

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

アームロボット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

アームロボット③

第5回

センサーを使ったカシコイロボット

講師用

目 次

0. センサーを使ったカシコイロボット

0.0. 「センサーを使ったカシコイロボット」でやること

0.1. 必要なもの

0.2. 組み立て前準備

1. センサーアームロボットの組み立て

1.0. 超音波距離センサーのおさらい

1.1. 超音波距離センサーの取り付け

1.2. 超音波距離センサーの動作確認

2. 超音波距離センサーを使ったアームロボット

2.0. 手から逃げるロボット

2.1. プログラムの改造

2.2. 超音波距離センサーの組みかえ

2.3. 自動搬送の機能の確認

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。

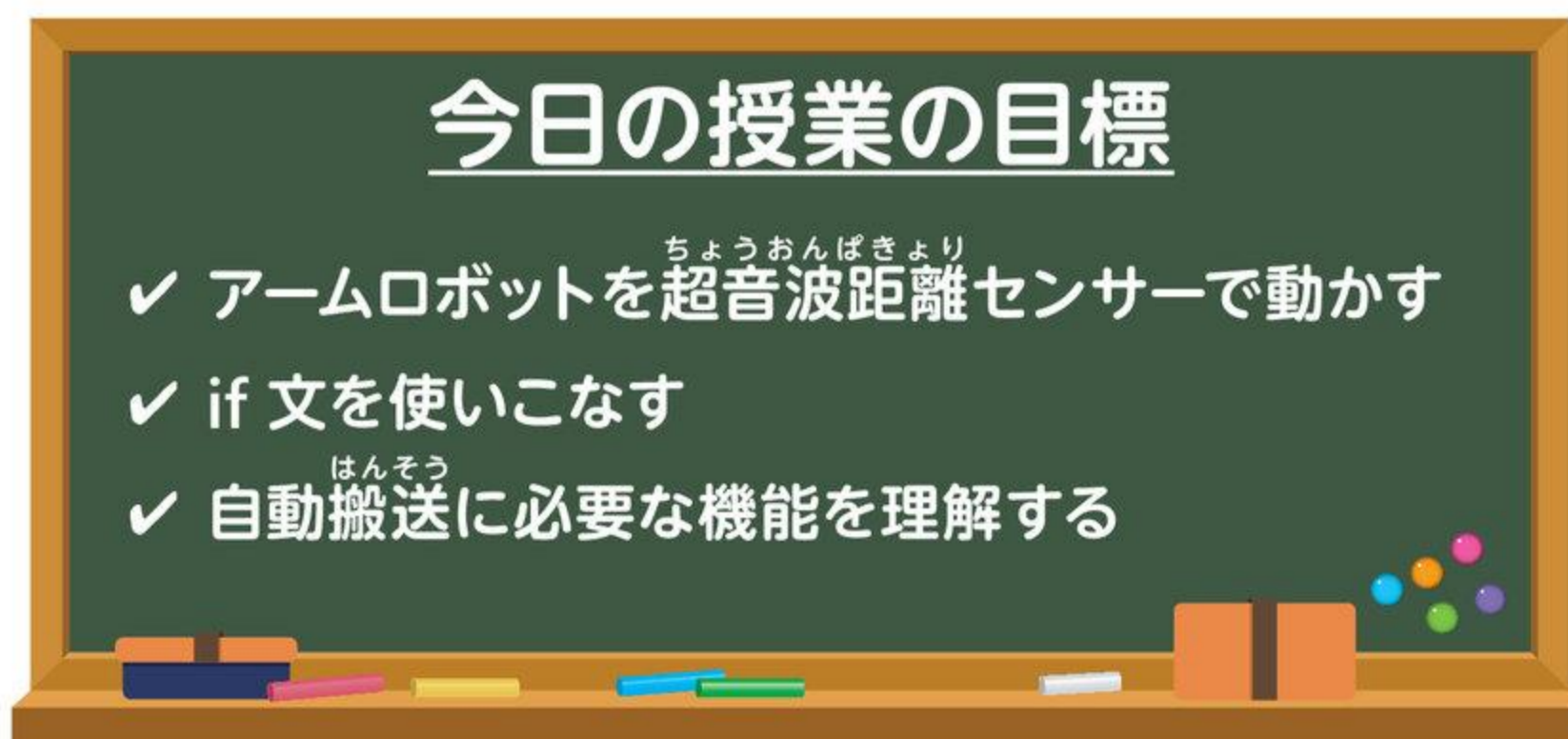
(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

○ アームロボットにプログラムを入力するとすぐに動く場合がありますので、AC アダプタを
取り外し、生徒とアームロボットとは十分に距離を取るよう指導してください。

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. センサーを使ったカシコイロボット (目安 10分)

0.0. 「センサーを使ったカシコイロボット」でやること



今回は、ロボットらしく自動で作業をするようなセンサーロボットの第一歩を勉強していきましょう。

アームロボットの動かし方は、前回までにいろいろと勉強しました。

コントローラーで操作をしたり、座標を活用したプログラムで目標位置を定めたり、文字や記号をかかせたりしました。

限られたパーツをうまく利用して、動きの精度を上げるのはなかなか大変な作業だったかと思います。今回は、さらにカシコイロボットにするために、超音波距離センサーを組み合わせさせていただきます。

超音波距離センサーは、いままでの授業で何度か登場しましたが、その特性を思い出して、最終的には、自動で判断して作業をするロボットにしていきたいと思います！

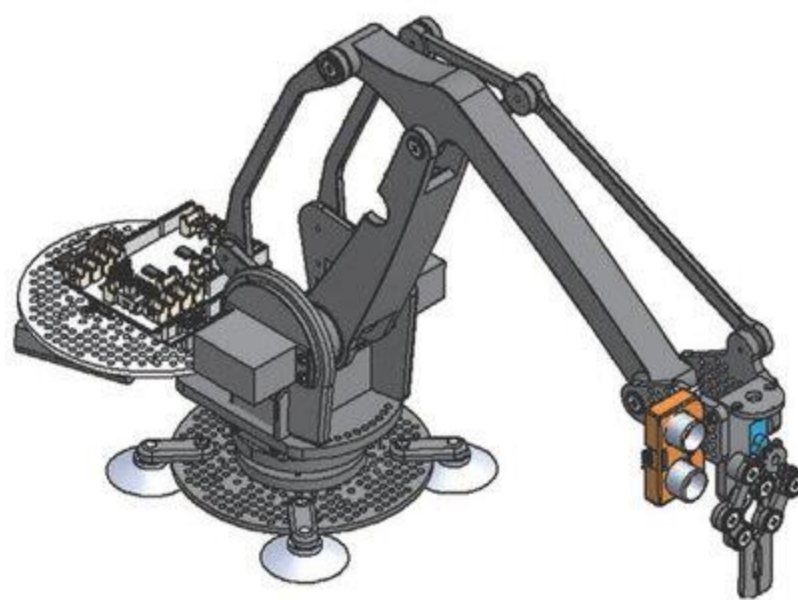
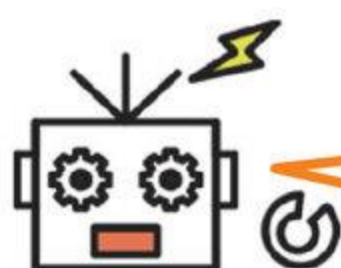


図 0-0 超音波距離センサーつきアームロボット



カシコイ、アームロボットにするぜ！ 生き物みたいにシヨウ！

0.1. 必要なもの

前回までに使っていたアームロボット本体とハンド部の他に、以下のパーツを用意しましょう。

なお、^{ちょうおんばきより}超音波距離センサーとセンサーL字ステイ、センサーカバーはあらかじめ M3L6 タッピングネジ (B) で組み立てておきましょう。

ラジオペンチ	1	ドライバー	1	USB ケーブル	1	センサーL字ステイ	1
							
マトリクス LED シールド	1	超音波距離センサー	1	センサーカバー	1	M3L6 タッピングネジ (B)	2
							
M3 ナット	2	600mm センサーケーブル	1	AC アダプター	1	M3L12 ネジ	2
							

図 0-1 必要なもの

講

超音波距離センサーは、あらかじめ1年目コースで使用したテストプログラムを実行することをお勧めします。その際は、マトリクス LED を使用します。

<テストプログラム>

RoboticsProfessorCourse1 > MagicItemB1 > USStest

0.2. 組み立て前準備

ハンド部を組みかえておきましょう。

まずは、第4回で使用した C-9 パーツ、C-10 パーツを C-1、C-2 パーツから取り外します。ネジの位置も矢印のように変更しましょう。

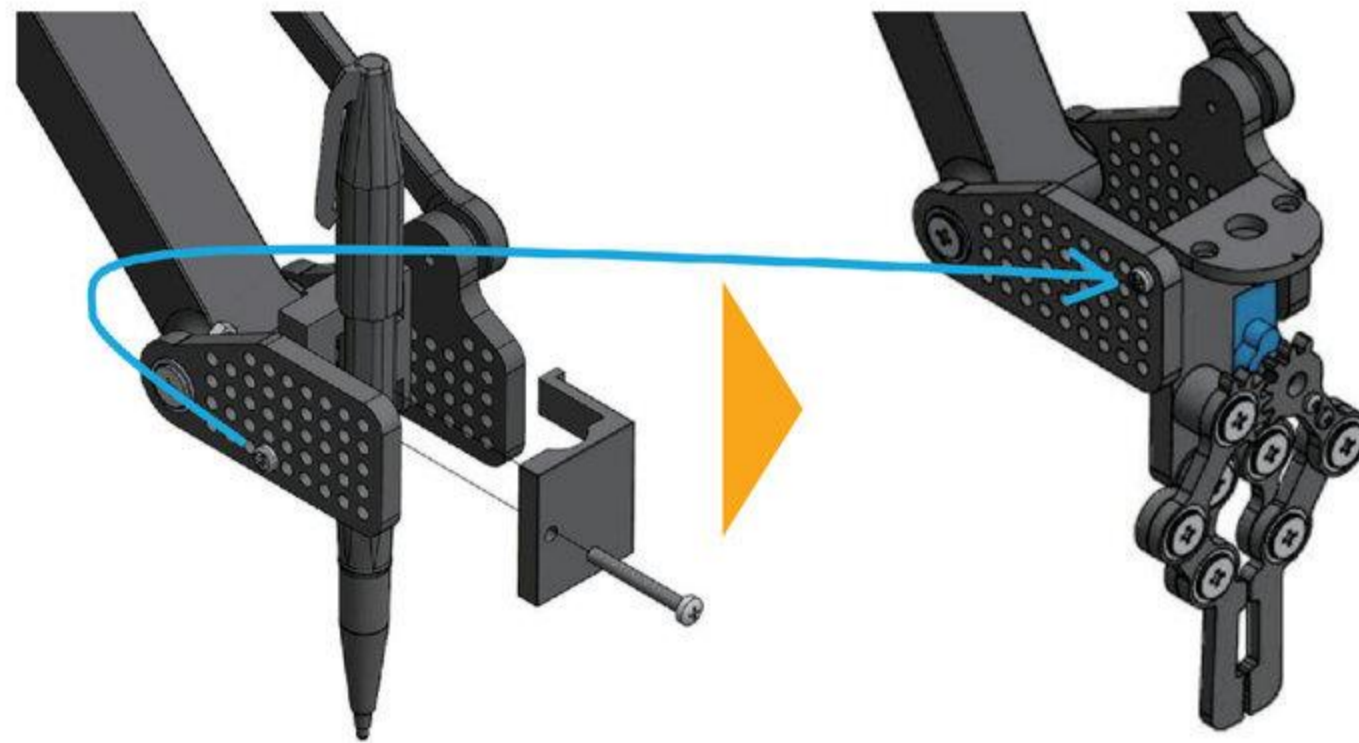


図 0-2 ハンド部の組みかえ

⚠ 注意！

動作中にアーム（リンク機構）の特異点とくいてんに入った時は、AC アダプターと USB ケーブルを抜いて電源を OFF の状態にしてください。特異点とくいてんに入ると動けなくなると、サーボモーターに物理的な負荷がかかり、マイコンにも大きな電流が流れ続けるため電氣的な負荷がかかります。

また、動作終了後も、サーボモーター内部から動作音が聞こえたり、静止している状態でガタガタと細かく振動したりするときは一旦電源を OFF にしましょう。このような状態で動かし続けると、マイコンボードやサーボモーターが発熱する可能性があるため、注意してください。

上記の他にも、何らかの障害物しょうがいぶつにアームがあたりサーボモーターが動き続けることや、長時間の連続動作は発熱はつえんや発煙の原因になります。動かさないときは、AC アダプターのジャック、USB ケーブルを小まめに抜きましょう。

1. センサーアームロボットの組み立て (目安 20 分)

1.0. ちょうおんぱきより 超音波距離センサーのおさらい

まずは、今回使用する^{ちょうおんぱきより}超音波距離センサーの性能を確認してみましょう。

図 1-0 は、^{ちょうおんぱきより}超音波距離センサーのメーカーでの性能テストの結果です。グラフと製品説明書からわかる^{ちょうおんぱきより}超音波距離センサーの性能を再度まとめます。

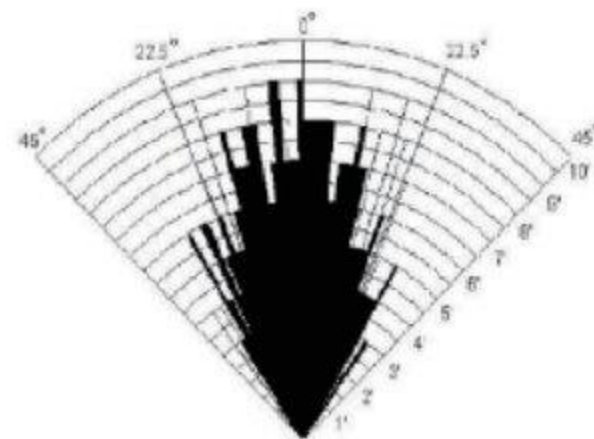
下記をふまえて学習を進めましょう。



POINT

測定範囲：センサーの中心（0度）から左右 15 度程度。

測定距離：^{きより}2cm ~ 400cm 程度。



Practical test of performance,
Best in 30 degree angle

図 1-0 ^{ちょうおんぱきより}超音波距離センサーの性能

1.1. ^{ちょうおんぱきより} 超音波距離センサーの取り付け

<組み立て手順①>

アームロボットのハンド部に^{ちょうおんぱきより}超音波距離センサーを取り付けます。

図の位置に M3L12 ネジ (× 2) と M3 ナット (× 2) で組み付けます。

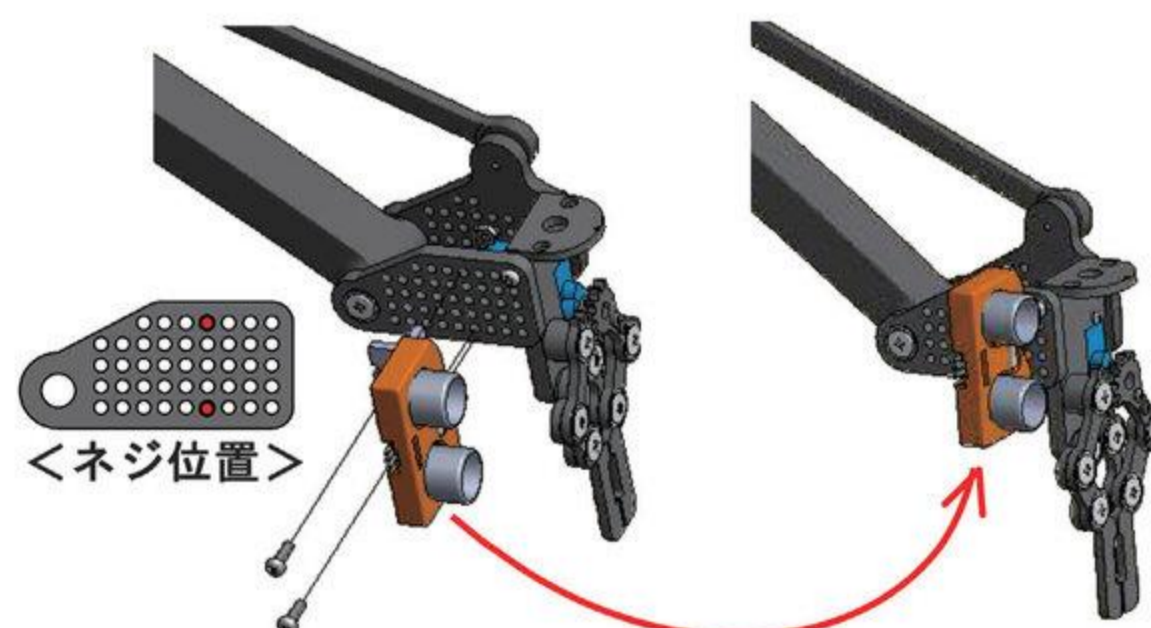


図 1-1 ^{ちょうおんぱきより} 超音波距離センサーの取り付け

<組み立て手順②>

アームロボットに取り付けてあるロボプロシールドに、マトリクスLEDシールドを取り付けます。マトリクスLEDシールドにマトリクスLEDや7セグメントLEDは取り付けませんので、注意しましょう。

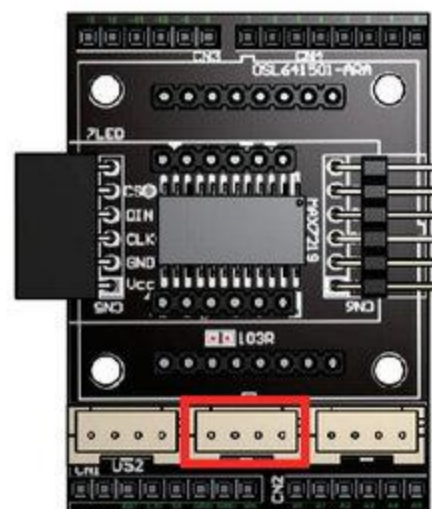


図1-2 マトリクスLEDシールド

<組み立て手順③>

600mm センサーケーブルをコネクタの向きに気を付けて、図のように超音波距離センサーに接続します。コネクタの接続向きをまちがえるとこわれることもあるので、注意しましょう。最後にケーブルのもう一方の端をマトリクスLEDシールドの **US1** に接続してください。

コネクタは
ツメがある方が前面

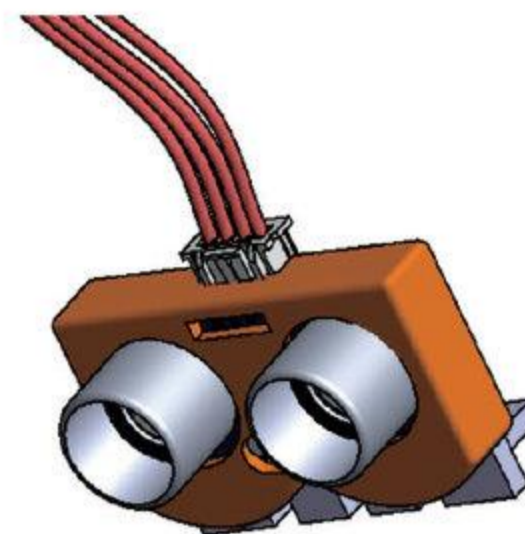
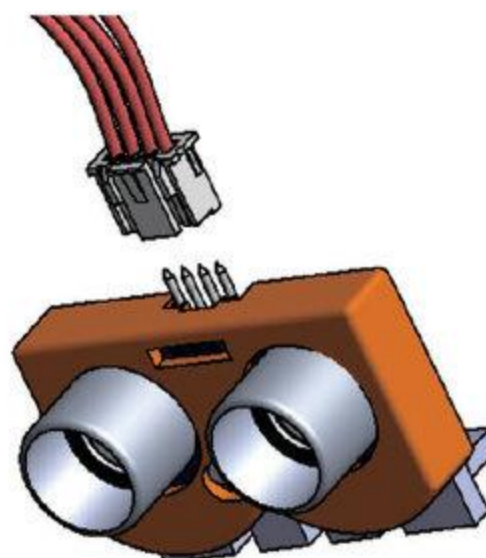
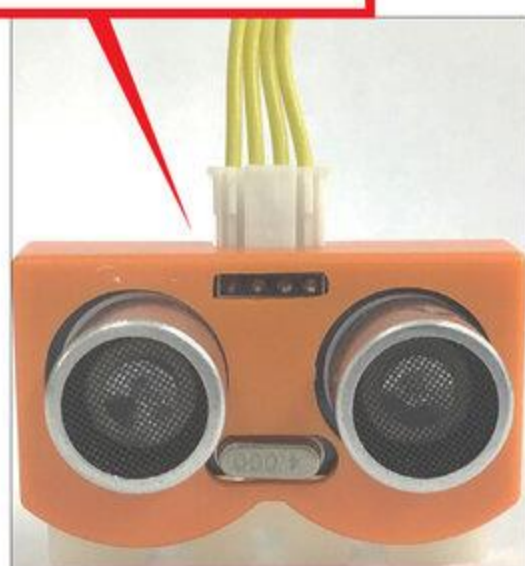


図1-3 600mm センサーケーブルのコネクタの接続

修正済み 要確認

1.2. ちょうおんぱきより超音波距離センサーの動作確認

ここでは、ちょうおんぱきより超音波距離センサーの動作を確認しましょう。ちょうおんぱきより超音波距離センサーの周囲に障害物を置かないように机の上を整理しましょう。

それでは、次のプログラムを実行してください。プログラム実行後、USB ケーブルは抜かないでください。

プログラムの書き込み

RoboticsProfessorCourse2 > ArmRobot5 > UltrasonicDemo

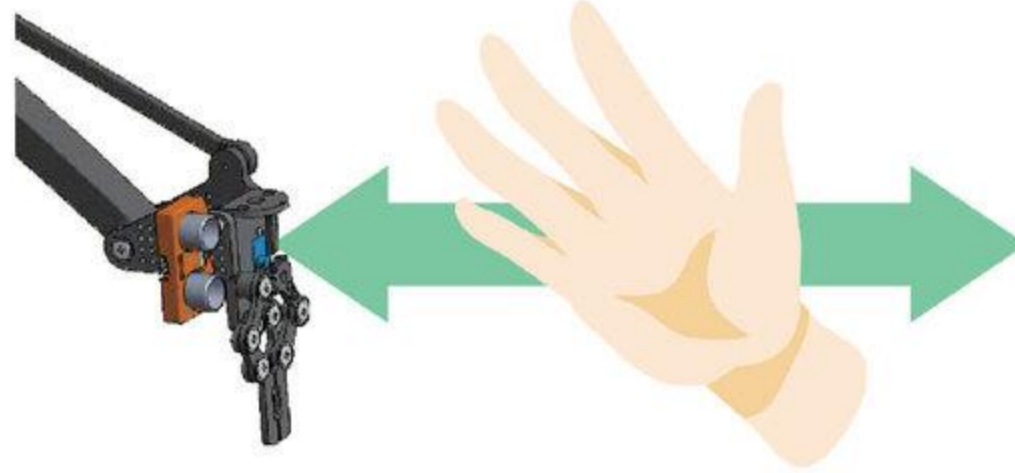


図 1-4 ちょうおんぱきより超音波距離センサーの動作確認

動作確認は図 1-4 のように超音波距離センサーの前に手をかざし近づけたり、離したりして行ってください。

検出結果は、シリアルモニターで見ることができます。設定は「9600baud」にしてください。

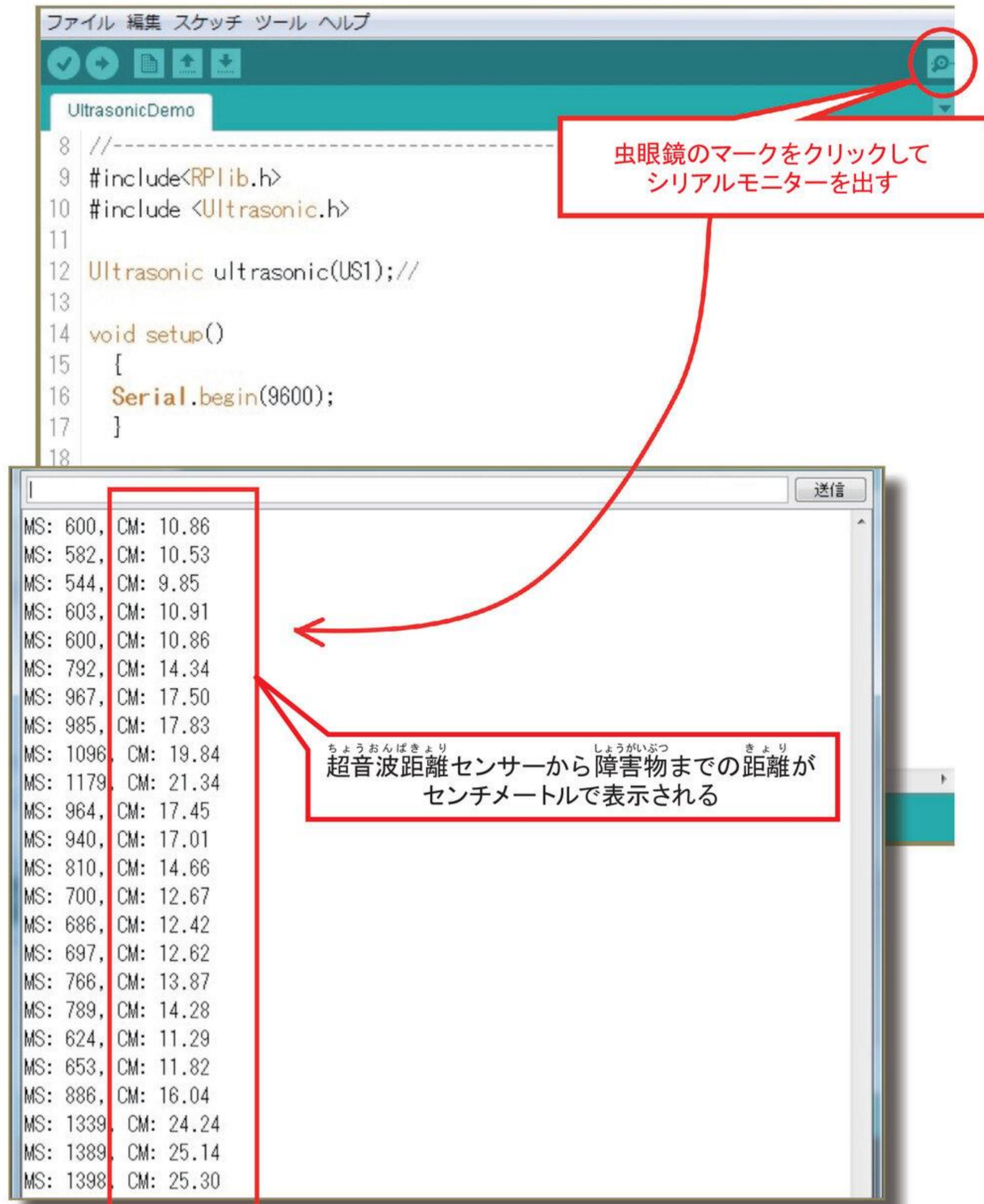


図 1-5 シリアルモニター見本

距離情報は正しく表示されましたか？ 超音波距離センサーの有効範囲に物があると、誤った値を示す可能性があります。また、実際の距離と超音波距離センサーが読み取った値にどの程度の誤差が出るかも確認しておきましょう。

2. ^{ちょうおんばきより}超音波距離センサーを使ったアームロボット (目安 70 分)

2.0. 手から逃げるロボット

まずは、以下のプログラムを開き、アームロボットの調整値を入力しましょう。

🔗 プログラムの書き込み

RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot1

```
armBot.setup( 0.0,0.0,0.0,0.0 );
```

() 内は、(S0 サボモーター [R], S1 サボモーター [L], S2 サボモーター [ROT], S3 サボモーター [GRIP]) の並びです。

入力したら、プログラムを実行して、シリアルモニターを開きましょう。動作スタートの合図はタッチセンサーです。

実行結果：アームロボットの正面左から手を近づけると、^{ちょうおんばきより}超音波距離センサーが手を検知して、右の方に逃げていきます。

シリアルモニターの値も観察しましょう。

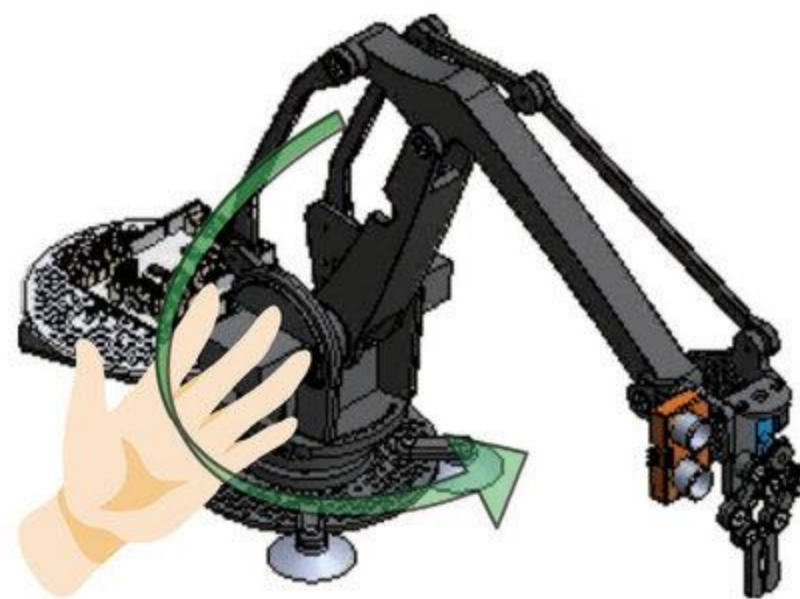


図 2-0 右に逃げるアーム

2.1. プログラムの改造

では、プログラム「SensorArmRobot1」を改造しながら、プログラムに対する理解を深めましょう。その前に、まずはプログラム「SensorArmRobot1」のポイントを学びましょう。

□ プログラム「SensorArmRobot1」より**抜粋**

```

void loop(){
  float cm;
  long microsec = ultrasonic.timing(); // センサー情報を計測
  cm = ultrasonic.convert(microsec, Ultrasonic::CM);
                                // センサー情報を変換

  if(cm < 20){ // もしUS1の20cmの範囲に反応があったら
    armBot.setPosition(100,0,45); // 右に旋回
    delay(1000);
  }
  else{ // そうでなかったら
    armBot.setPosition(100,0,0); // 旋回しない
  }
}

```

このプログラムでは、超音波距離センサーがはかった手までの距離を `cm` という変数にしています。これまでなら `int cm` と宣言していたところですが、`float` という種類の変数を用いることで `cm` が小数の値をとれるようにしてあります。これで、int 型の変数より精密な測定ができるようになります。

そのあとの if 文を見ると、どんなときに旋回するよう命令されているか読み取れますね。それでは、さっそくプログラムを書きかえてみましょう。

やってみよう！

1. 手をかざすと、左方向に逃げるロボットにしてみよう。
2. 手との距離が 10cm 未満になったら反応するようにプログラムを変更してみよう。
3. 次の3つの条件で動くようにしてみよう。
 - ①手との距離が 10cm 未満の場合は左に逃げる
 - ②手との距離が 10cm 以上 20cm 以下の場合は右に逃げる
 - ③それ以外は中心から動かない

💡 ヒント

`armBot.setPosition(前後方向, 高さ, 左右回転);` の命令を思い出そう！

講

解答プログラムはそれぞれ以下となります。

1. RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot1Ans1
 2. RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot1Ans2
 3. RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot1Ans3
- 3のみ解答例の一部を巻末に掲載します。

今度は、手をかざすと、手の動きに合わせてアームが前後に動くプログラムにしてみましょう。

まずは、次のプログラムを実行してください。調整値を忘れずに入力しましょう。

今回も、動作開始はタッチセンサーで命令します。

プログラムの書き込み

RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot2

プログラムの書き込みができたなら、シリアルモニターを開いてください。

下の図のようにアームロボットの正面から手を近づけたり、遠ざけたりしてみましょう。

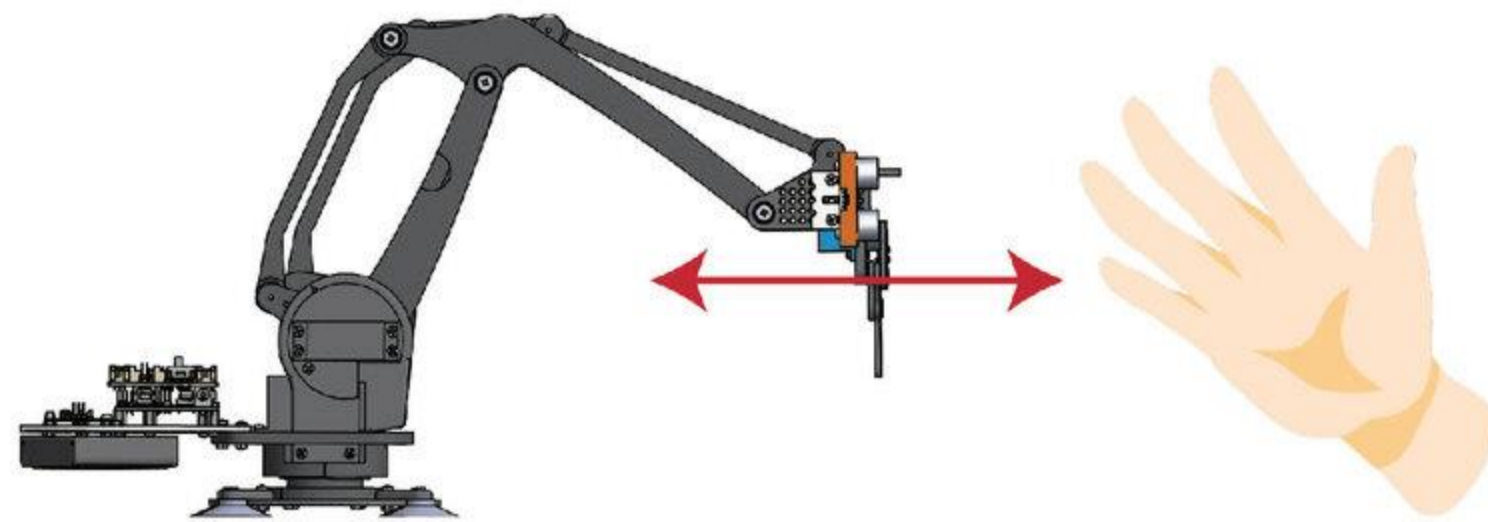


図 2-1 前後するアーム

ステップアップ

プログラム「SensorArmRobot2」のままではまだ未完成だと、気づいたかな？手を近づけるとアームを折りたたむけど、手を遠ざけてもアームを折りたたんだままで伸ばさないよね。では、プログラム内に「手が遠ざかったらアームを伸ばす」という命令を追加してみよう！

💡 ヒント

アームを折りたたむという部分をプログラム内から見つけ出して、参考にしてみよう！

講

解答プログラムは以下となります。
RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot2Ans
解答例の一部を巻末に掲載します。

2.2. ちょうおんぱきより超音波距離センサーの組みかえ

ここまでは、ハンド部に取り付けた^{ちょうおんぱきより}超音波距離センサーからの^{きより}距離をはかっていました。今度は、アームロボットの台座（ベース部）にセンサーの位置を変えてみましょう。ACアダプターとUSBケーブルを一度外してください。取り付けは、以下の図を参考にしましょう。

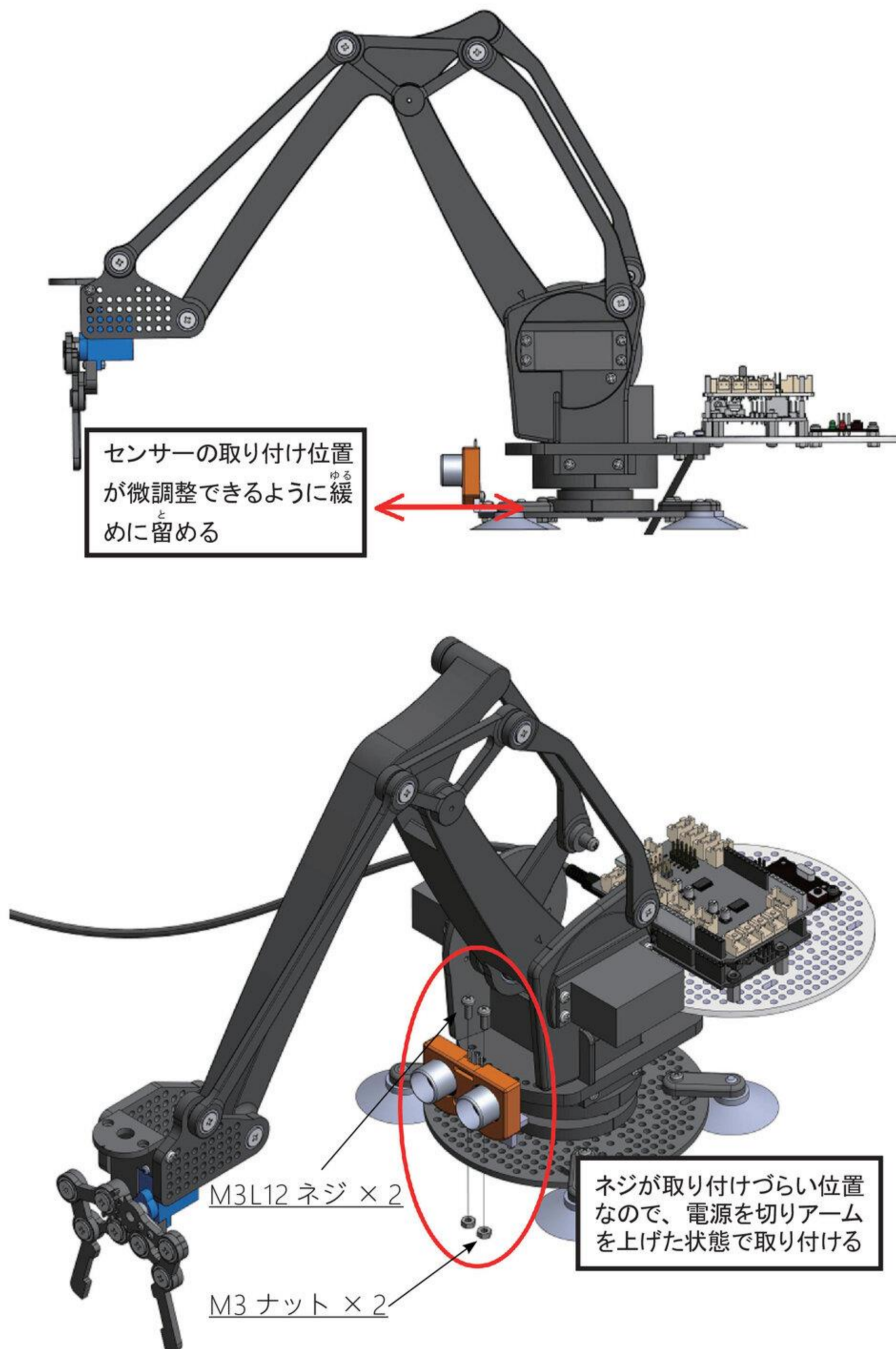


図 2-2 ちょうおんぱきより超音波距離センサーの取り付け

2.3. 自動搬送^{はんそう}の機能の確認

台座（ベース部）に取り付けた超音波距離^{ちようおんぱきより}センサーが、対象物を検知し、つかんで搬送^{はんそう}する動作にチャレンジします。

4つの手順（プログラム）に分けて動作をつくります。それぞれのプログラムを確認しながら完成させていきましょう。どのプログラムも動作スタートの合図はタッチセンサーです。

1) 対象物までの距離^{きより}の測定

まず、アームロボットが対象物までの距離^{きより}を測定できるようにしましょう。次のプログラムを実行し、シリアルモニターを開きましょう。設定は9600baudにします。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot3_1

実行結果：超音波距離^{ちようおんぱきより}センサーの前に手をかざしたり、物を置くと、その距離^{きより}が計測されシリアルモニターに表示されます。

□ プログラム「SensorArmRobot3_1」より抜粋^{ぼっすい}

```
void loop(){
  (中略)
  // 距離をシリアルで表示
  Serial.print("distance: ");
  Serial.println(mm);
  delay(100);
}
```

命令 [print]

動作内容：シリアルモニターに指定の文字・数値を表示する

使い方：Serial.print("Robopro");

// シリアルモニターに「Robopro」という文字列を表示する

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

同じ行の後ろに文字を追加していく [print] と、次の文字を下の行に移す [println] を使い分けています。

やってみよう！

プログラム「SensorArmRobot3_1」を書きかえ、シリアルモニターの表示を以下のように^{へんこう}変更してみよう。

```
distance: 300
```



```
distance: 30 cm
```

講

「計測値を mm → cm へ変換」「末尾に単位を記載」という2つの変更が必要です。単位変換は値を10でわることで行えるので、解答例は以下のようになります。

```
Serial.print("distance: ");  
Serial.print(mm / 10);  
Serial.println(" cm");
```


2) 対象物へ向かう

次は、計測した物体までの位置を目標にして、そこまで手を伸ばすようにプログラムします。
`armBot.setPosition();` を使って、距離^{きょり}に応じてアームが対象物へ向かう動作をつくります。

🔄 プログラムの書き込み

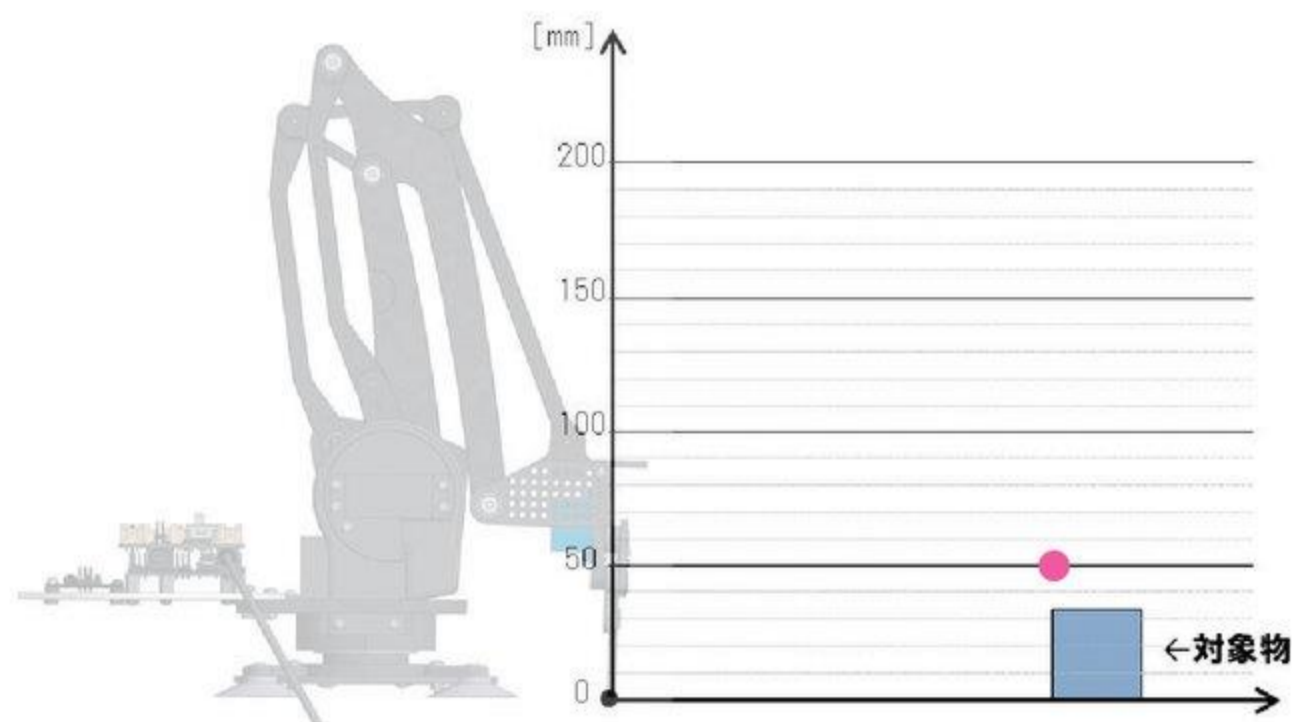
RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot3_2

📄 プログラム「SensorArmRobot3_2」より抜粋^{ばっすい}

```
void loop(){
  float mm;
  long microsec = ultrasonic.timing();           // センサー情報を計測
  mm = ultrasonic.convert(microsec, Ultrasonic::MM); // センサー情報を変換
  (中略)
  armBot.setPosition(mm, 50, 0);                // 対象との距離によって動く
  delay(100);
}
```

やってみよう!

プログラム「SensorArmRobot3_2」の `loop()` 内にある
`armBot.setPosition(mm, 50, 0);` は、ハンド部の先をどこに移動させる処理かな？
 下の図に●で示してみよう。



💡 ヒント

プログラム内にある `mm` という変数には、超音波距離センサーが検知した対象物までの距離^{きょり}が入っているよね。超音波距離センサーが発した超音波は、対象物のどこに当たってはね返っているのかな？

講

超音波距離センサーが計測できるのは超音波が反射した面、すなわち対象物の一番手前の面までの距離であるため、移動距離を `mm` とすると対象物の手前でハンドが止まってしまいます。その為、次のプログラムでは対象物を掴む際にハンドの前後位置を `mm+5` とすることで、もう少し奥にハンドが来るよう調整しています。対象物の大きさによって数値を微調整してください。

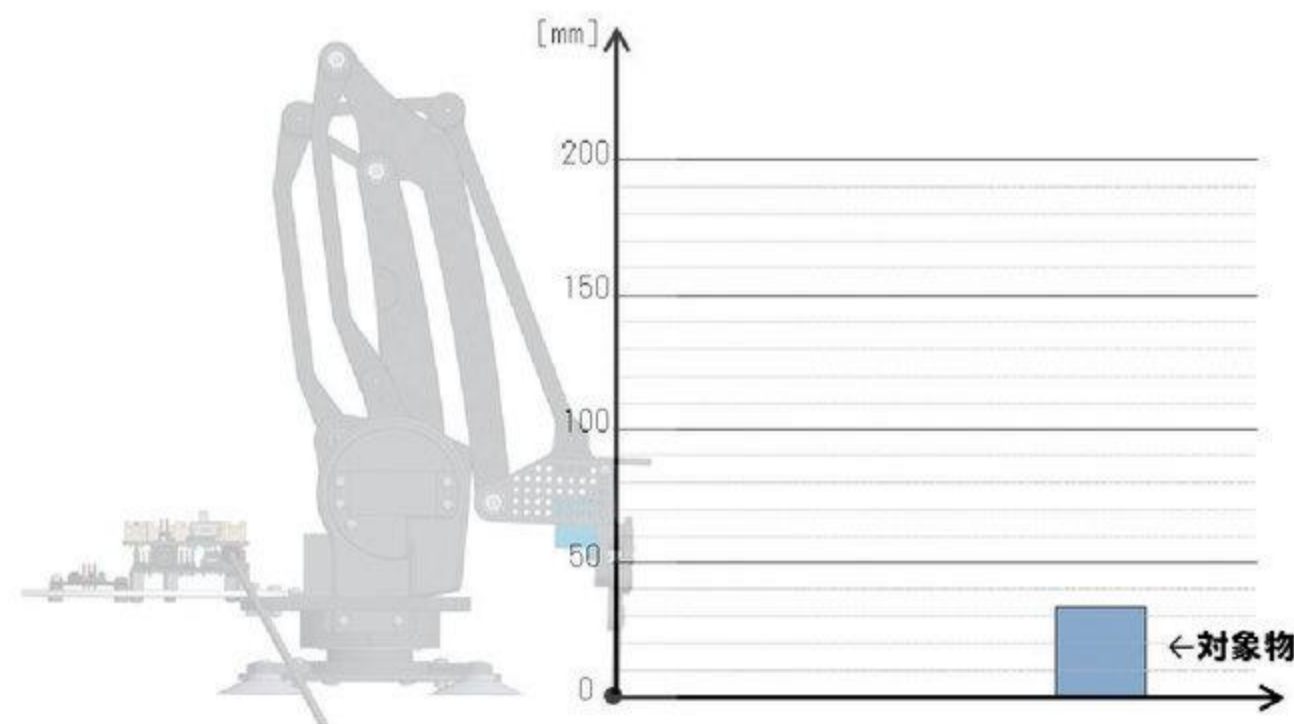
3) 対象物をつかみ、^{はんそつ}搬送する

対象物の近くまでハンドを移動させることができましたね！あとは対象物を^{はんそつ}搬送する動きを追加しましょう。

ステップアップ

プログラム「SensorArmRobot3_2」を書きかえ、対象物をつかんで持ち上げ、ベース部を45度回転させた場所におろすという動きにしてみよう。

ハンド部をどこに移動させればいいのかわからなくなったら、下の図にメモをして考えてみよう！



💡 ヒント

アームロボットのハンド^{そうさ}操作に関する文は `armBot.setPosition();` 以外に以下のようなものがあったね。

`armBot.gripperCatch();` ハンド部を閉じる。

`armBot.gripperRelease();` ハンド部を開く。

また、対象物の位置にうまくハンド部が来ないときは、プログラム「OriginAdjust」で調整値をはかりなおしてみよう！

講

解答例は以下のプログラムです。

対象物をつかむ動作まで：

RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot3_3

対象物をつかみ、45度はなれた場所に搬送しておろす：

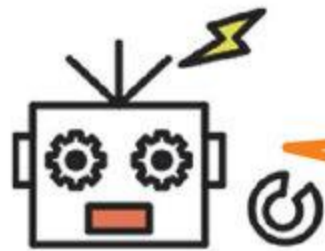
RoboticsProfessorCourse2 > ArmRobot5 > SensorArmRobot3_4

3. まとめ（目安5分）

さて、カシコイセンサーロボットができる道のりはわかりましたか？

プログラムの構造自体は今まで使ったものとほとんど同じですね。if文を使いこなすことができる^{じょうけんぶんき}とさまざまな条件分岐がつくれて、思いどおりの人工知能を持ったロボットになります！

今回は、次回の自動搬送^{はんそう}アームロボットのウォーミングアップです。次回は自動搬送^{はんそう}を応用してロボット競技をしましょう！



センサーやif文のテクニックでカシコイロボットがデキタ！
使い方次第でまだまだカシコクなれるよ。

《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。






ラジオペンチ	1	ドライバー	1	USB ケーブル	1	コントローラー	1
							
AC アダプター	1						
							

図 3-0 次回必要なもの

講

- 以下の理解度を確認します。
 - ・アームロボットを超音波距離センサーで動かす
 - ・if文を使いこなす
 - ・自動搬送に必要な機能を理解する
- 次回のテーマは「自動搬送ロボット」です。
引き続き、今回使用したアームロボットや超音波距離センサーを使います。
次回は、第5回までのおさらいとロボットを使ったゲームをします。

P.8 やってみよう！ 解答例 (SensorArmRobot1Ans3)

```
if(cm < 10){ // もしUS1の10cmの範囲に反応があったら
    armBot.setPosition(100, 0, -45);
                // 左に旋回
    delay(1000);
}
else if(cm >= 10 && cm <= 20){
    // もしUS1の10cmから20cmの範囲に反応があったら
    armBot.setPosition(100, 0, 45);
                // 右に旋回
    delay(1000);
}
else{ // そうでなかったら
    armBot.setPosition(100, 0, 0);
                // 旋回しない
}

delay(100);
}
```

P.9 ステップアップ 解答例 (SensorArmRobot2Ans)

```
if (cm <= 10){
    d--;
}
else{
    d++;
}
d = constrain(d, -100, 100); // 動く範囲を制限
armBot.setPosition(150 + d, 50, 0); // 対象との距離によって動く
delay(10);
}
```