に傾いたときで0、最も後ろに傾いたときで7になりますが、この値のままではモーターの回転速度としては不適です。

よって、map文を使い `map(y, 7, 0, -255, 255);` とする等、変数 y をモーター速度のスケールに変換する工夫が必要になるでしょう。

「AccelDrop」を利用した解答例を巻末に記載します。

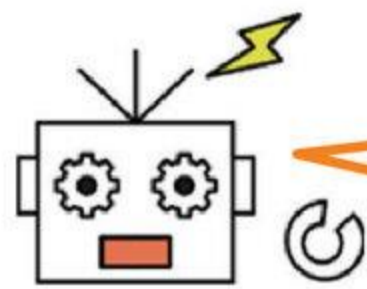
なお、このプログラムはあくまで「ロボットを立たせる」ためのものではなく必要な動作の方針を分かりやすくしたただけのもので、調整を重ねても倒立させるのは非常に困難です。より多くの設定事項を盛り込んだプログラムが次回に登場しますので、今回の課題はあくまで「プログラミングの練習課題」という捉え方でご覧ください。

3. まとめ（目安5分）

第5回では、「倒立振り子ロボット」の仕組みについてざっくりと説明しました。そして、ロボットの組み立てと配線、動作確認まで終わりました。

次回はいよいよ、フィードバックを用いて倒立させて動かします！ 私たちが日常的になにげなくしていることをロボットにさせる場合、さまざまな分析をして、ロボットに応用していくことになります。

いろいろと調整やカスタムの必要が出てきますが、トライ&エラーで頑張ってくださいませましょう！



次回は難易度が高いけど、がんばってイコー！

講

- 以下の授業の目標を再確認します。
 - ・ 倒立振り子ロボットの仕組みを学ぶ
 - ・ 倒立振り子ロボットを組み立てる
 - ・ 動作確認をする
- 次回のテーマは、「倒立振り子ロボットを動かして遊ぶ」であることを告知します。倒立振り子ロボットを持ち帰る際には断線防止のため、301ブレッドボードと各配線を外すようにご指導ください。また、電源の消し忘れに注意してください。

《次回必要なもの》

次回は、今回使ったロボットと以下のパーツを持ってきてください。

また、電池残量が少なくなるとロボットの動作に差が出るので、交換用の電池も用意しておくといでしょう。

USB ケーブル	1	コントローラー	1
			

図 3-0 次回必要なもの

P.19 ステップアップ 解答例

```
#include <Sprite.h>           // Sprite before Matrix
#include <Matrix.h>
#include "Wire.h"
#include <math.h>
#include "I2Cdev.h"
#include "MPU6050.h"
#include <RPLib.h>

Matrix myMatrix = Matrix(11, 13, 1);
RPMotor mc0(MC0);
RPMotor mc2(MC2);

MPU6050 accelgyro; //姿勢センサーを使うときのオマジナイ

int16_t ax, ay, az;
int16_t gx, gy, gz;
int16_t mx, my, mz;
float temp;

void setup(){

    myMatrix.clear(); //マトリクスLEDの表示を消す
//----- 姿勢センサーの準備
    Wire.begin();
    accelgyro.initialize(); //姿勢センサーの準備
}

int x,y,velo;
void loop(){

    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
// 姿勢センサーの検出

    x = map(ax, -16000, 16000, 0, 7); // 0-7の間にする
    y = map(ay, -16000, 16000, 7, 0); // 0-7の間にする
    velo = map(y, 7, 0, -255, 255);

    myMatrix.write(x, y, HIGH); // LEDを点灯
    delay(50);
    myMatrix.write(x, y, LOW); // LEDを消す
    mc0.rotate(velo);
    mc2.rotate(velo);
}
```