

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

二足歩行ロボット①

(第1回/第2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1回授業日 2024年 月 日

だい かい じゅ ぎょう び
第2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1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

二足歩行ロボット①

第1回

二足歩行ロボットの組み立て①

講師用

目 次

0. 二足歩行ロボットの組み立て①

0.0. 「二足歩行ロボットの組み立て①」でやること

0.1. 必要なもの

0.2. 二足歩行ロボットについて

1. 組み立て

1.0. ボディ部分の組み立て

1.1. 関節部分の組み立て

1.2. サーボモーターの仕組み

1.3. サーボモーターの原点位置合わせ

2. サーボモーターのプログラム

2.0. サーボモーターへの命令をシリアルモニターで確認する

2.1. プログラム「Servo Origin0」の確認

2.2. Servo ライブラリを見てみよう

2.3. サーボモーターの暴走を防ぐ

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。

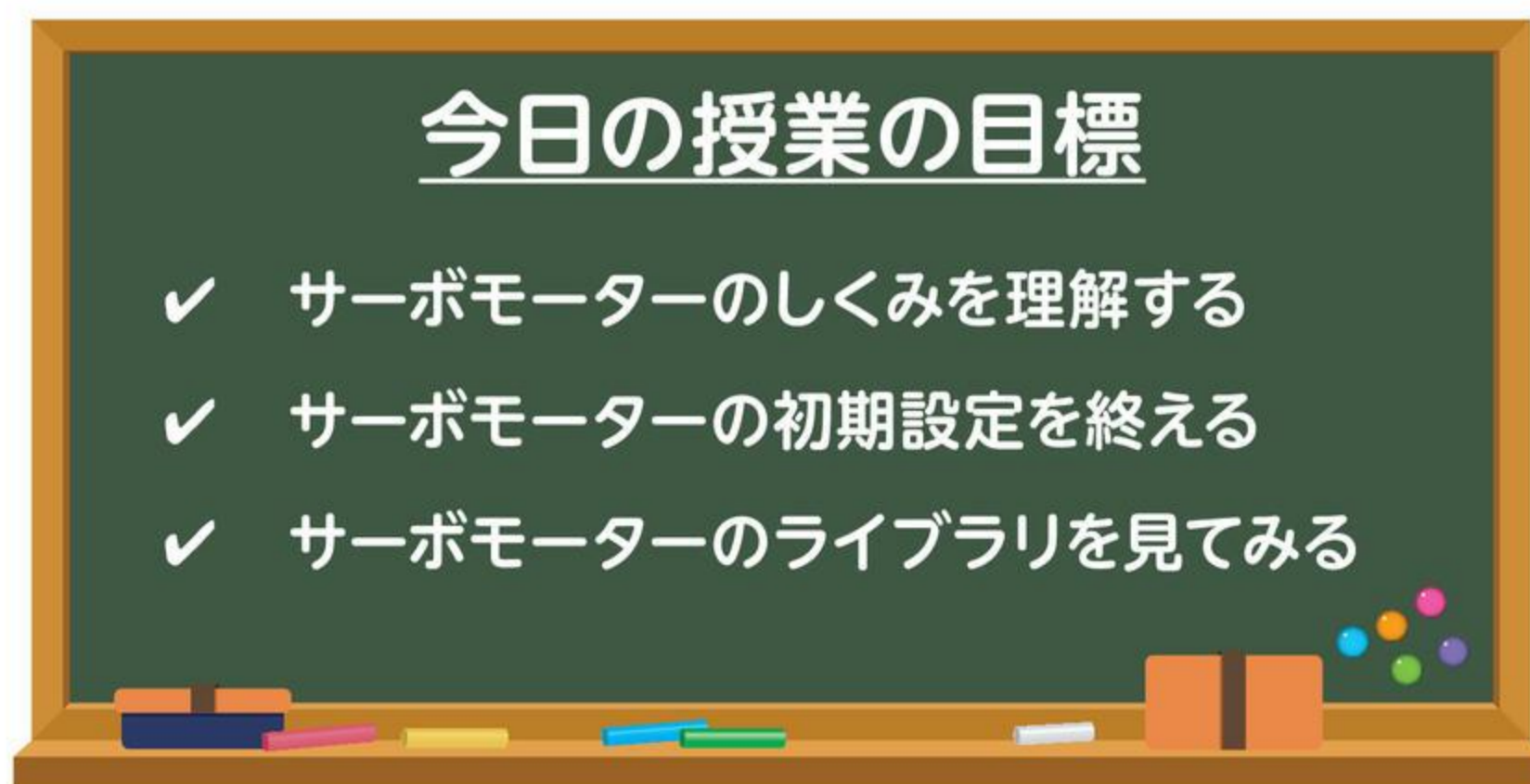
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(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. 二足歩行ロボットの組み立て① (目安 10分)

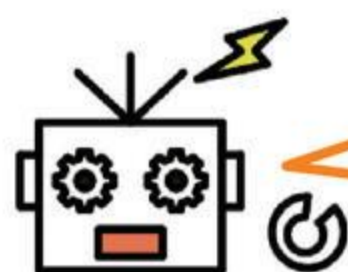
0.0. 「二足歩行ロボットの組み立て①」でやること



「二足歩行ロボット」では、第1回から第3回までにロボットの組み立てをしながら、サーボモーターのしくみや様々なプログラムを復習して、二足歩行ロボットをつくり上げる過程を勉強していきます。

ロボットといっても、いろいろな形のものがありますが、ヒューマノイド（人間型）ロボットはとても人気があります。人間型のロボットは、単純に動くだけでなく、周囲の状況をセンサーで検知したり、ディスプレイやスピーカーを活用して様々な情報を発信したりします。

まず今回は、ロボットの組み立てと、二足歩行ロボットの歩行には欠かせないサーボモーターのしくみの理解を深めていきます。ロボットを正しく歩行させるために、非常に重要なポイントです。気合を入れていきましょう！！



サーボモーターを制スル者は、歩行を制ス！

0.1. 必要なもの

SG90マイクロサーボモーターセットは、3年目セットに1セット追加があります。
(2年目セットから1個、3年目セットから1個を使用します。)

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	電池ボックス 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
			
超音波距離 ^{きょり} センサー 1	センサーカバー 1	センサーケーブル 1	M3L6 タッピングネジ (B) 2
			
MG995 サーボモーター 3	サーボモーター付属パーツ 3	SG90 マイクロサーボモーターセット 1	AC アダプター 1
			
A-1 (二足歩行ロボットパーツ) 1	A-2 (二足歩行ロボットパーツ) 1	A-3 (二足歩行ロボットパーツ) 1	B-1 (二足歩行ロボットパーツ) 1
			

図0-0 必要なもの①

B-2 (二足歩行ロボットパーツ) 1	B-3 (二足歩行ロボットパーツ) 1	B-4 (二足歩行ロボットパーツ) 2	C-1 (二足歩行ロボットパーツ) 2
			
C-2 (二足歩行ロボットパーツ) 2	C-3 (二足歩行ロボットパーツ) 2	C-4 (二足歩行ロボットパーツ) 2	D-1 (二足歩行ロボットパーツ) 2
			
サーボホーン 3	SG90 マイクロサーボモーターセット 1	AC 変換ケーブル 1	M2L6 タッピングネジ (A) 8
			
M3L6 タッピングネジ (B) 4	M2L12 ナベワッシャー-タッピングネジ (A) 12	針金フォーミング加工 2	M2L8 タッピングネジ (B) 12
			
M3L10 タッピングネジ (B) 2	M3L6 フラットヘッドビス 13	オイルブッシュ (二足歩行ロボット用) 32	
			

図0-1 必要なもの②

0.2. 二足歩行ロボットについて

人間を模造した二足歩行ロボットは、アニメや映画でも大変人気がありますね。このような人間型のロボットは「ヒューマノイドロボット」とも呼ばれています。歴史的にヒューマノイドロボットが研究されはじめたのは1970年頃で、ロボットの歩行原理、上半身の機構や歩行の安定化の研究がくり返し行われて今日に至っています。

図0-2が今回製作する二足歩行ロボットです。両足、^{こし}腰の計3個のサーボモーターで歩行動作をします。

そして、両腕の2個のマイクロサーボモーターで腕の上げ下げをできるロボットです。第1・2回では、組み立てを行い、このロボットの歩行原理を勉強していきましょう。

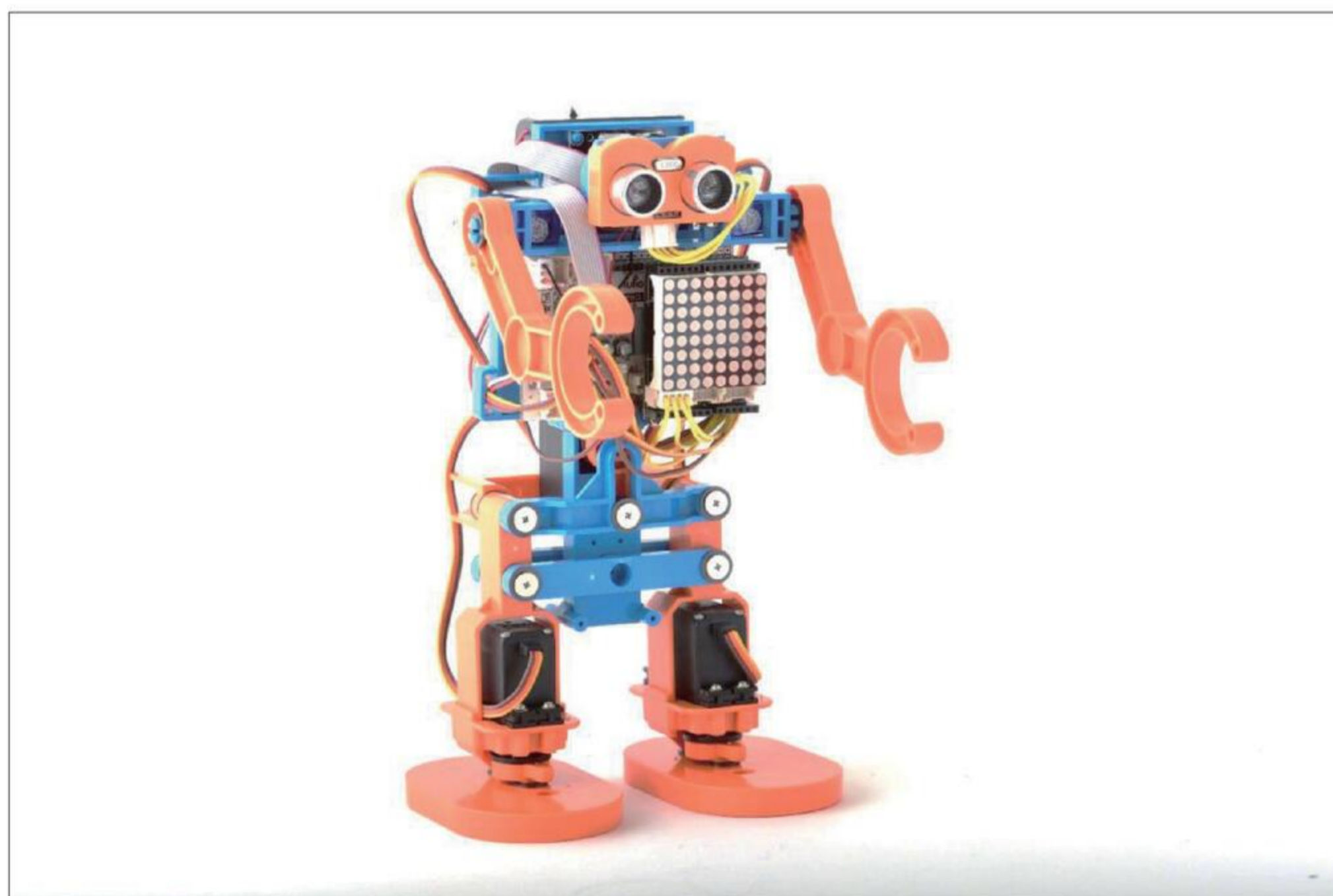


図0-2 二足歩行ロボット

1. 組み立て (目安 55分)

1.0. ボディ部分の組み立て

<組み立て手順①>

まずはロボットのボディ部分から組み立てていきます。図のようにA-1パーツに無線受信モジュールとマイコンボードをM2L6タッピングネジ(A) (×6) で組み付けてください。

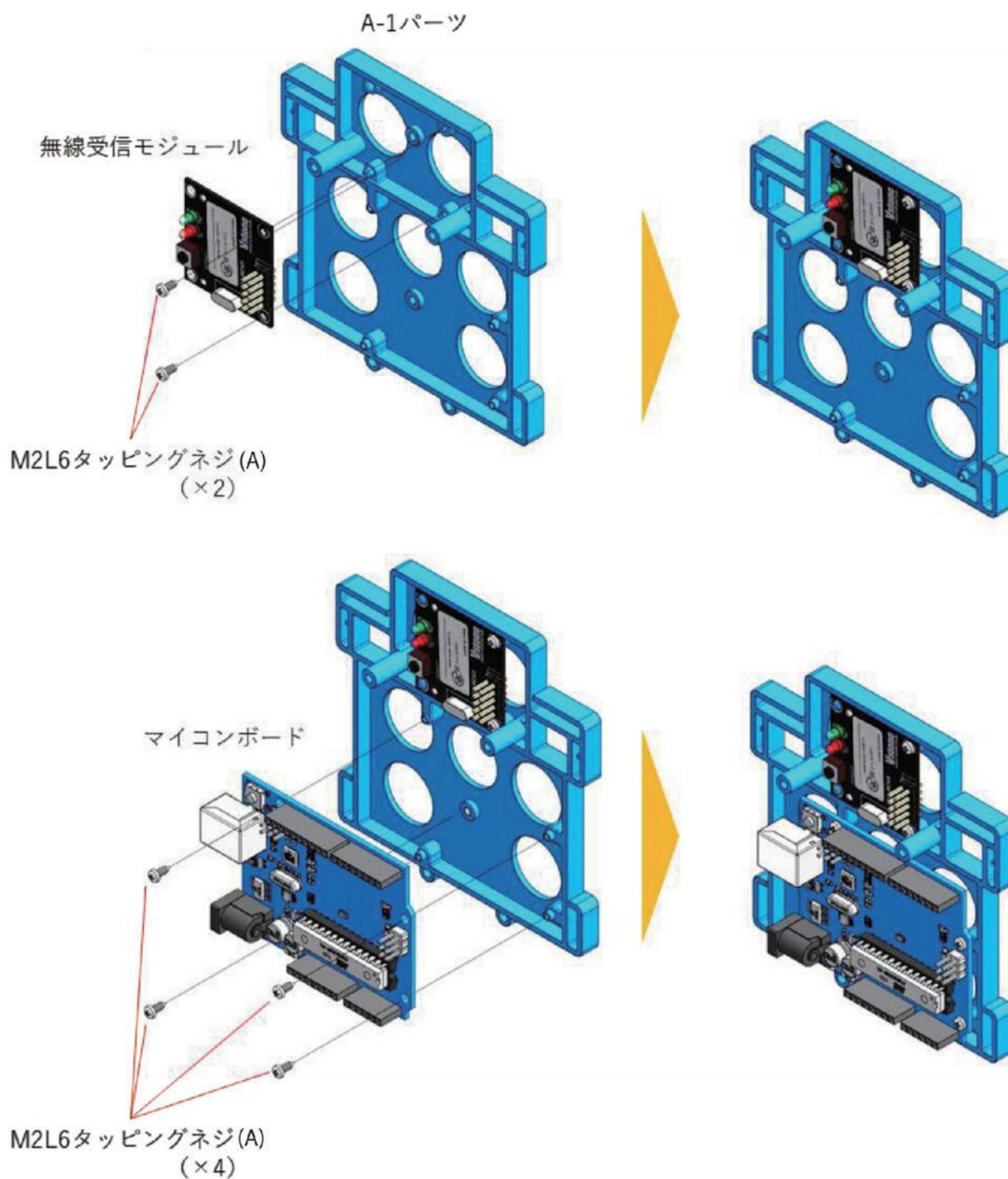


図1-0 無線受信モジュールとマイコンボードの組み付け

<組み立て手順②>

次に、図のように、マイコンボードを組み付けた面の反対側に、M3L6タッピングネジ (B) (×2) で電池ボックスを組み付けてください。

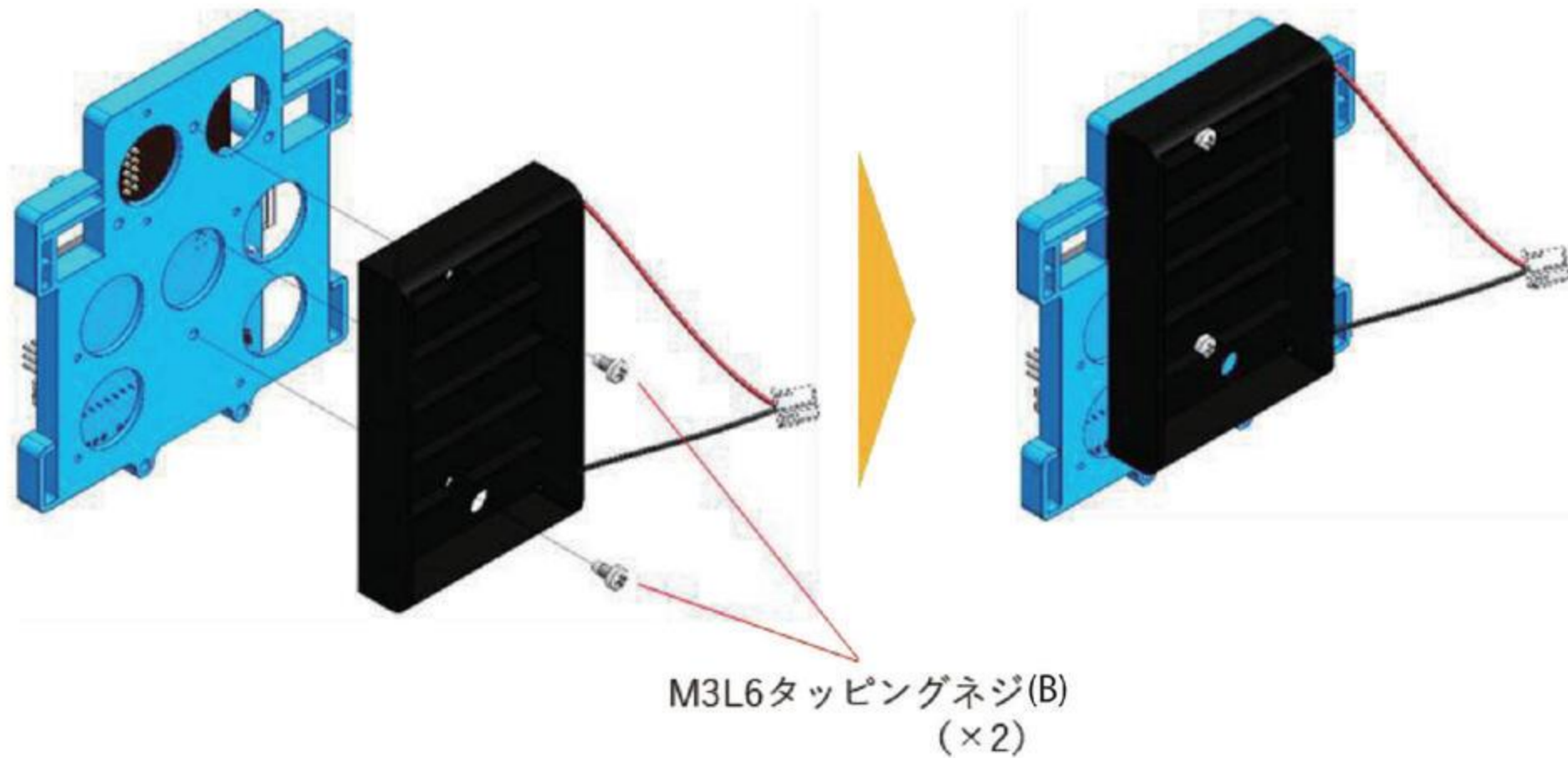


図1-1 電池ボックスの組み付け

<組み立て手順③>

次に、マイコンボードにロボプロシールドを接続し、ロボプロシールドの[CN9]コネクタと無線受信モジュールをリボンケーブルで接続します。また、電池ボックスのケーブルも接続します。

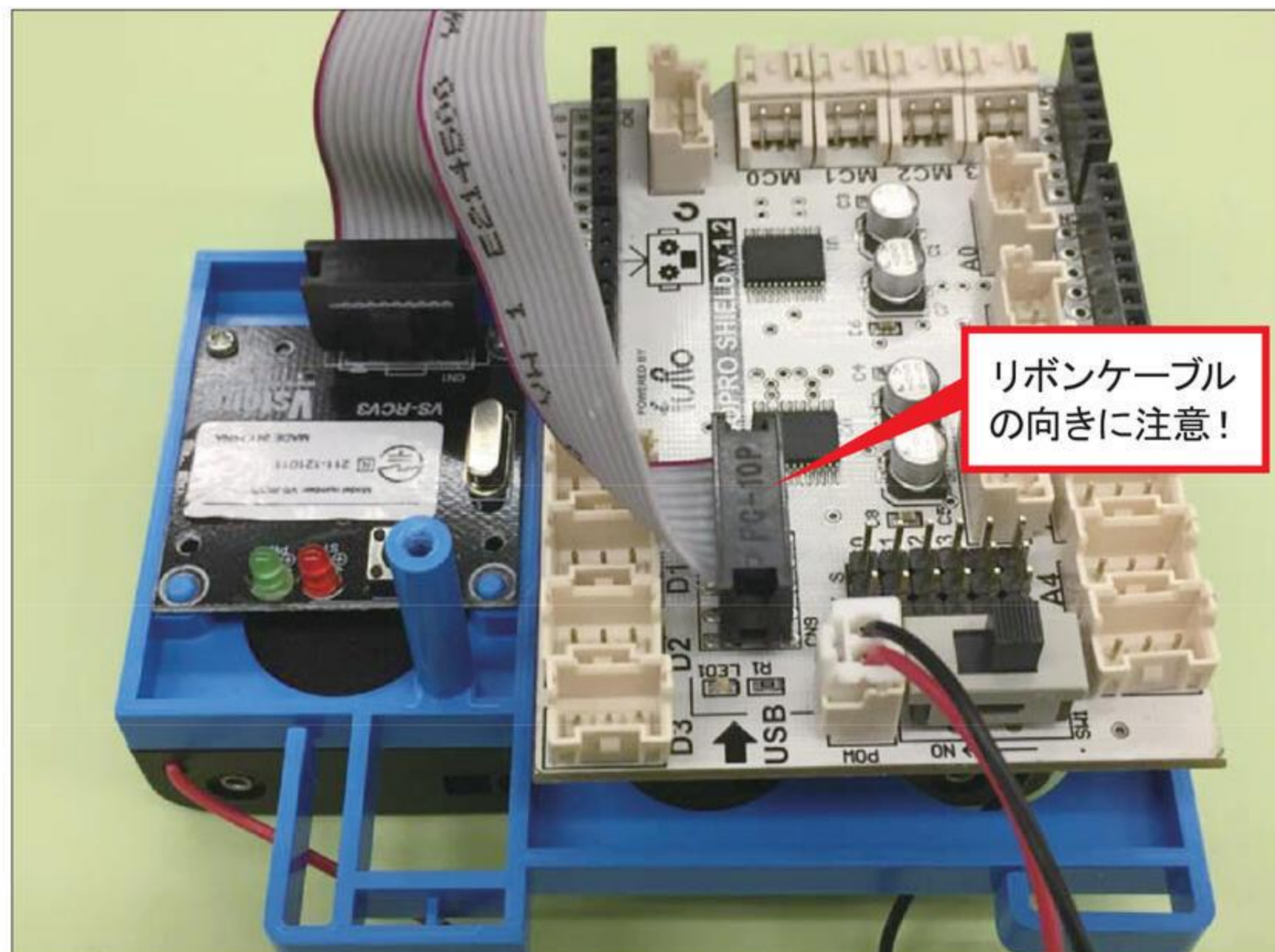


図1-2 ロボプロシールドと無線受信モジュールの取り付け

講

サーボモーターの原点位置調整を行うために電池ボックスをロボプロシールドに接続します。電池残量が少なくなっているものは交換させてください。

1.1. 関節部分の組み立て

次に、ロボットの足回りの関節部分の組み立てを行います。使用するサーボモーターに図のようにゴムブッシュをしっかりと取り付けて、ゴムブッシュの穴にハトメを入れてください。

以前の授業でこの工程を行っている場合、そのままの状態で使用します。

<組み立て手順①>

サーボモーターにゴムブッシュ (×4) をとりつけます。



図1-3 ゴムブッシュの取り付け

<組み立て手順②>

ゴムブッシュを取り付けた後に、ゴムブッシュの穴にサーボモーターの出力軸の反対側の面からハトメ (×4) を入れてください。



図1-4 ハトメの取り付け

合計3個のサーボモーターにゴムブッシュとハトメを取り付けてください。

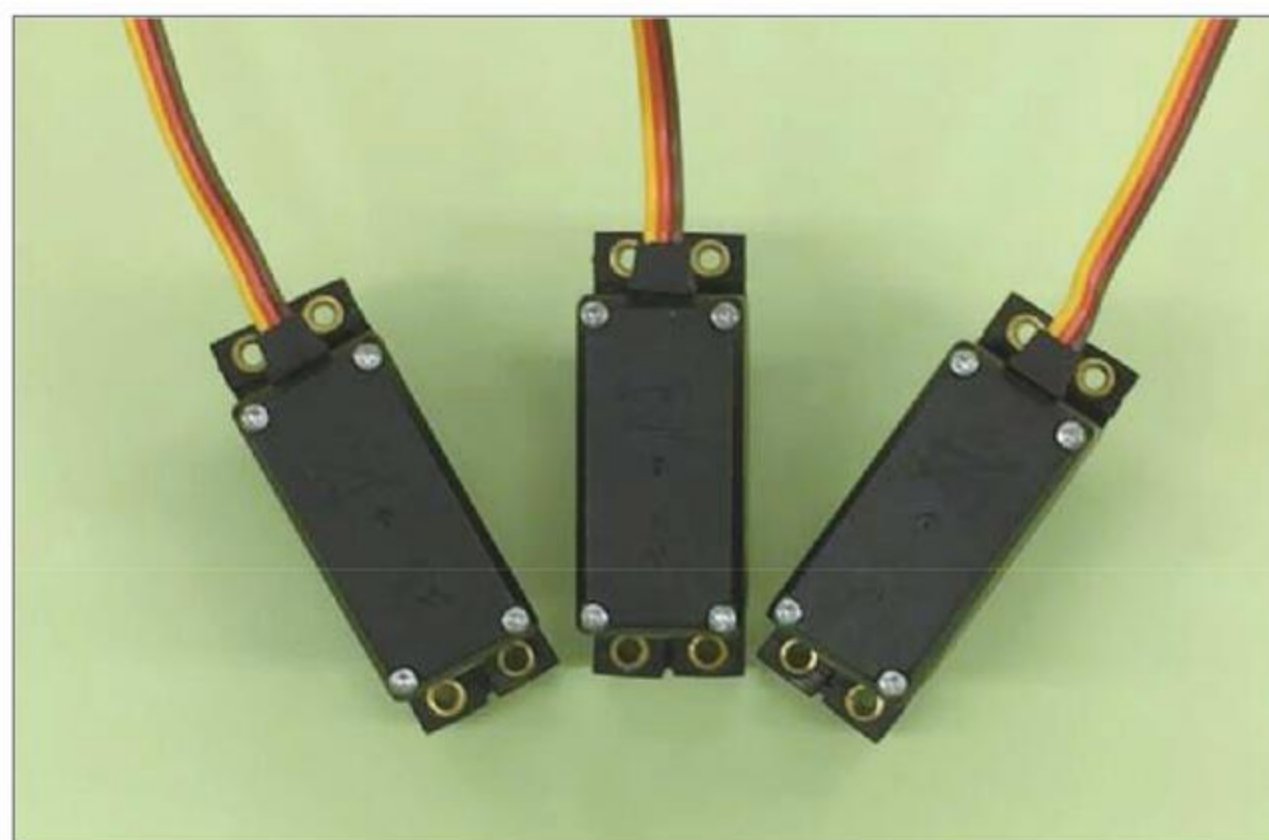


図1-5 ハトメの取り付け

<組み立て手順③>

次に、二足歩行ロボットの腰こしと足の関節になる部分を組み立てます。

A-2パーツ、C-2パーツ (×2) とM2L12ナベワッシャータッピングネジ (A) (×12) を準備して、図のようにサーボモーターを組み付けてください。

C-2 パーツは、同じものを2セット組み立ててください。

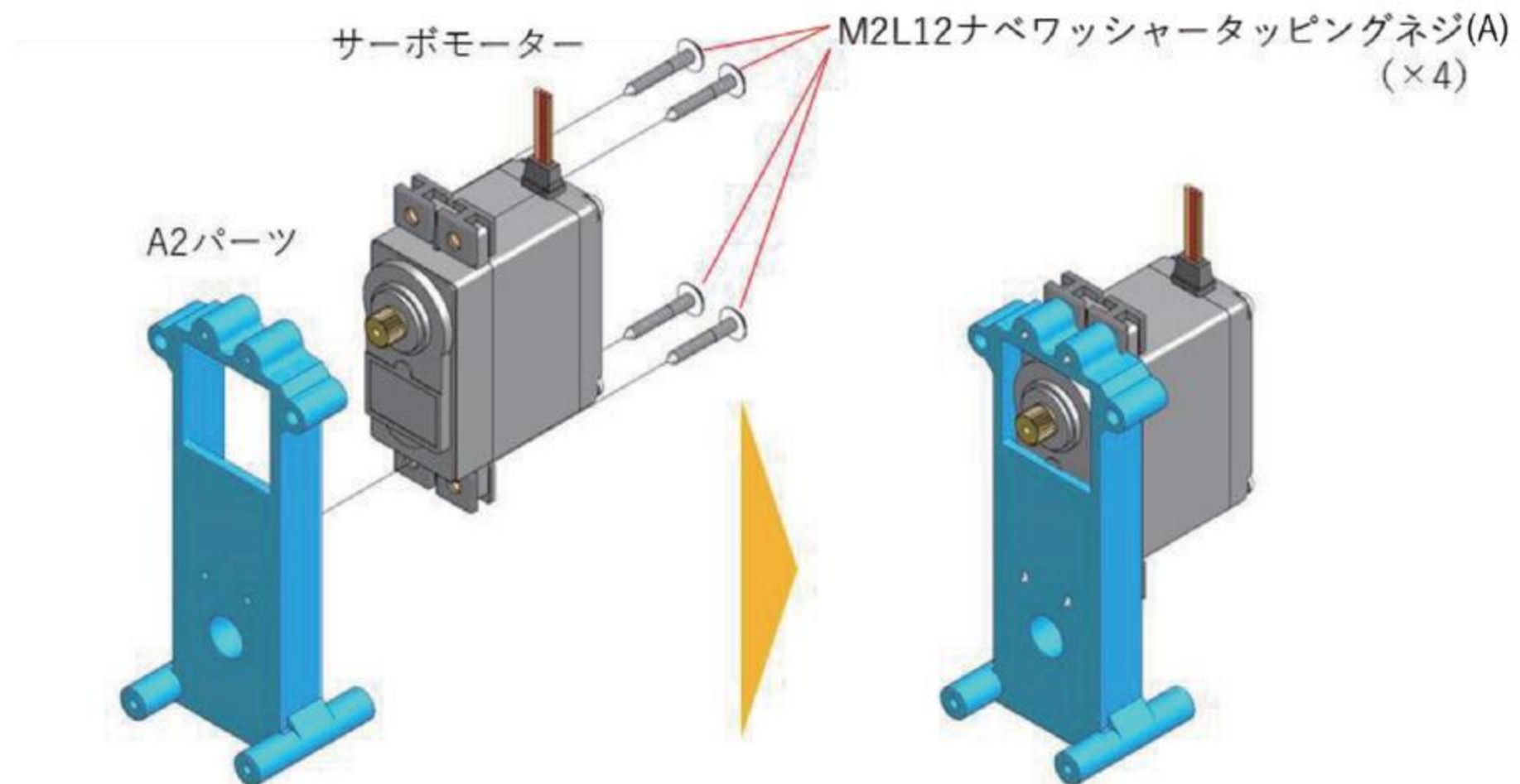


図1-6 A-2 (腰こし関節部分)

M2L12ナベワッシャータッピングネジ(A) (×4)

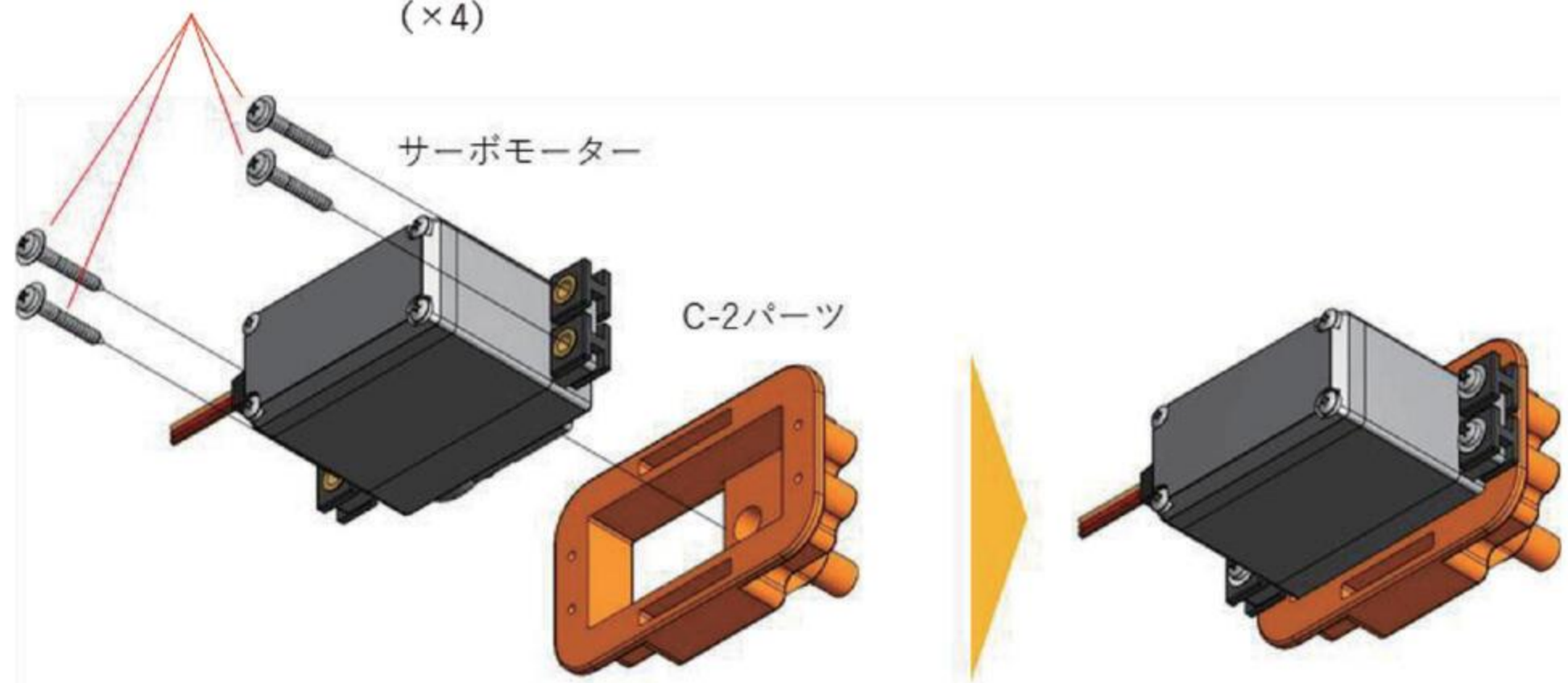


図1-7 C-2 (足こし関節部分)

⚠注意!

サーボモーター付属セットの中に、みぞが途中までしか刻まれていないネジが入っていることがあります。このネジは「M2L12ナベワッシャータッピングネジ (A)」ではありません。むりやり組みつけると、ネジがパーツ内で折れたり、パーツが破損したりする危険があります。注意しましょう。

<組み立て手順④>

次に、サーボモーターの出力軸^{じく}に取り付ける部分をつくります。

B-1パーツ、C-1パーツ (×2) と、M2L8タッピングネジ (B) (×12)、サーボホーン (×3) を準備して、図のように組み立ててください。

C-1パーツは、同じものを2セット組み立ててください。

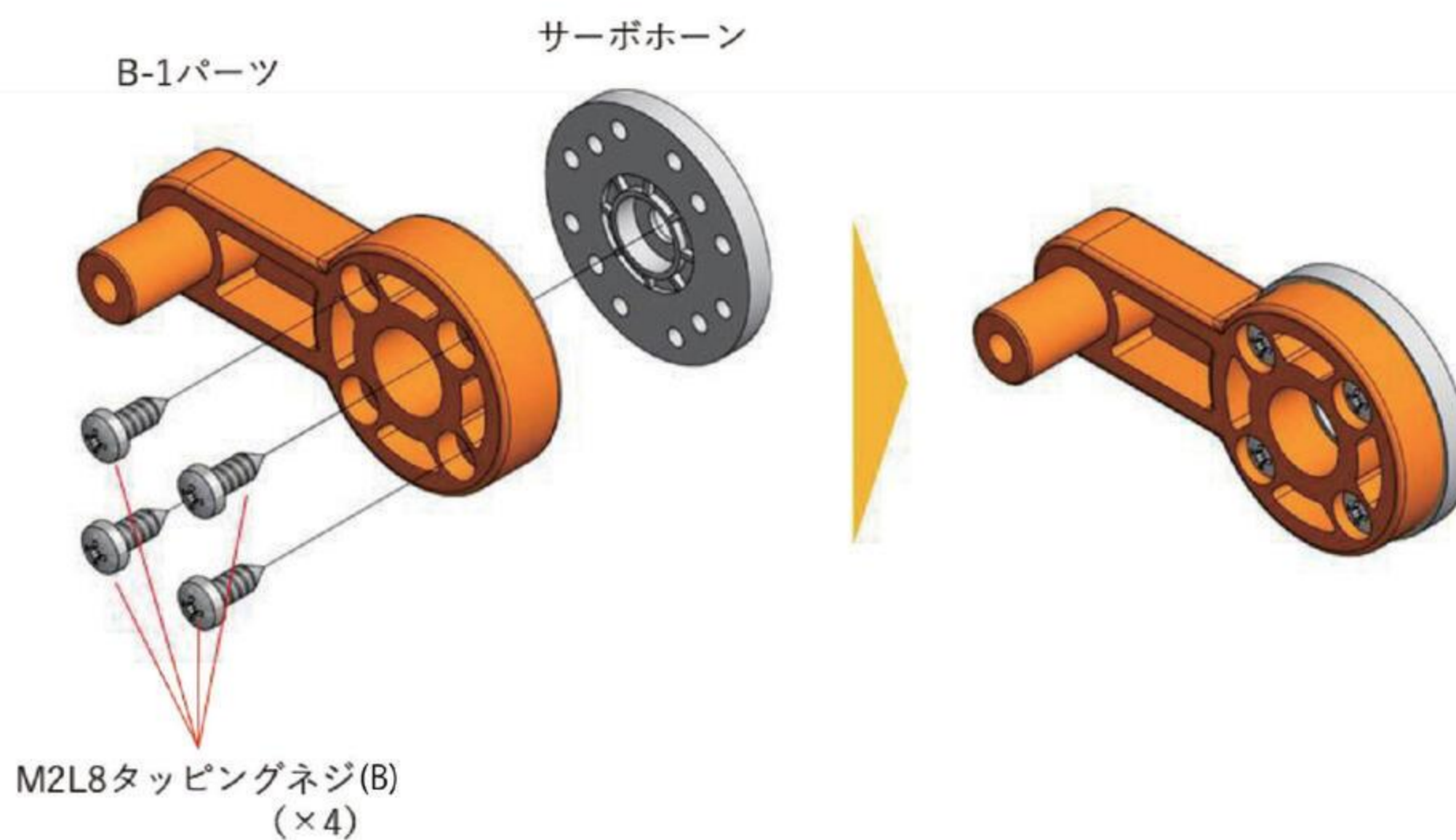


図1-8 B-1パーツ

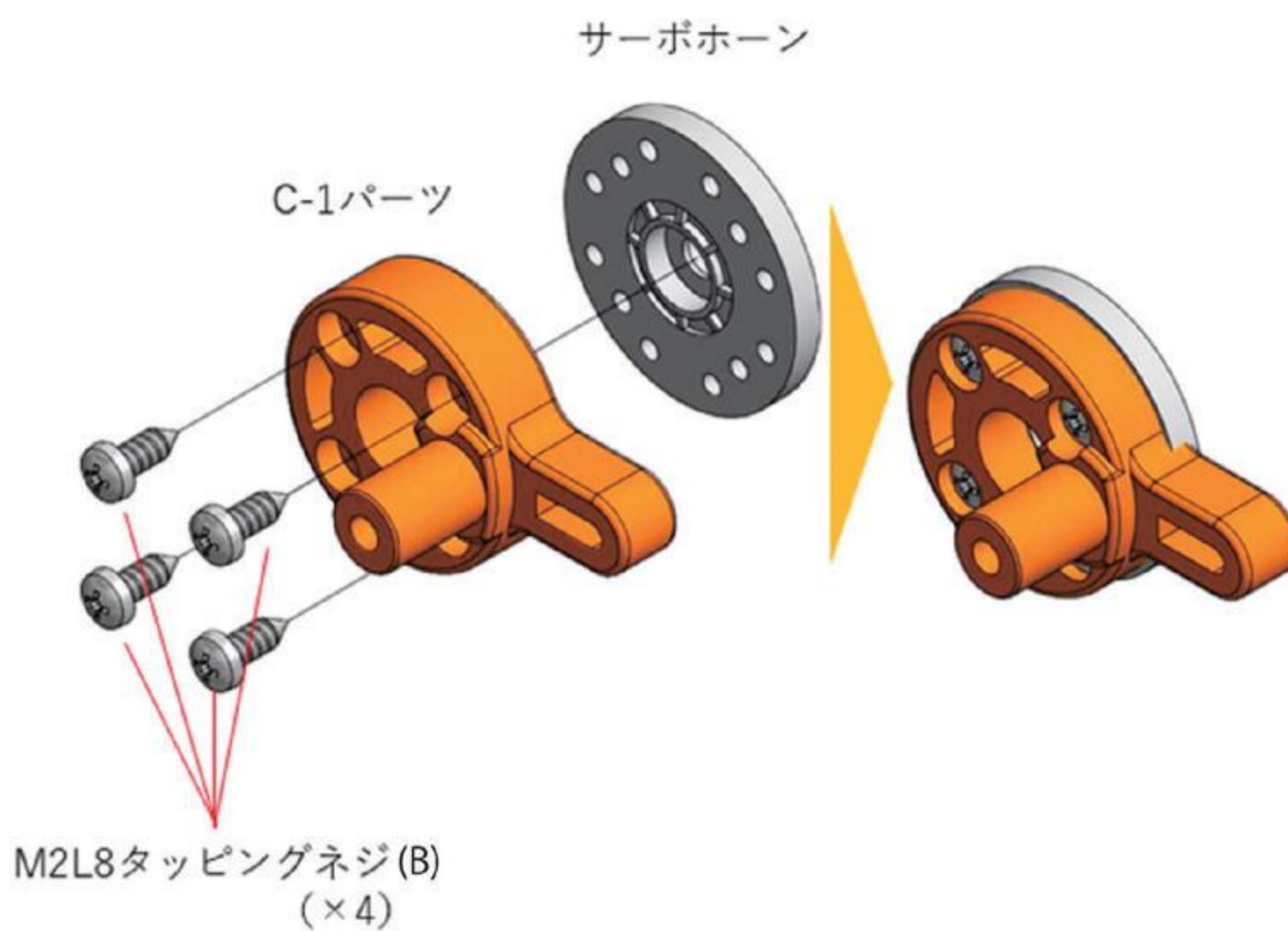


図1-9 C-1パーツ

<組み立て手順⑤>

図1-10がサーボホーンを組み付けたものです。ここまでの作業を終えたら、組み立てたものを一度整理してください。このあと、サーボモーターの初期設定（原点位置合わせ）を行い、組み終えたB-1パーツとC-1パーツをサーボモーターに組み付けます。



図1-10 サーボホーンを組み付け

1.2. サーボモーターの仕組み



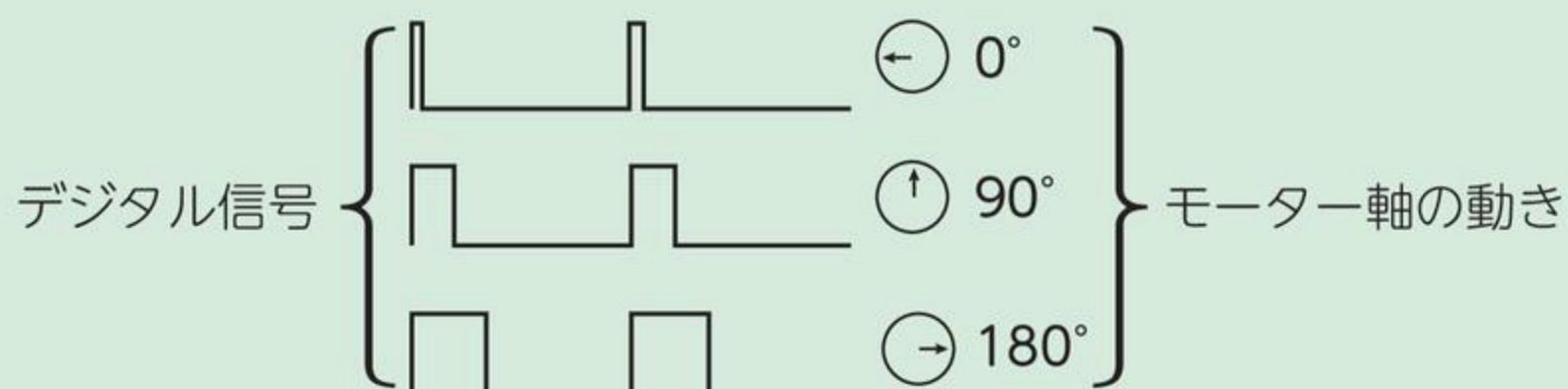
豆知識

サーボモーターの原点位置合わせについて

サーボモーターには、モーター内部にサーボ機構（物体の位置、方位、姿勢などの目標値を追従する制御系）があり、現在の位置を目標の位置と比較してその差を追従するよう制御するフィードバック制御を行っています。そのため、基準となる位置（原点）が必要になります。

たとえば、アナログ時計の電池交換をしたあとに、時計の針を正しい時刻に合わせないと、間違った時刻を示し続けるのと同じように、サーボモーターの原点位置合わせはロボットに命令通りの動作をさせるために必要な作業なのです。

サーボモーターは、図のようにマイコンからのデジタル信号でモーター軸が入力した角度に回転します。なお、二足歩行ロボットでは、原点の角度を90度に設定しています。

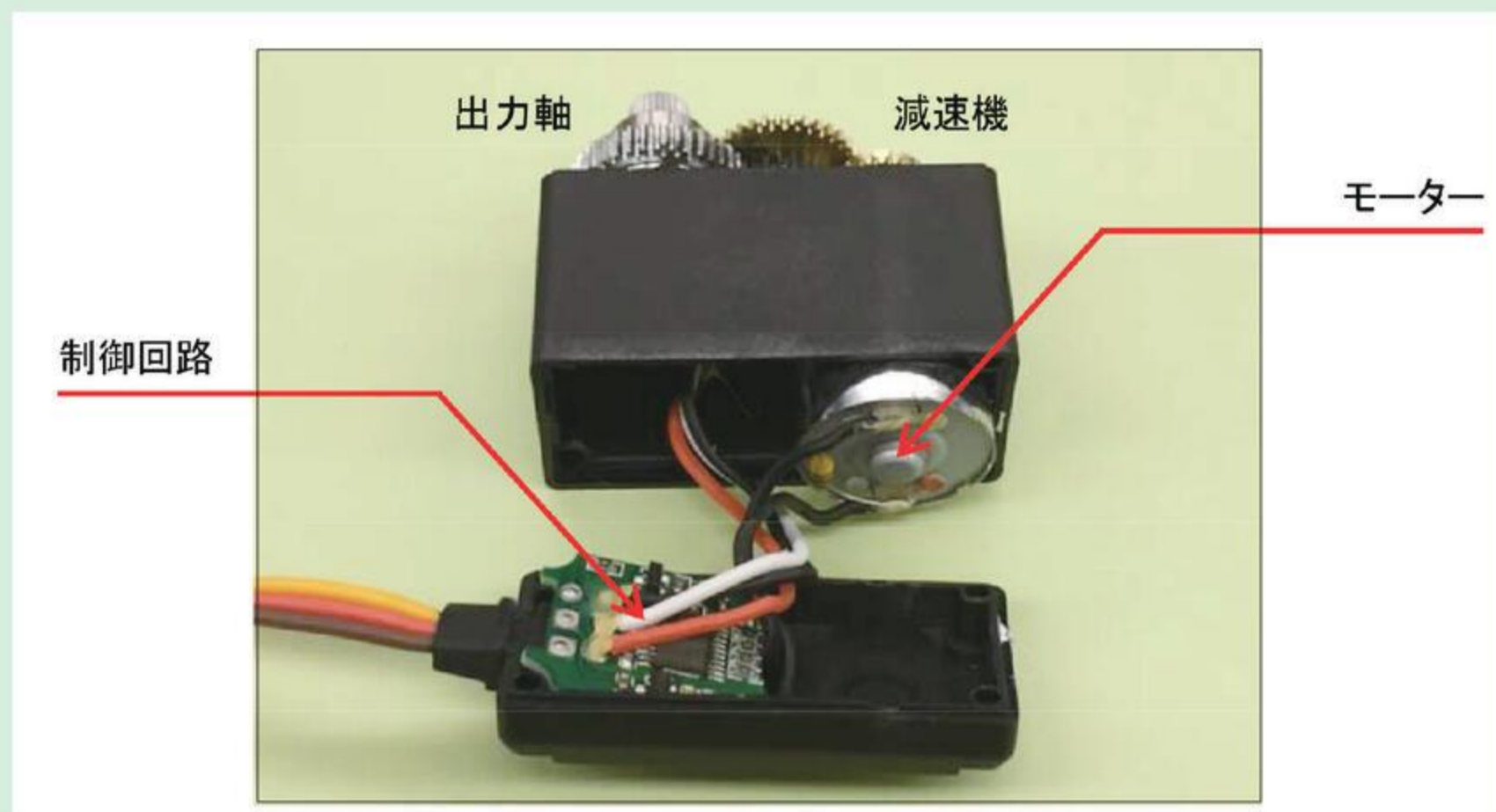
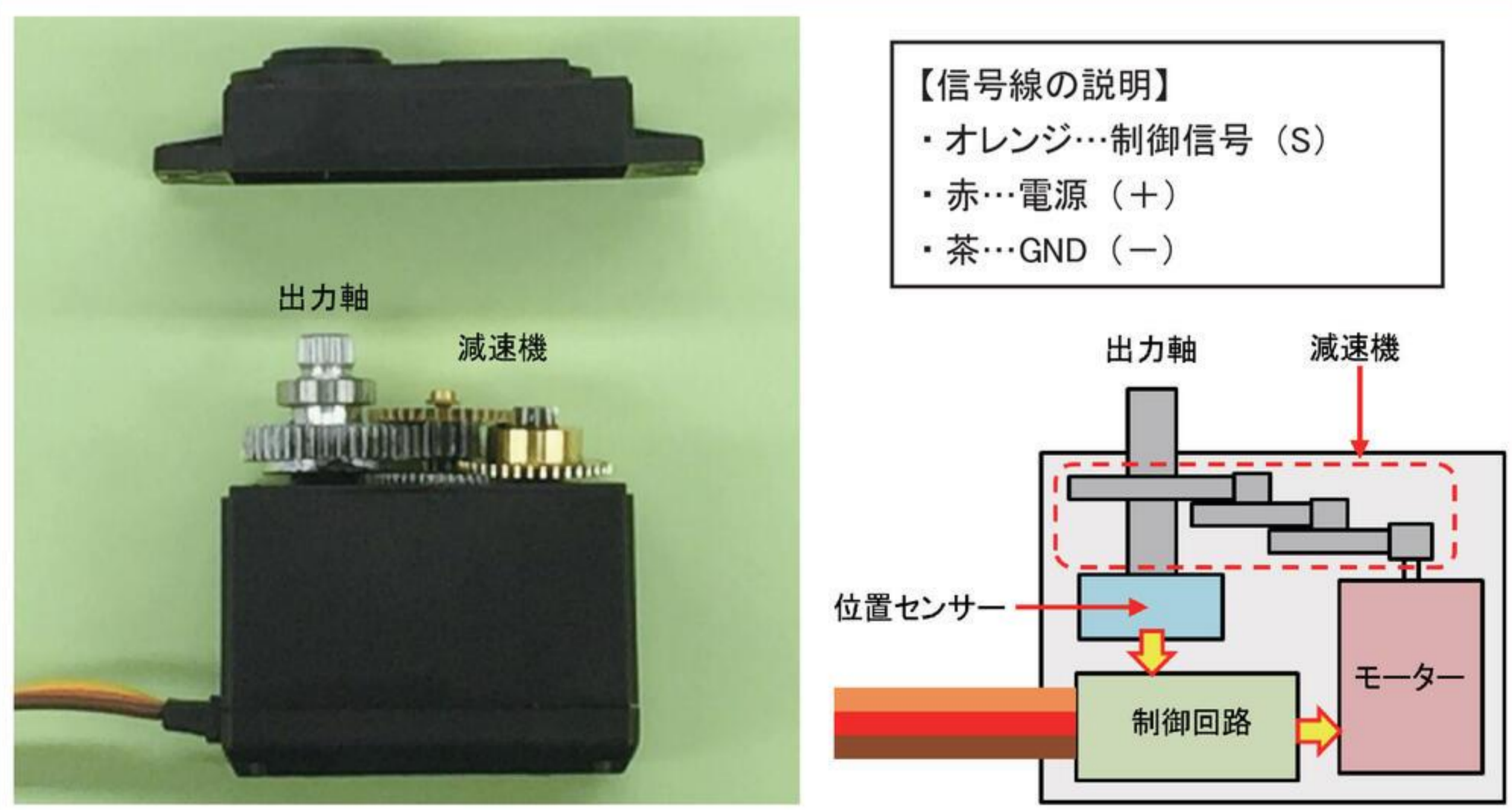




豆知識

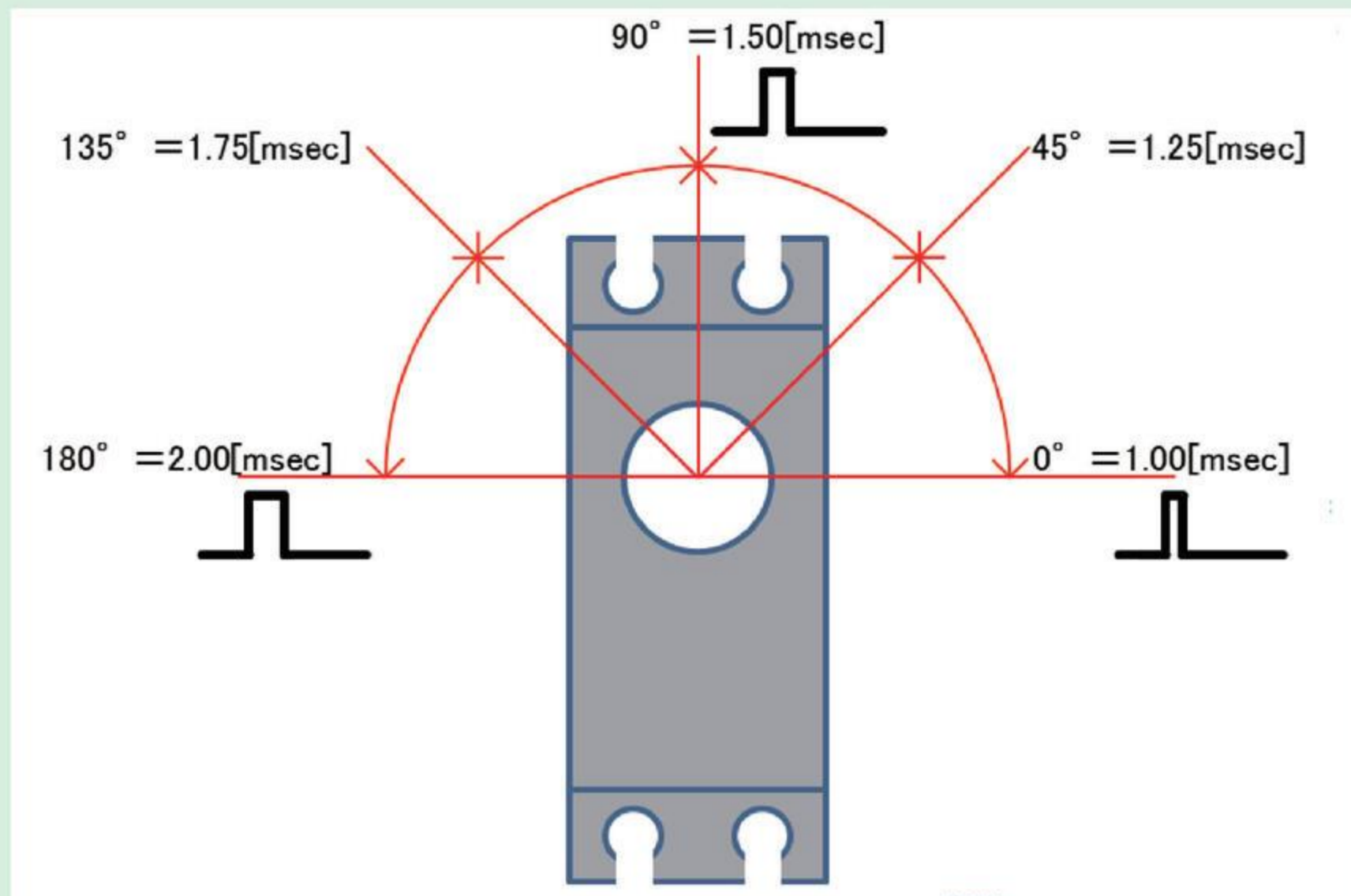
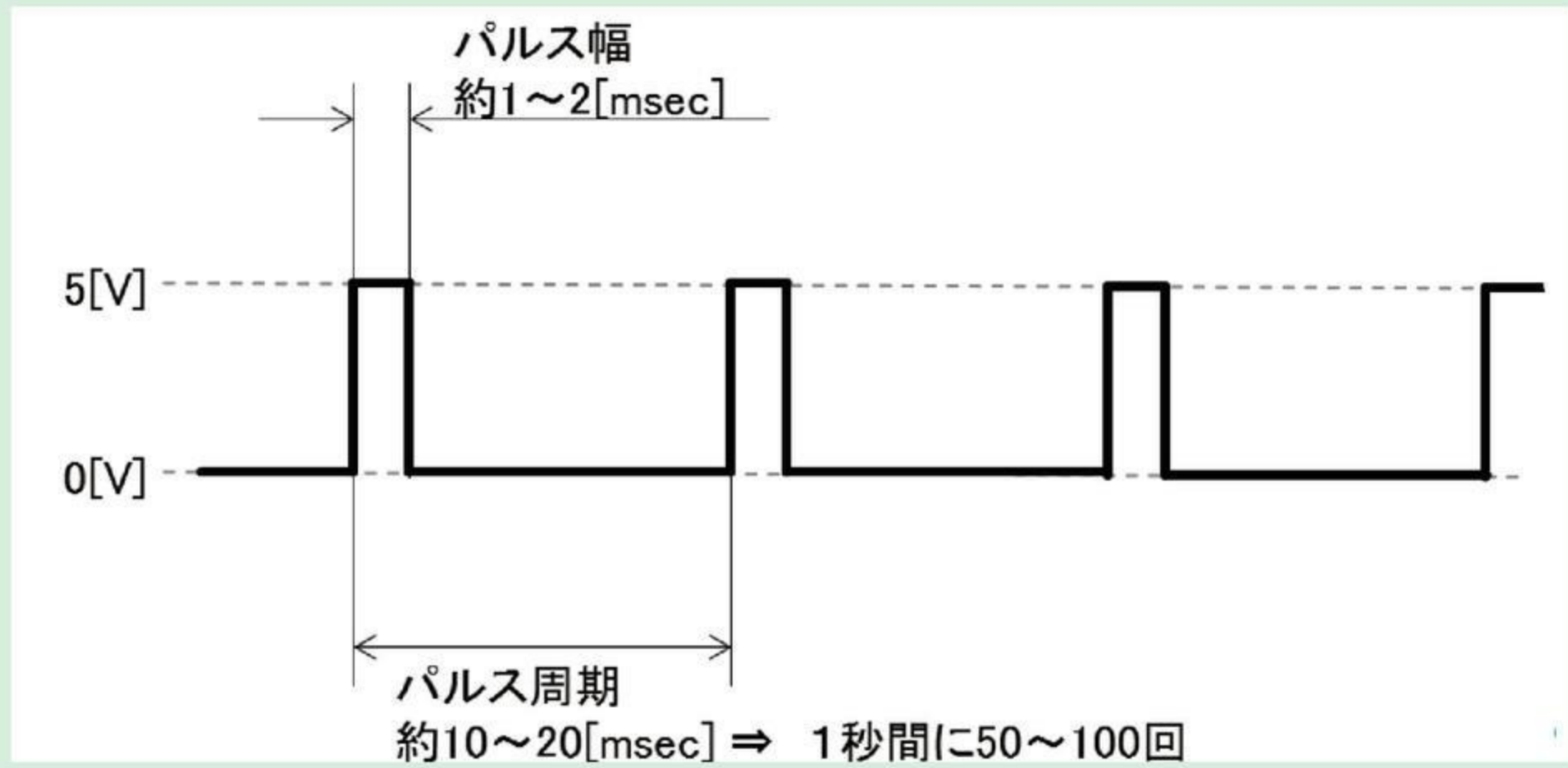
サーボモーターの中身を見ると、図のような仕組みになっています。図のように、サーボモーターには、モーターと減速機が組み込まれていて、出力軸の現在位置が角度センサーで検出されます。出力軸の目標角度は、サーボモーターの3本の信号線のうちのオレンジ色の信号線からデジタル信号のパルス幅としてマイコンに入力します。サーボモーター内部の制御回路は、現在の出力軸の角度と、パルス信号で入力された目標角度の差を無くすように、出力軸を動かし、常に出力軸は目標位置を追従するように動きます。また、パルス信号の幅を変化させることで、このサーボモーターの出力軸は0～180°まで回転します。

なお、実際の可動範囲や速度はサーボモーターの仕様により異なります。ですから、ロボットの開発者は、オリジナルのロボットを設計して製作するときは事前にサーボモーターの仕様を確認しておく必要があるということです。





豆知識



1.3. サーボモーターの原点位置合わせ

それでは、サーボモーターの原点位置合わせを行いましょ。二足歩行ロボットを正しく歩行させるために大事な工程になりますので、しっかりと行いませう！

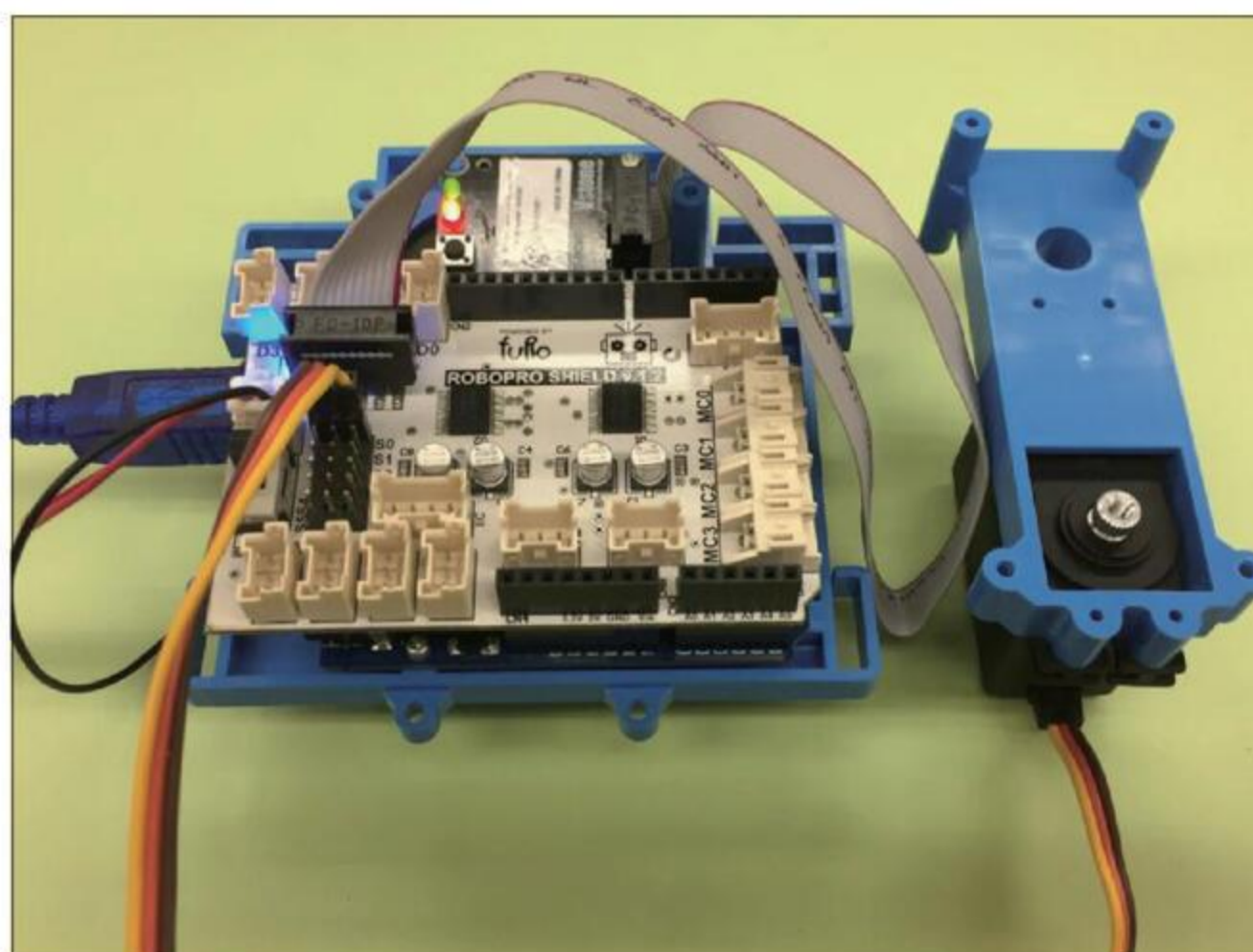
さきほどの作業で組み上げたサーボモーターと、マイコンボードを取り付けたボディ部分（A-1パーツ）を準備して、以下の手順を行ってください。



POINT

原点位置合わせの手順

- ・電池ボックスに単三電池6本をセットして電源を入れる。
- ・コントローラーと無線受信モジュールをペアリングする。
- ・ロボプロシールドの **S0** 端子にサーボモーターのコネクターを接続する（**図1-11**参照）。
- ・USBケーブルをパソコンと接続して、プログラムを書き込む。



ロボプロシールドのGV5の印刷を確認しながらコネクターを接続してください。

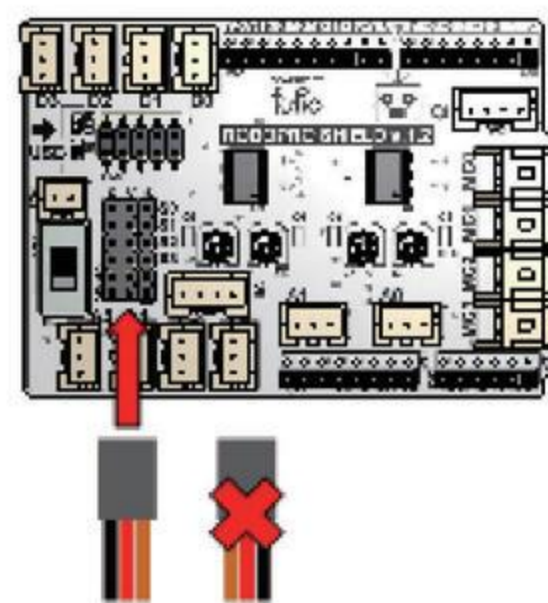


図1-11 サーボモーターのコネクターの接続

プログラムの書き込み

RoboticsProfessorCourse3 > BiRobot1 > ServoOrigin0

実行結果：サーボモーターの出力軸が原点位置に動く。

原点位置の角度はサーボモーターの可動域（0～180度）の中間位置90度に設定されています。使用する3個とも原点位置合わせを行ってください。

なお、コントローラーの右アナログスティックを上下に動かすと、アナログスティックの上下の動きに合わせてサーボモーターの出力軸が0～180度の範囲で動き、アナログスティックを離すと出力軸は原点に戻ります。可動域も簡単に確認しておきませう。

<組み立て手順①>

使用する3個のサーボモーターの原点位置調整をしたら、**図1-12**のように、B-1パーツをサーボモーターの出力軸に嵌め込み、サーボモーター付属パーツに入っている M3L6ネジ で図のように組み付けてください。

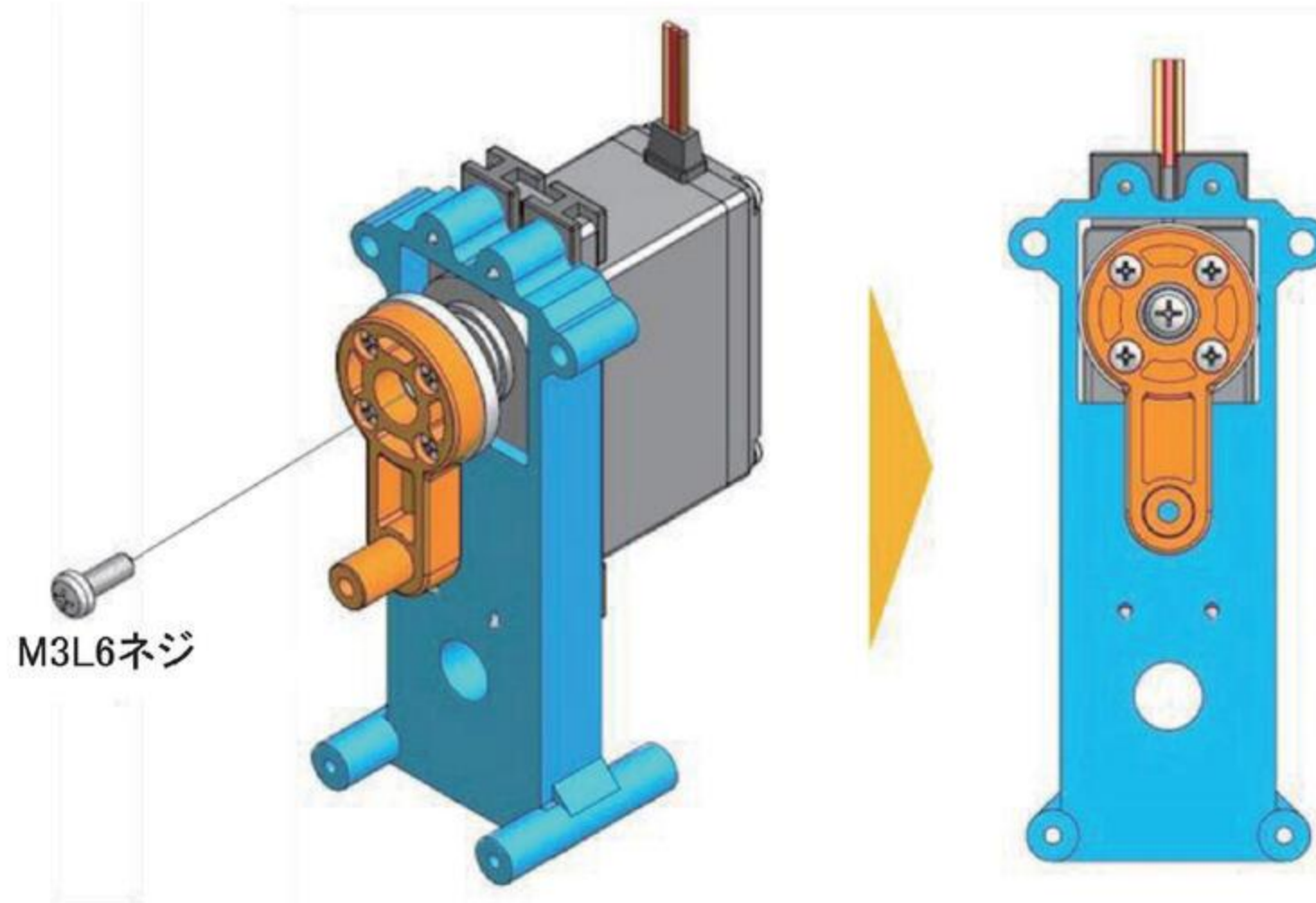


図1-12 B-1パーツとサーボモーターの組み立て

<組み立て手順②>

C-1パーツをサーボモーターの出力軸に嵌め込み、**図1-13**のように M3L6ネジ で出力軸に組み付けてください。

C-1パーツは、同じものを2セット組み立てて下さい。

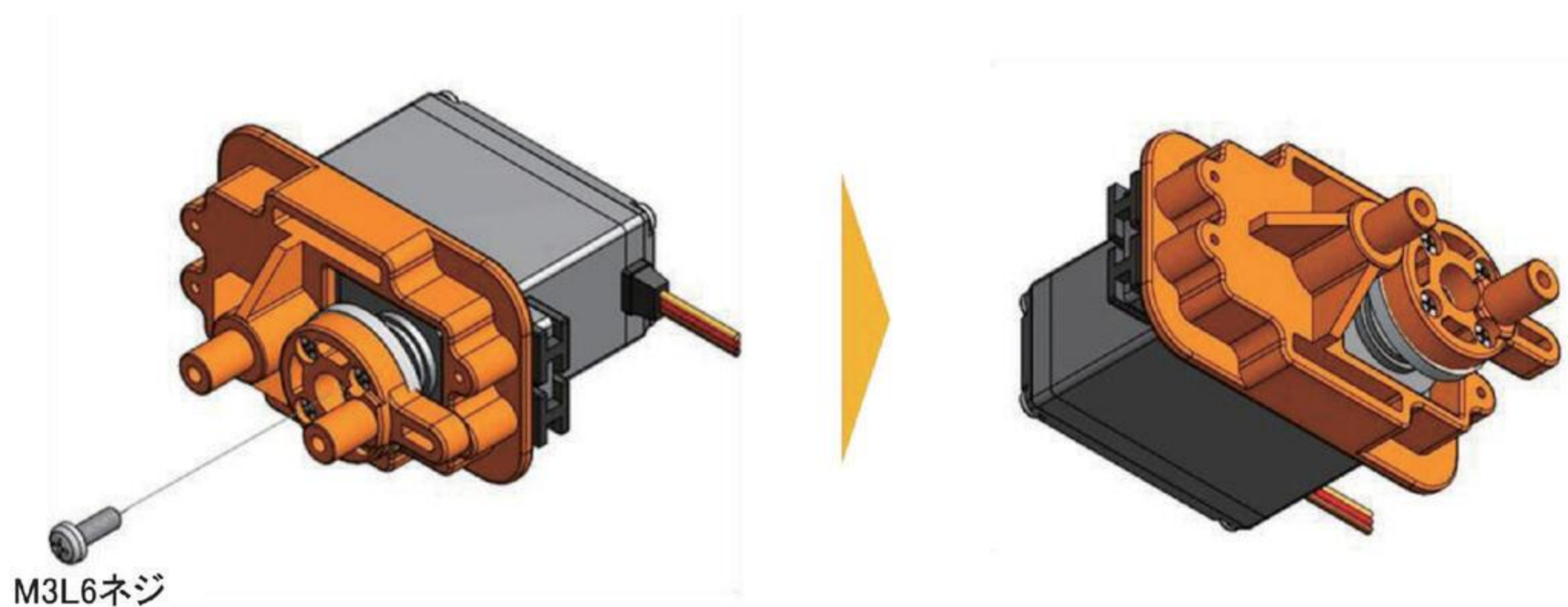


図1-13 C-1パーツとサーボモーターの組み立て

2. サーボモーターのプログラム（目安 40 分）

2.0. サーボモーターへの命令をシリアルモニターで確認する

サーボモーターへのパーツ取り付けを終えたら次に、サーボモーターの動作を確認しましょう。今度は、出力軸を動かすと同時に、シリアルモニターにコントローラーからサーボモーターへの指令値を表示させましょう。

1) 手順 1

USBケーブルをマイコンボードとパソコンに接続します。ロボプロシールドの [S0 端子] にサーボモーターのコネクターを接続して、電源スイッチをONにし、コントローラーをペアリングしてください。

2) 手順 2

以下のプログラムを書き込みましょう。

∞ プログラムの書き込み

RoboticsProfessorCourse3 > BiRobot1 > ServoOrigin0

3) 手順 3

プログラムを書き込んだら、画面右上にある虫メガネのアイコンをクリックして、シリアルモニターを開いてください。そして、「9600baud」に設定します。最初は、「0=>90」と表示されます。これは、初期値が原点の90度であることを表しています。

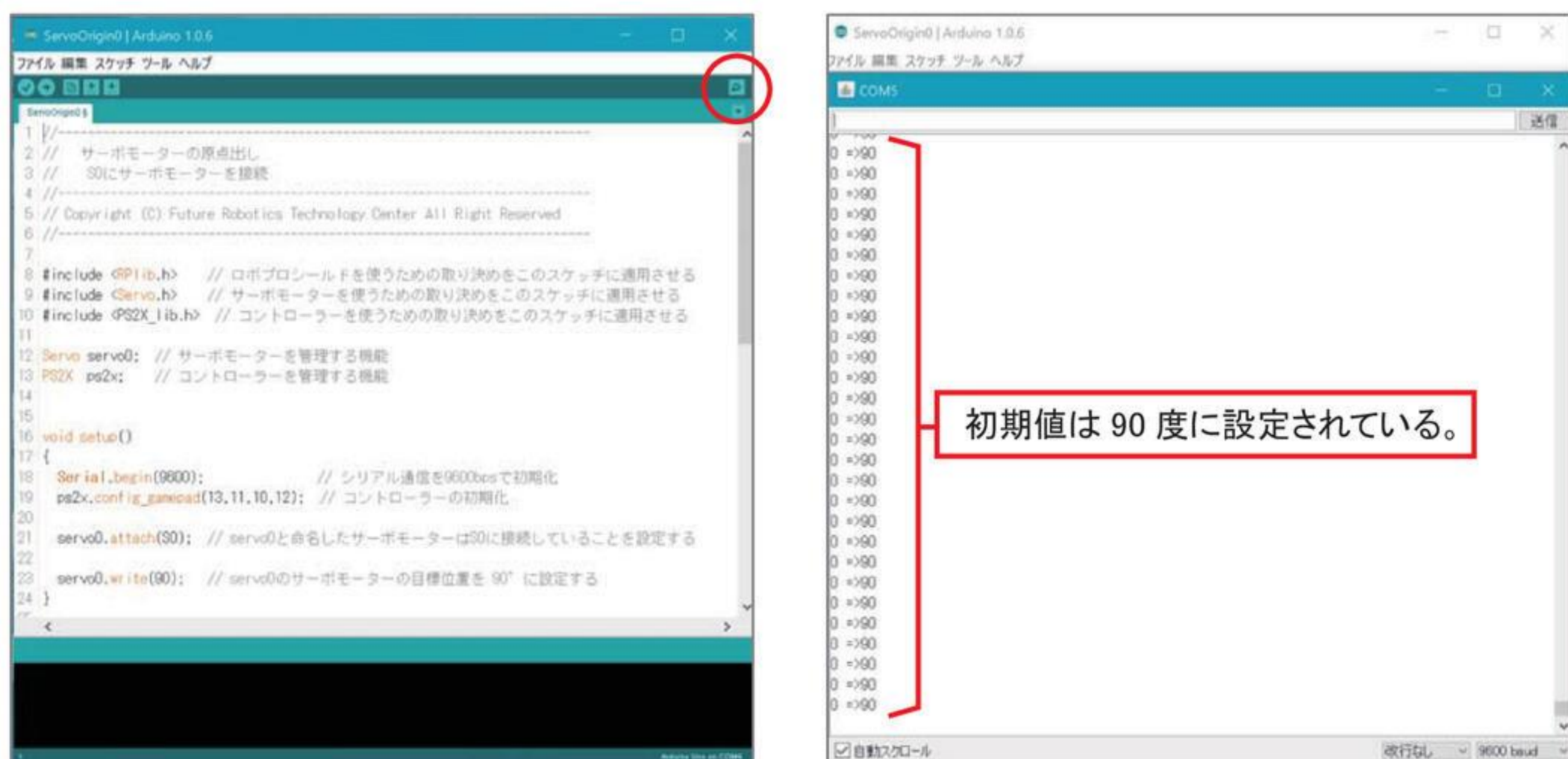


図2-0 シリアル通信

4) 手順 4

次に、シリアル通信をしている状態で、コントローラーの右アナログスティックを上下させて、値の変化を確認してみましょう。

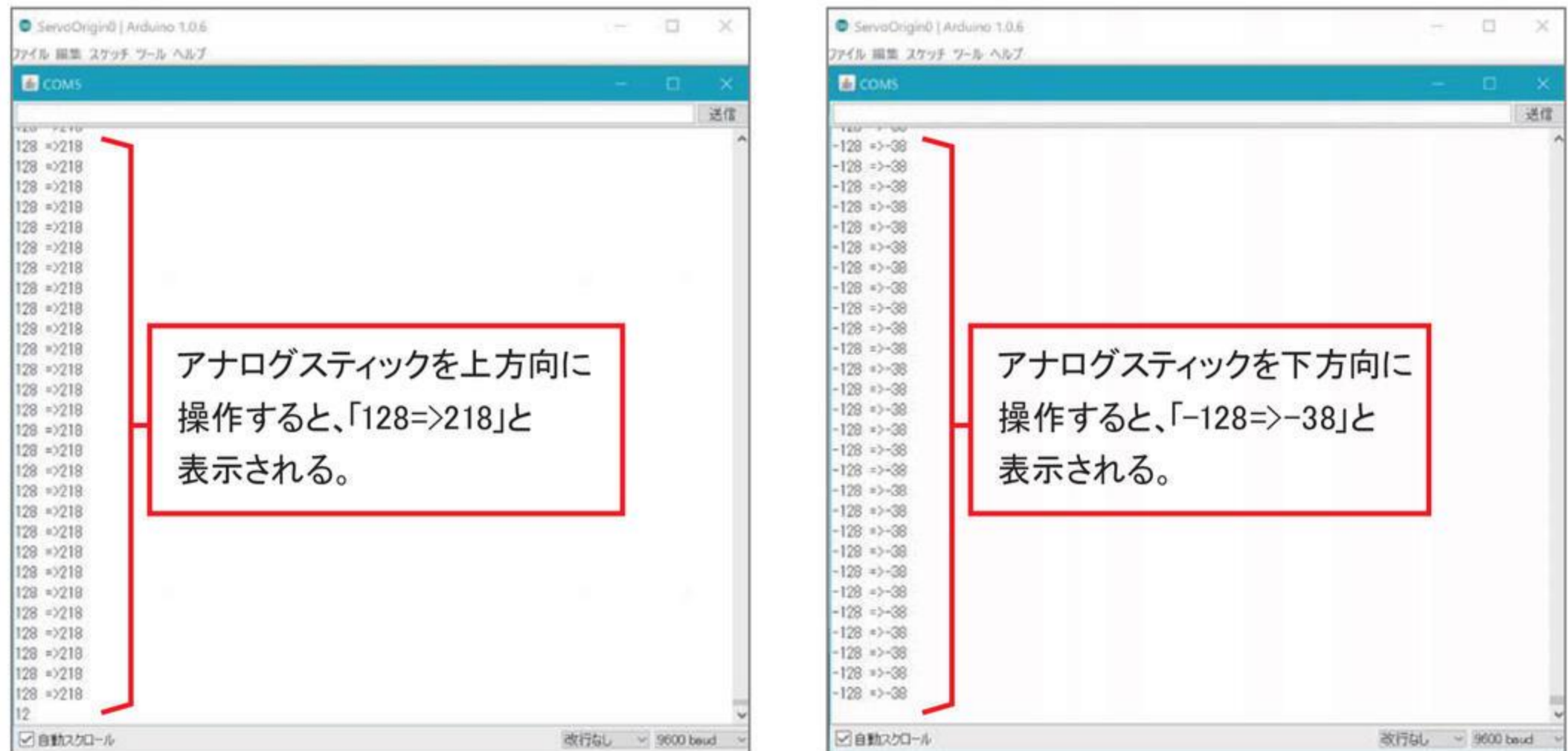


図2-1 シリアル通信（コントローラー操作時の遷移）

2.1. プログラム「ServoOrigin0」の確認

では、プログラム「ServoOrigin0」を確認してみましょう。

コントローラーの右アナログスティックから取得したアナログ値 rx を、サーボモーターの原点となる90度に加えて、`servo0.write(pos0);`に反映させて出力軸^{じく}を動かしています。また、シリアルモニターに右スティックのアナログ値と、角度を表示させています。

□ プログラム「ServoOrigin0」より抜粋 ばっすい

```
pos0 = 90 + rx;           //右スティックの上下方向真中の位置を90にした角度
                          //司令値に変換
                          //-38 ~ 218の値になる

servo0.write(pos0);      //角度司令値をservo0に反映

Serial.print(rx, DEC);   //右スティックのたて位置を表示
Serial.print(" =>");
Serial.print(pos0, DEC); //角度指令値を表示
Serial.print("¥r¥n");
```

2.2. Servo ライブラリを見てみよう

プログラム「ServoOrigin0」では、アナログスティックの値に90を加えて範囲を-38～+218に変換していました。

しかし、はじめに説明したとおり、サーボモーターの出力軸は0度～180度の間でしか動かせませんから、値が-38や+218になってしまうと困ってしまいます。

もし角度指令値が可動域を超える値になってしまった場合、エラーを防ぐ処理が必要です。しかも、その処理は今回に限らず、サーボモーターに命令をするようなプログラムをつくるときには常に必要ですよ。

こういった「このパーツを使うときには大抵この処理が必要」という命令は、各プログラム内にいちいちかくのではなく「ライブラリファイル」を活用します。

ライブラリは2年目でも学びましたが、各種パーツを使うとき、それぞれのパーツに関する命令をまとめてかいておくプログラムです。

サーボモーターにまつわる様々な命令は「Servo.h」「Servo.cpp」という2つのライブラリファイルにかかれていますので、新たにプログラムをつくるときは「今回はサーボモーターを動かすので、このライブラリファイルにかかっている各種命令を使う（引用する）よ」という宣言をかいておくだけで済むのです。

□ プログラム「ServoOrigin0」より抜粋

```
#include <Servo.h>
```

では、ライブラリファイル内にかかっている、「角度指令値が可動域を超えている場合」の命令を見てみましょう。

□ ライブラリファイル「Servo.cpp」より抜粋

```
void Servo::write(int value)
{
    if(value < MIN_PULSE_WIDTH)
    {
        if(value < 0) value = 0;
        if(value > 180) value = 180;
    }
}
```

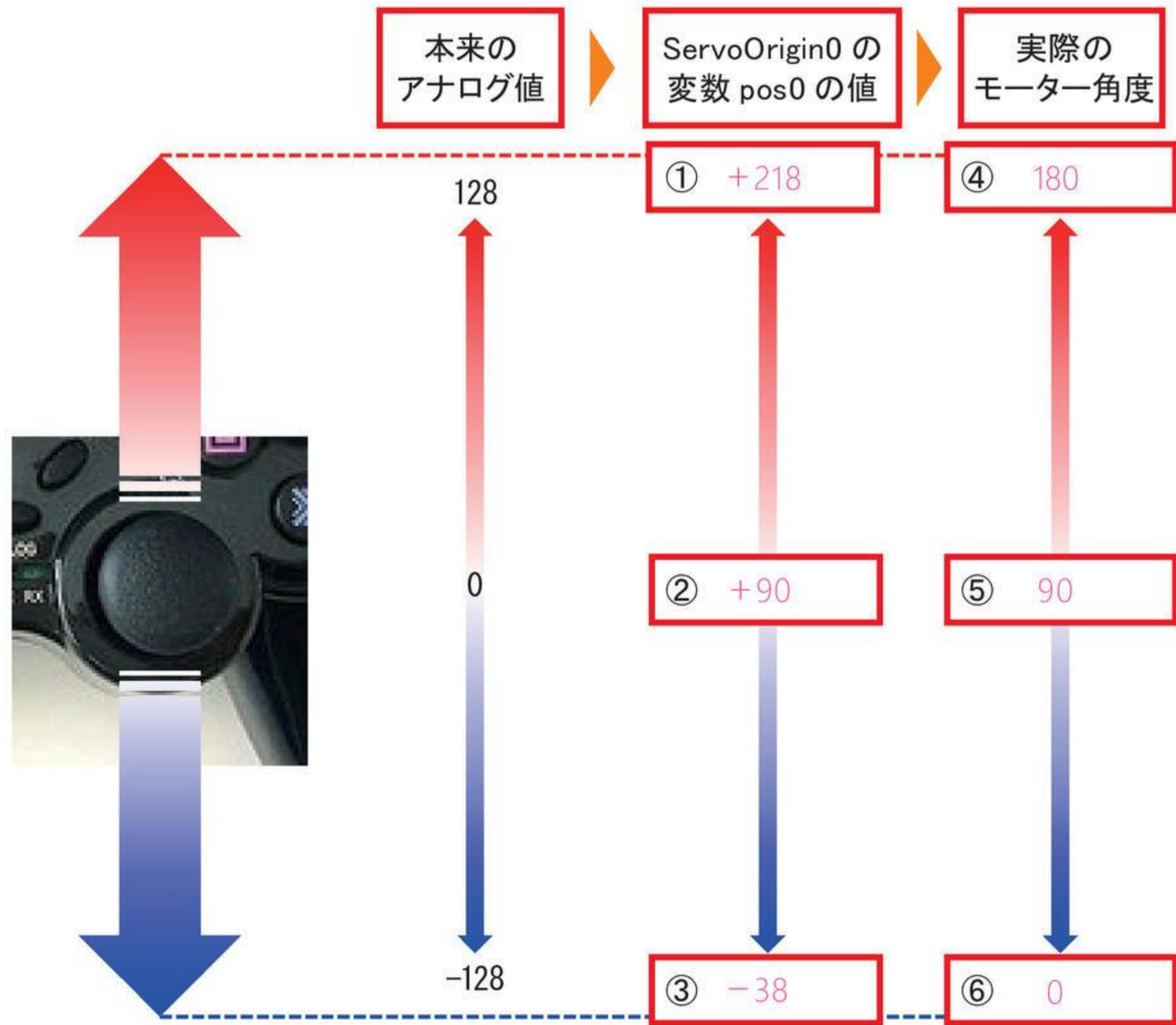
関数 `write()` の内容を説明している部分です。

緑色の部分は、引数 `value` が0未満のときは0に、180より大きいときは180に、それぞれ無理やり変換するような命令になっていますね。こういった基本的な処理は、ライブラリファイルにかいておく方がその後のプログラミングで何度も活用できるため便利です。

やってみよう！

これまでの流れを改めて読み取り、アナログスティックの値の範囲^{はんい} (-128 ~ +128) がどのように変換されていったか確認しよう！

下の図の①~⑥に、当てはまる数値を記入してね。



-38 ~ +218の範囲^{はんい}を0 ~ 180に変換するときは、正確には図2-2のようになっています。

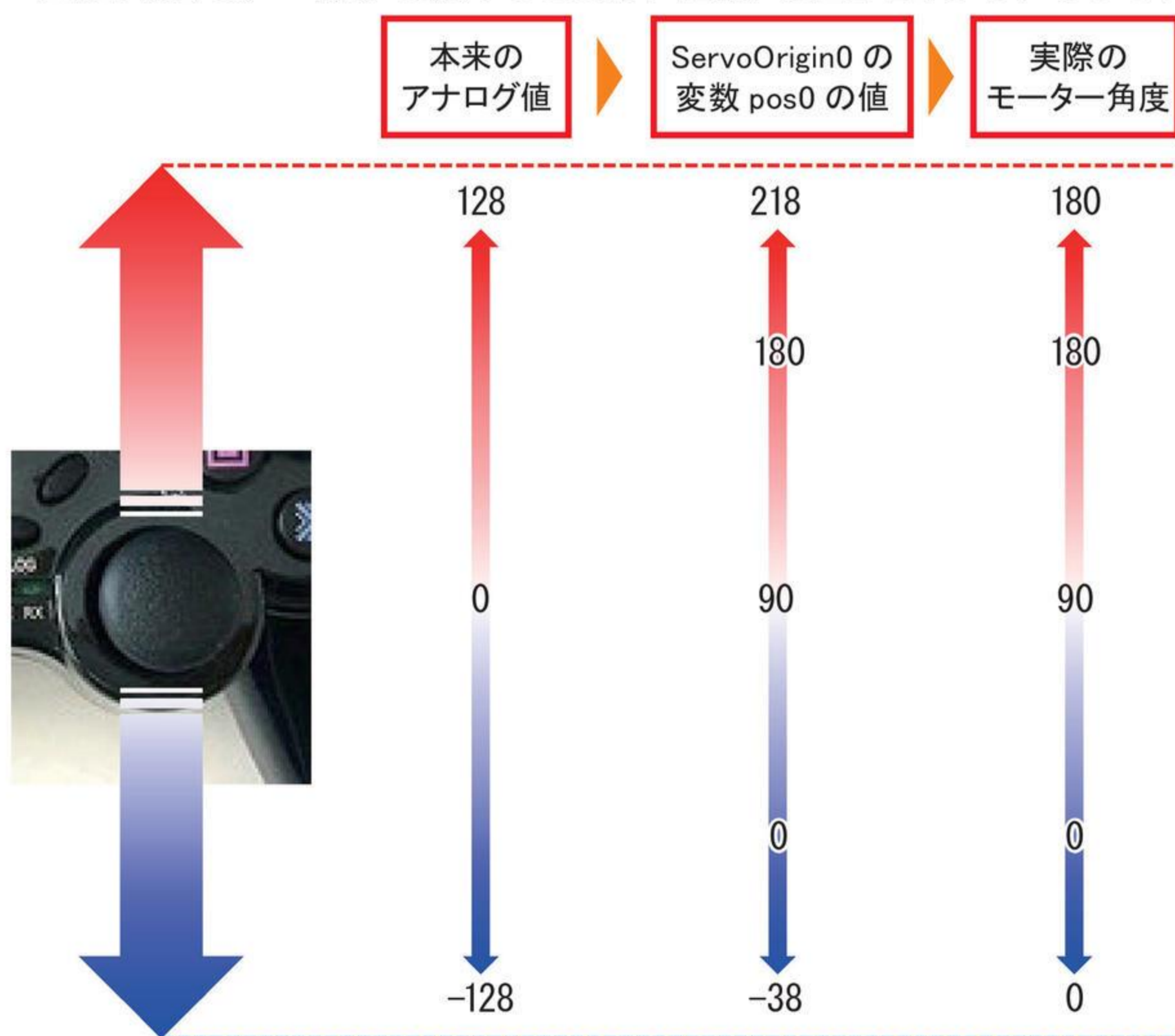


図2-2 アナログスティックの値を角度指令値に変換する

アナログ値が90以上であれば、角度指令値は180になるというわけです。

さて、ここまでサーボモーターの制御について確認してきましたが、プログラムに引数^{ひきすう}として用いていたのは「角度」でした。つまり、出力軸^{じく}を90度に回転させたいならば、`servo0.write(90);`としていたわけです。

しかし前のページでも学習したように、サーボモーターの制御は「パルス幅」^{へんこう}を変更して行うのでした。ですから、出力軸^{じく}を90度に回転させたいならば、サーボモーターには「90度にして！」ではなく「1.5ミリ秒の幅のパルス信号を出して！」という命令を出す必要があります。「角度」で出した命令も、内部では「時間」に変換しなければならないのですね。

その処理^{しより}も、ライブラリファイルにかかれています。

□ 「Servo.cpp」より抜粋^{ぼつすい}

```
value = map(value, 0, 180, SERVO_MIN(), SERVO_MAX());
```

`map`はこれまでも何度か扱いましたね。今回は「0度～180度」^{はんい}の範囲で出した「角度」の命令を、`SERVO_MIN() ~ SERVO_MAX()`^{はんい}の範囲に変換するような処理^{しより}になっています。これは「パルス幅」の値で、「ServoOrigin0」ではパルス幅の最小値`SERVO_MIN()`が544マイクロ秒（0.544ミリ秒）、最大値`SERVO_MAX()`が2400マイクロ秒（2.4ミリ秒）で固定されています。

ただ、パルス幅の最大値・最小値を上限・下限ぎりぎりにしていると、これ以上はモーターが動けないという物理的限界（メカリミット）の位置で停止します。つまりモーターがフルパワーで動いたまま無理やり静止させられていることになり、内部が発熱したりする可能性がある大変危険な状態です。これを避けるため、パルス幅を限界値から少しだけずらしておきましょう。

2.3. サーボモーターの暴走を防ぐ

まずは、以下のプログラムを実行してください。

このプログラムは、ほとんどプログラム「ServoOrigin0」と同じですが、^{ゆいいつ}唯一、サーボモーターの初期化部分が異なります。

∞プログラムの書き込み

RoboticsProfessorCourse3 > BiRobot1 > ServoOrigin1

□プログラム「ServoOrigin1」より^{ぼっすい}抜粋

```
//servo0と名づけたモーターをS0に接続  
//パルス幅の上限と下限を設定する  
servo0.attach(S0, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH - 0);
```

□ライブラリファイル「Servo.h」より^{ぼっすい}抜粋

```
#define MIN_PULSE_WIDTH      544  
#define MAX_PULSE_WIDTH     2400
```

関数`attach();`は、^{ひきすう}引数が1つであれば「モーターの接続ソケット」だけを、^{ひきすう}引数が3つであれば「モーターの接続ソケット・パルス幅の最小値・最大値」を、それぞれ指定できます。つまり「ServoOrigin1」では、パルス幅の最小値は544マイクロ秒、最大値は2400マイクロ秒と指定されていることになります。

このままでは最小値・最大値とも「ServoOrigin0」とまったく変わらないので、プログラムをかきかえて最大値を下げていきましょう。

ステップアップ

プログラム「ServoOrigin1」の黄色部分をかきかえ、パルス幅の限界値を好きな値に調整してみよう。サーボモーターの回転角度に変化はあるかな？

💡ヒント

サーボモーターの仕様上、パルス幅は4の倍数になるようにしたほうが都合がいいよ！

講

大幅に数を下げると、アナログスティックを倒したときの可動域が目に見えて縮小したり、原点位置が当初と明らかにずれたりといった影響が出てしまいます。`MAX_PULSE_WIDTH - 12`など、初期値から12あるいは16程度ずれた値にすると原点位置や可動範囲への影響を最小限にしつつ、暴走や加熱を抑えることができます。

3. まとめ（目安5分）

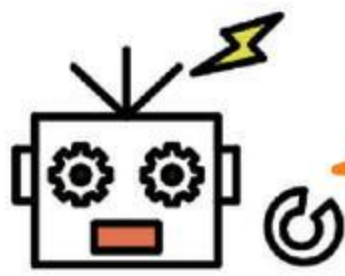
今回は、二足歩行ロボットの組み立ての一部を進めて、サーボモーターの仕組みと、プログラムを、ライブラリの中まで踏み込んで勉強しました。

ライブラリはとても便利で、人のかいたサンプルコードを流用することができます。ライブラリの中にまで踏み込んで見ていくと、説明文にはかかれてない細かなことまで見えてきます（英文で解読も大変ですが）。

プログラミングを勉強しはじめたばかりのときは、なかなかライブラリの中まで読むということは難易度が高くて難しいと思いますが、慣れてくると、いちいちコメント文をみるよりもプログラムの中身を見た方が早いということもあります。

ここでは、ライブラリは中を覗いて詳しく制御方法を調べられるということ、そして、どのように調べていくのか、その手順だけでも感じてもらえると、将来プログラミングスキルが上達したときに、「あのとき言っていたことはこの事だったのか！」と、納得できる瞬間がくるでしょう。

二足歩行ロボットの組み立ては、まだまだ序盤です。次回に備えて、パーツをなくさないように綺麗に整理整頓してしまっておきましょう。



かく隠していた、プログラムのウラ側を知ることがデキタネ！

《次回必要なもの》

今回は、残りの二足歩行ロボット用パーツを持ってきてください。

講

第1回の理解度を確認してください。

- ・サーボモーターの仕組みを理解する
- ・サーボモーターの初期設定を終える
- ・サーボモーターのライブラリをしてみる

今回は、六脚ロボットでも学習したサーボモーターの仕組みと制御方法、プログラムの復習をメインにしました。二足歩行ロボットでは、モーターとマトリクスLEDなどのハードを使用します。

今回は、組み立てと使用するハードの学習をします。