

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①

第2回

角速度を学ぶ

講師用

目 次

0. 角速度を学ぶ

0.0. 「角速度を学ぶ」でやること

0.1. 必要なもの

0.2. 動作確認

1. 角速度

1.0. 角速度とは

2. 角速度の応用

2.0. 角速度を使って角度をはかる

2.1. 角速度を音で体感する

2.2. 角速度を図として表示させる

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。

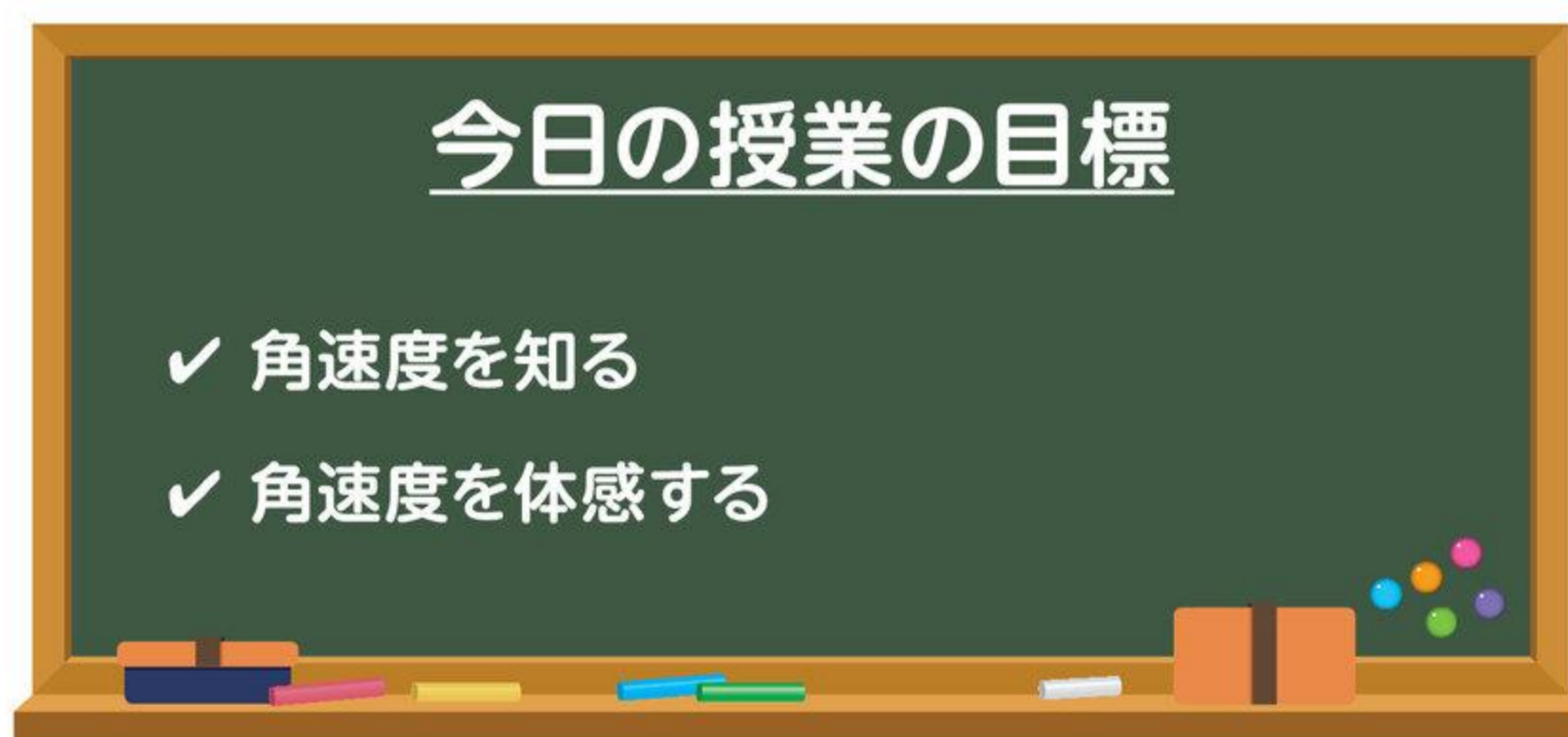
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(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. 角速度を学ぶ (目安 15分)

0.0. 「角速度を学ぶ」でやること

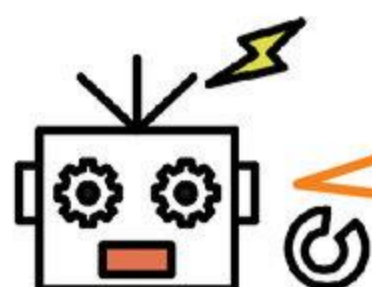


今回も、「^{とうりつしんし}倒立振り子」を作るための知識を確認しましょう。前は、^{しせいけんしゆつ}姿勢検出シールドの機能の1つである「加速度センサー」について学びましたが、今回は「角速度センサー（ジャイロセンサー）」です。

そもそも「角速度」とは何でしょうか？ 物体が1秒間でどのくらいの角度を回転したかといった、単位時間あたりの回転角のことです。例えばタイヤが回転するのにも、角速度があります。この値を知っておくと、タイヤがどのように動いていて、どのような^{じょうきょ}状況にあるのか、などを把握することができるのです。

その「角速度」を^{けんしゆつ}検出するのが今回の「角速度センサー」です。人間の耳の中にある^{へいこうかんかく}平衡感覚をつかさどる^{きかん}器官（^{さんはんきかん}三半規管）と同じような役割をされていて、「ジャイロセンサー」とも呼ばれます。手ぶれ防止機能でレンズごと動くようなビデオカメラなどに入っていて、「セグウェイ」にも使われています。

今回はそんな「角速度センサー」を体験して使いこなせるようになりましょう！



前は「加速度」、今回は「角速度」だよ。似ているけど、別物だ！

0.1. 必要なもの

姿勢検出ユニットは、前回使用したのを使います。

並び順は、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です。

それぞれのシールドのピンはピンソケットにきちんとはめ込んでください。図0-1で注意事項も確認しましょう。

第1回で使っていた「スピーカー」が取り付けっぱなしになっている場合、はじめのうちは取り外しておきましょう。

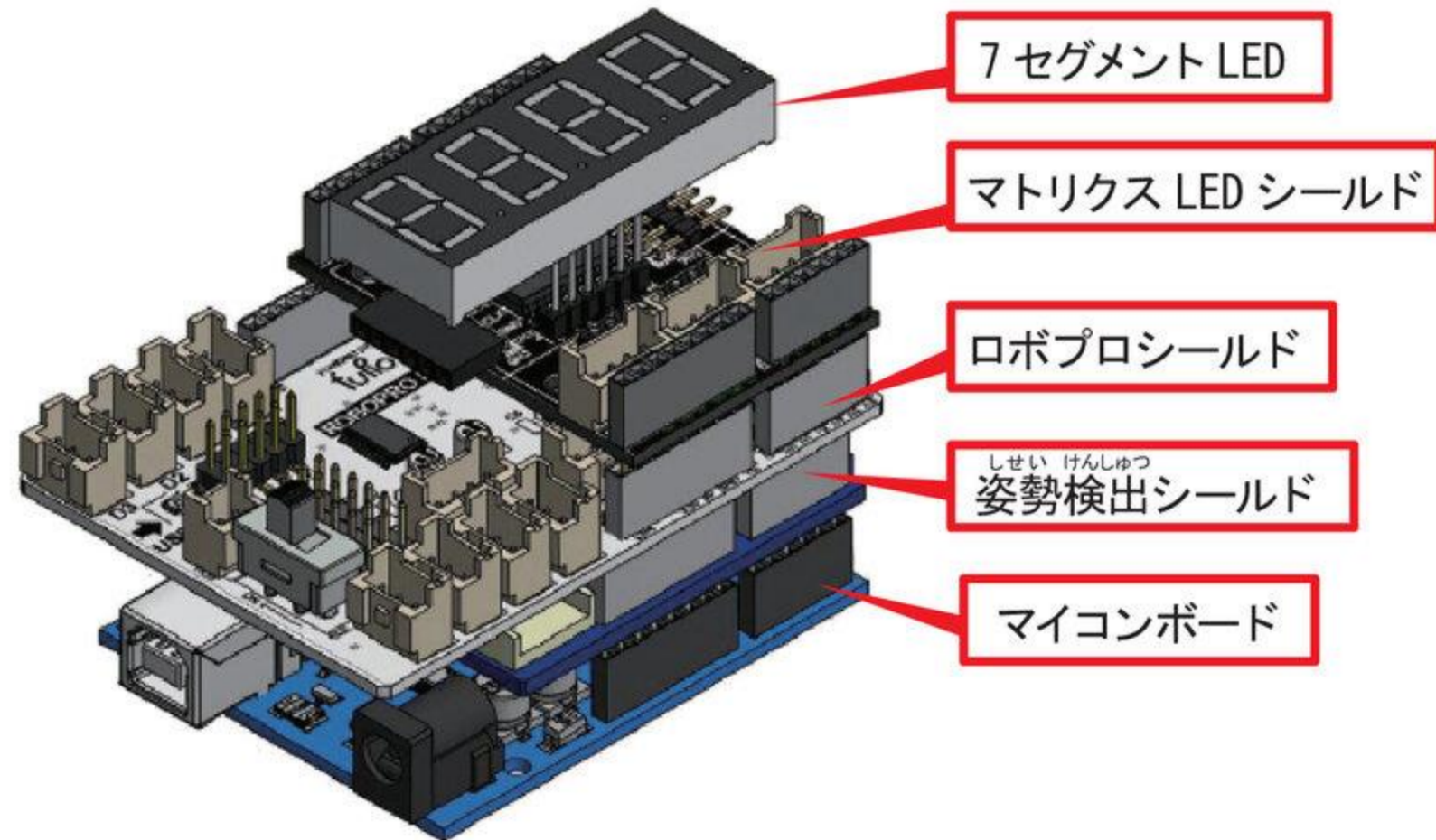


図0-0 姿勢検出シールドの接続

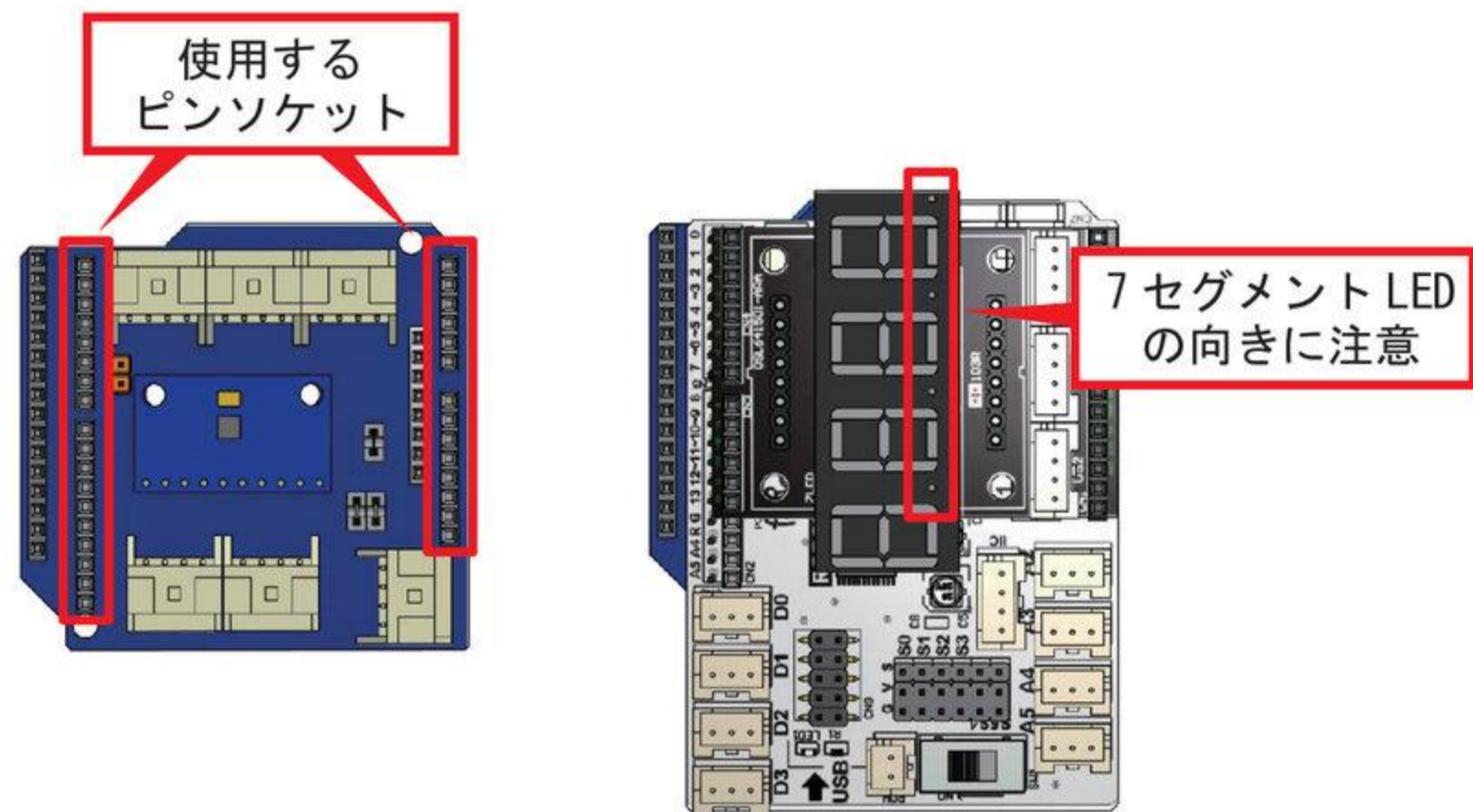


図0-1 姿勢検出シールドの組み立て時の注意

講

ピンヘッドの折れ曲がりにはラジオペンチで丁寧に直して接続させてください。折れ曲がった状態で接続すると、取り外しのときに、ピンソケットが基板から剥離する恐れがあります。

0.2. 動作確認

はじめに、前回と同じように動作確認をします。^{しせいけんしゅつ}姿勢検出ユニットをUSBケーブルでパソコンと接続し、電池ボックスも接続してください。

接続できたら^{しせいけんしゅつ}姿勢検出ユニットをテーブルにいったん置いて、以下のプログラムをかき込みましょう。

∞プログラムの書き込み

RoboticsProfessorCourse2 > Inverted1 > MPUtest

プログラムがかき込まれたら、シリアルモニターを立ち上げて、通信速度を「115200baud」とします。



図0-2 シリアルモニターの表示のやり方

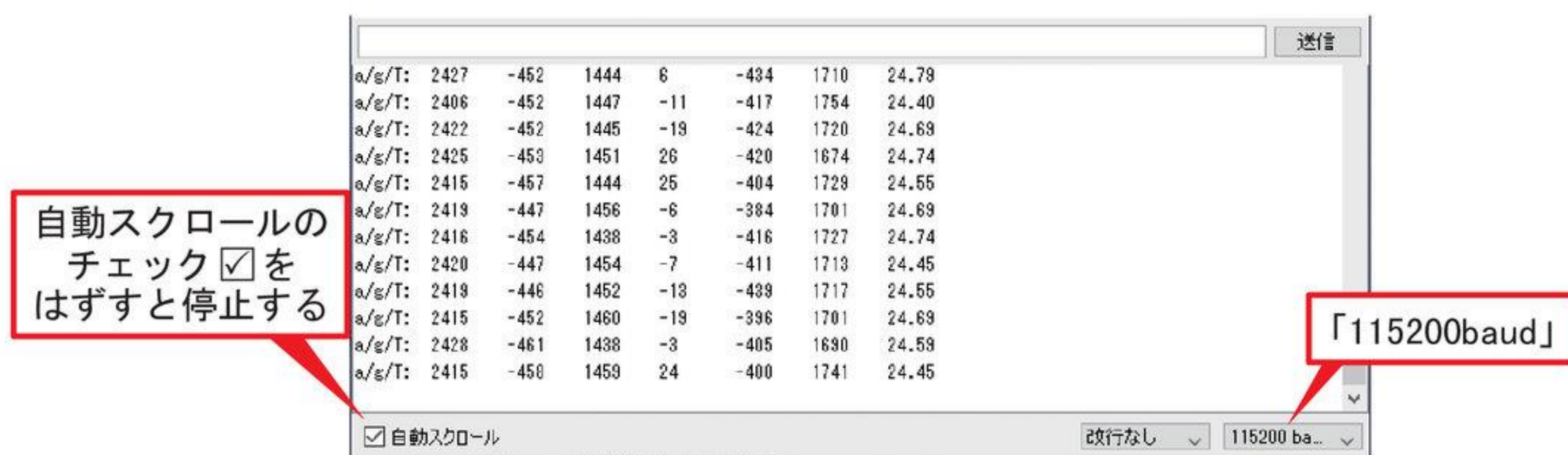


図0-3 シリアルモニターの操作

図0-3のように7列の数字が表示されて、絶えず更新されている状態が正常です。^{しせいけんしゅつ}姿勢検出ユニットを動かすと、数字が変化します。変化しない数字があれば、シールド同士がしっかりと接続されているか、ピンが曲がってしまってソケットに入っていないものがないか、確認してください。ただし、一番右側の数字は温度センサーのため、あまり変化がありません。

講

電池ボックスは、この場面では使用しないので邪魔になるようであれば接続しなくても大丈夫です。ただし、この後姿勢検出ユニットを左右や上下に振るなどする場面では使用した方が便利ですので、必要になったら再び接続し、使用してください。

今回は「角速度」の観察ですので、4列目、5列目、6列目の値に注目しましょう。
 姿勢検出ユニットは、落下させたり、ぶつかけたりしないように慎重に取り扱ひましょう。

							送信
a/g/T:	2427	-452	1444	6	-434	1710	24.79
a/g/T:	2406	-452	1447	-11	-417	1754	24.40
a/g/T:	2422	-452	1445	-19	-424	1720	24.69
a/g/T:	2425	-453	1451	26	-420	1674	24.74
a/g/T:	2415	-457	1444	25	-404	1729	24.55
a/g/T:	2419	-447	1456	-6	-384	1701	24.69
	1列	2列	3列	4列	5列	6列	7列

図0-4 シリアルモニターの列

表0-0 シリアルモニターの列表示の設定

1列	x軸方向の加速度 (軸方向への加速度)
2列	y軸方向の加速度 (軸方向への加速度)
3列	z軸方向の加速度 (軸方向への加速度)
4列	x軸の回転角速度 (軸まわりの単位時間あたりの回転角)
5列	y軸の回転角速度 (軸まわりの単位時間あたりの回転角)
6列	z軸の回転角速度 (軸まわりの単位時間あたりの回転角)
7列	温度 (°C) センサーの温度

姿勢検出ユニットを各軸を中心に、緑の矢印方向に回転させてみましょう。
 また、回転させる速度を変えて値を観察してみましょう。

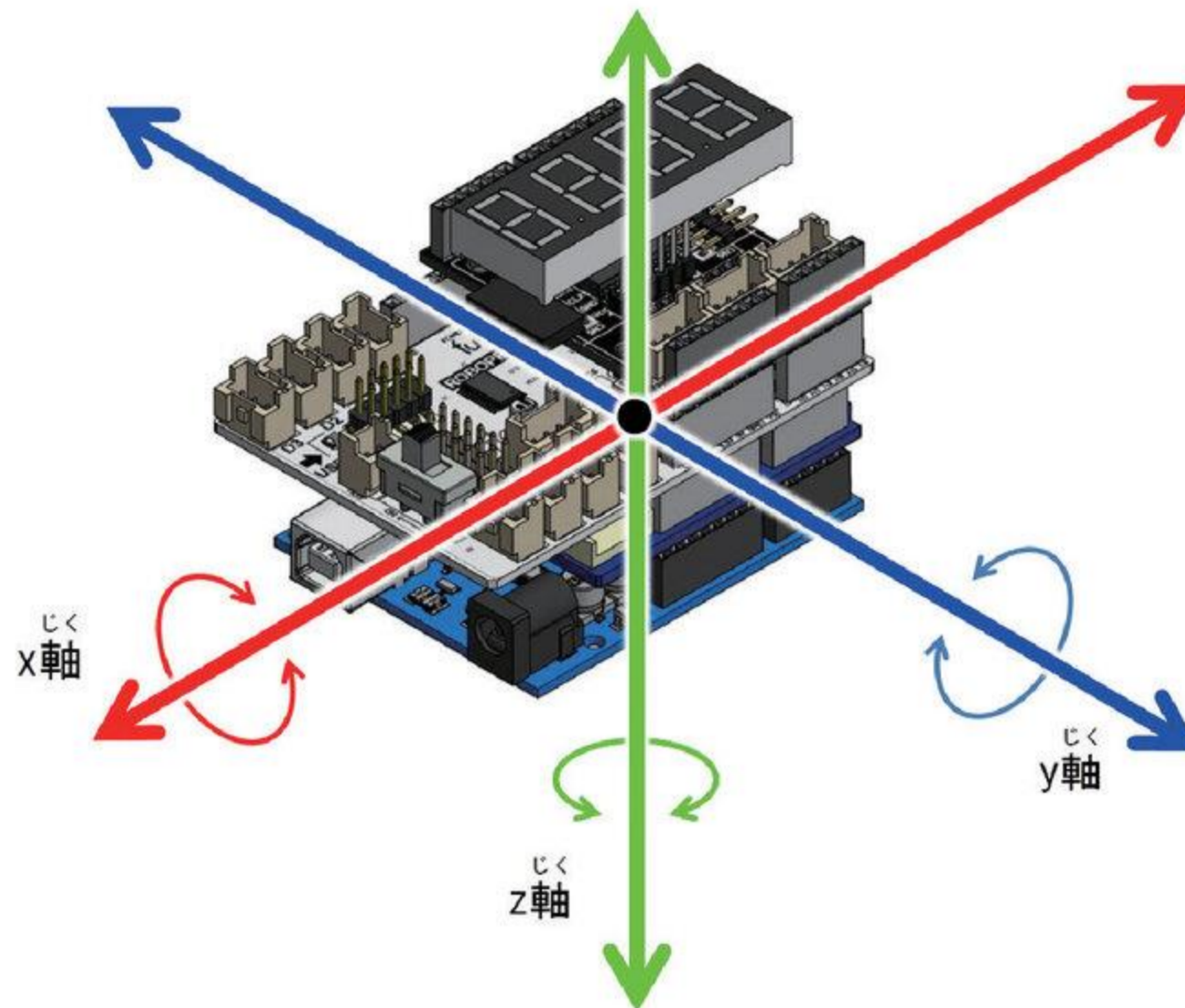


図0-5 軸方向

1. 角速度 (目安 15分)

1.0. 角速度とは

皆さんは「ペニー・ファージング」とよばれる自転車の一種を知っていますか？

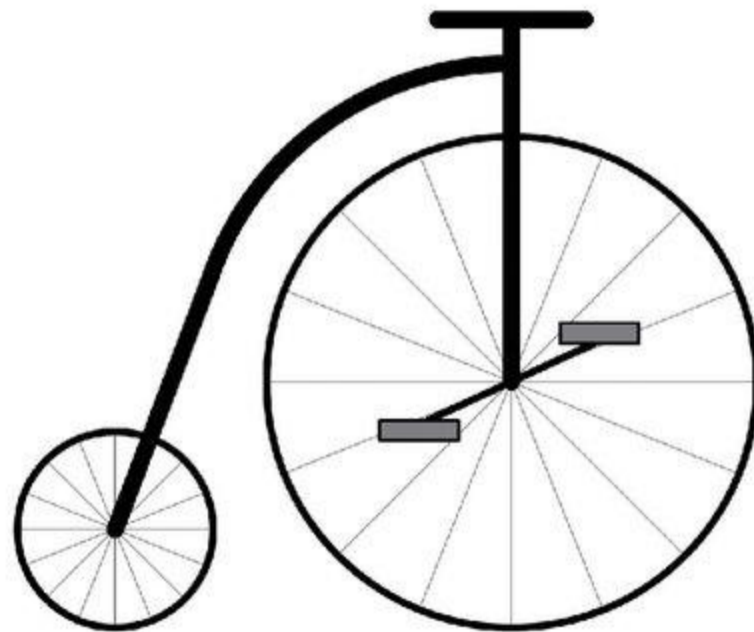


図 1-0 ペニー・ファージング型自転車の模式図

このように、前輪が後輪と比べて極端に大きくなっている自転車です。1800年代後半ごろは「自転車」といえばこの形、と言われるほど一般的なものでした。

乗り手が前輪のてっぺんに座るため限度があるものの、「前輪は大きければ大きいほど良い」という考えが強かったそうです。現代の自転車に慣れ親しんでいる私たちには「なぜ？」としか思えませんが、これは当時の自転車に「ギア」や「チェーン」といった機構がなく、ペダルが直接前輪に取り付けてあったために生まれた考え方のようです。

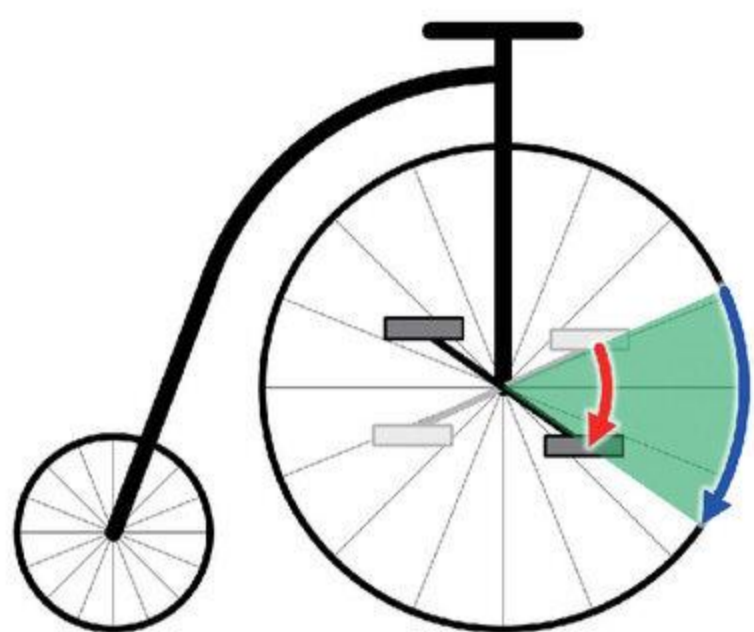


図 1-1 ペニー・ファージングのペダルをこぐ

ギアが無いので、ペニー・ファージングの前輪はペダルの動きと完全に連動しています。ペダルを1周分こげば前輪も1周するというわけです。

ということは、ペダルを動かした距離（赤い矢印）より前輪が回転する距離（青い矢印）が大きいほど1こぎで進める距離が長くなる、つまり速く走れるということになりますね。

そのためには、前輪をなるべく大きくする必要があった、というわけです。大きいものだと前輪の直径が1.5mをこえるものもあったそうです。サドルに座ったら地面がかなり下に見える事でしょう。

さて、たった今ペダルと前輪の差を考えましたが、よく考えてみると、ペダルと前輪の「回転速度」は同じでしょうか、それともちがうでしょうか。

それぞれの「移動距離（弧の長さ）」で考えれば、明らかに前輪の方が長い距離を移動しています。しかし、「角度」で考えるとどうでしょうか。ペダルを 90° 回転させれば前輪も 90° 回転するわけですから、「ペダルを回した角度と前輪が回転する角度は等しい」ということになります。

このように、回転運動をしている物体の速さを「距離」ではなく「角度」で考えるのが「角速度」です。ペダルと前輪は、「速度」は等しくないが「角速度」は等しいと言えます。

角度の単位は「度 (degree)」、時間の単位は「秒 (second)」ですから、「1秒あたり何度回転するか」で角速度を求めれば単位は「deg/s」となります。つまり、角速度は「角度÷時間」で求められるというわけですね。



豆知識

高校数学では、約 57.29 度の角度を「1ラジアン」とする「弧度法」とよばれる角度の表しかたも学びます。この場合、角度の単位が度ではなく「ラジアン (radian)」になるので、角速度の単位は「rad/s」となります。こちらの単位を使った方が計算しやすくなる場面も多く、国際的にみても使われる機会の多い単位なので、弧度法を学んだ際に意識しておくといいかもかもしれませんね。

では、角速度を体感してみましょう。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 > Inverted2 > GyroLED

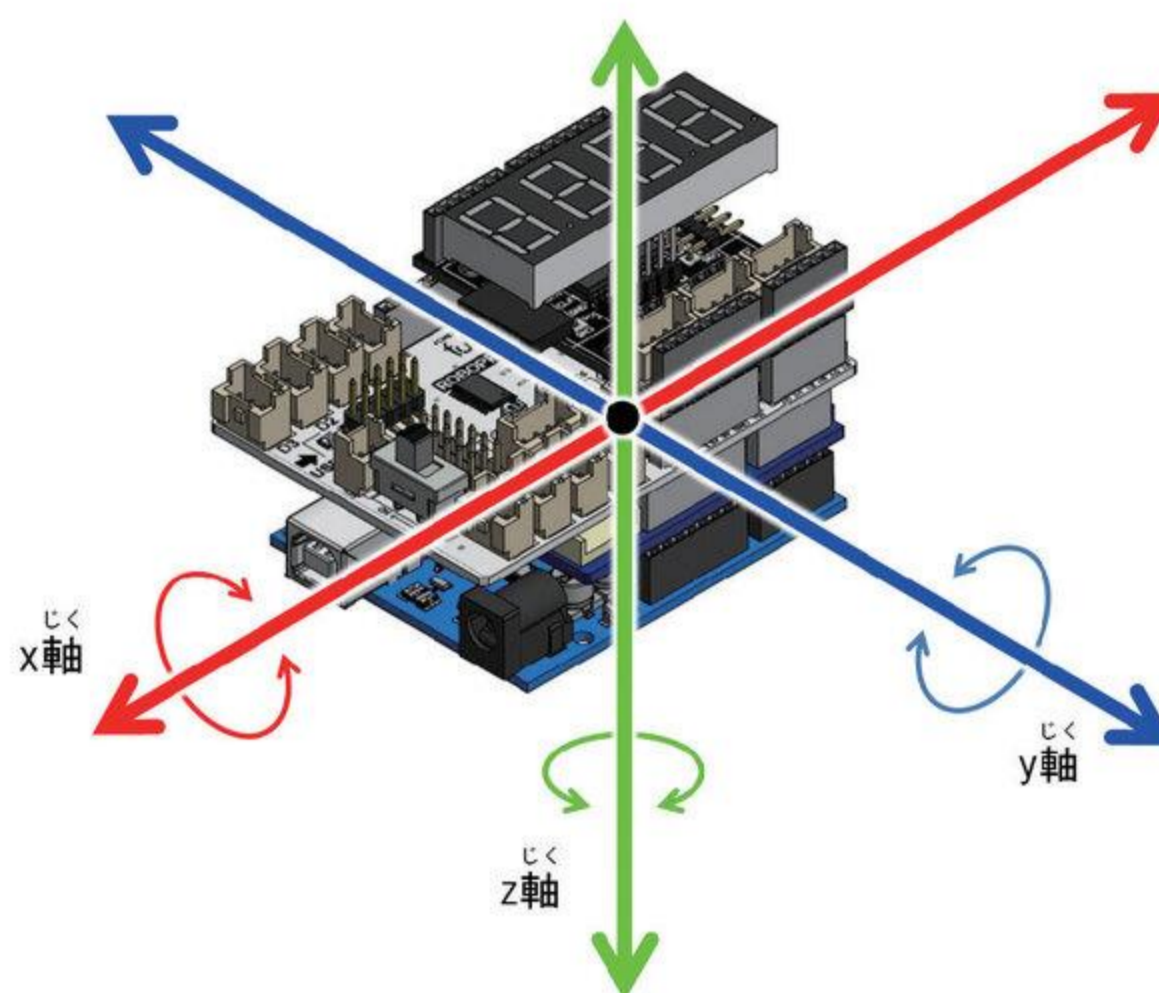


図1-2 軸方向

実行結果：図1-2のx軸まわりの回転角速度が7セグメントLEDに表示される。

やってみよう！

プログラム「GyroLED」を変更して、x軸だけではなく、y軸、z軸まわりの値もLEDに表示させてみよう。以下の黄色の部分の `gx` を、`gy` と `gz` にそれぞれ変更して、yとz軸まわりの回転による値の変化を見てみよう。

```
lc.setDec(0, gx/16.4); // x軸まわりの回転角速度をLEDに表示 (deg/s)
```

講

安全対策として、姿勢検出ユニットを動かす際には、USBケーブルをパソコンから外し、電池ボックスからの電力で動かすように指導してください。また、電池ボックスのリード線がちぎれないように手でおさえて実験するようにしてください。



ジャイロセンサーの活用

角速度を検出するセンサーは一般的にジャイロセンサー（ジャイロスコープ）と呼ばれています。このジャイロセンサーはいろいろな乗り物に搭載されて、欠かせないセンサーとなっています。例えば、飛行機や船舶、ロケット、レーシングカーといった乗り物では、位置情報を把握し、姿勢制御を行うために高性能なジャイロセンサーが使われています。また、ロボットやラジコンヘリコプターなどでも姿勢制御用として使用されています。身近なところでは、タブレット型端末、スマートフォン、ゲーム機などに使用されています。また、手ぶれ補正機能付きデジタルカメラや、ジャイロセンサーで距離や速度情報を管理するカーナビゲーションシステムなどその使われ方はさまざまです。


ジャイロセンサーを入力装置として使う場合は、本体を傾けるとその方向や度合いを検出し、手の動きによる感覚的な操作が可能になります。



やってみよう！

フィギュアスケートには、「トリプルアクセル」という技があるよね。空中で3回転半まわる技だね。では、^{たいくう}滞空時間が0.6秒だった場合の、トリプルアクセルの角速度を計算してみよう。単位は「deg/s」で教えてね。



 $3.5 \times 360 \div 0.6 = 2,100 \text{deg/s}$

やってみよう！

^{しせいけんしゅつ}姿勢検出ユニットを持ったまま、フィギュアスケートのようにぐるりとからだを1回転したときの回転速度を計測できるようにしよう。そのためには、ジャイロセンサーが^{けんしゅつ}検出した一番大きな値を記憶させる（表示を最大値で止める）必要があるよね。ということで、命令「max」を使って、プログラム「GyroLED」を改造してみよう。なお、実際に計測する場合はけがをしないように注意しよう！

命 令 「max」

実行結果：数値1と数値2を比べて、大きい方の数値を返す

使い方：max([数値1],[数値2])

命令「max」は以下のように使うよ。以下の場合結果として、「a = 20」になるよ。

```
int a;  
a = max(10, 20);
```

 ヒント

以下のように、プログラム「GyroLED」に黄色の部分をかき足そう。

 プログラム「GyroLED」より**抜粋**

```
int g_max; // 変数の名前をg_maxと命名
void loop() {

    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
    // 姿勢センサーの検出
    g_max = max(g_max, gx);
    lc.setDec(0, g_max / 16.4);
    // x軸まわりの回転角速度をLEDに表示 (deg/s)
    delay(100);
}
```

講

値が常に動いていると、確認するだけで大変な作業になります。その為、max関数を活用します。max関数を使うと、計測した値の最大値を保持することができます。なお、今回使用するセンサーの最大値は、2,000deg/sとなります。実験の際は、生徒同士の手がぶつかったりして怪我をすることのないようご注意ください。

解答プログラムは以下となります。

RoboticsProfessorCourse2 > Inverted2 > GyroLEDmax

2. 角速度の応用 (目安 75 分)

2.0. 角速度を使って角度をはかる

前回は図 2-0 のように、テーブルにセンサーを置いたときに、地面に対して垂直方向にある軸に対する角度を検出できました。あれは、重力加速度という地球の力を使ったことにより鉛直方向（水平面に対して垂直の方向）からの角度をはかったわけです。では、z 軸を中心とした回転の角度、つまり水平方向の角度はどうやったらはかれるのでしょうか？

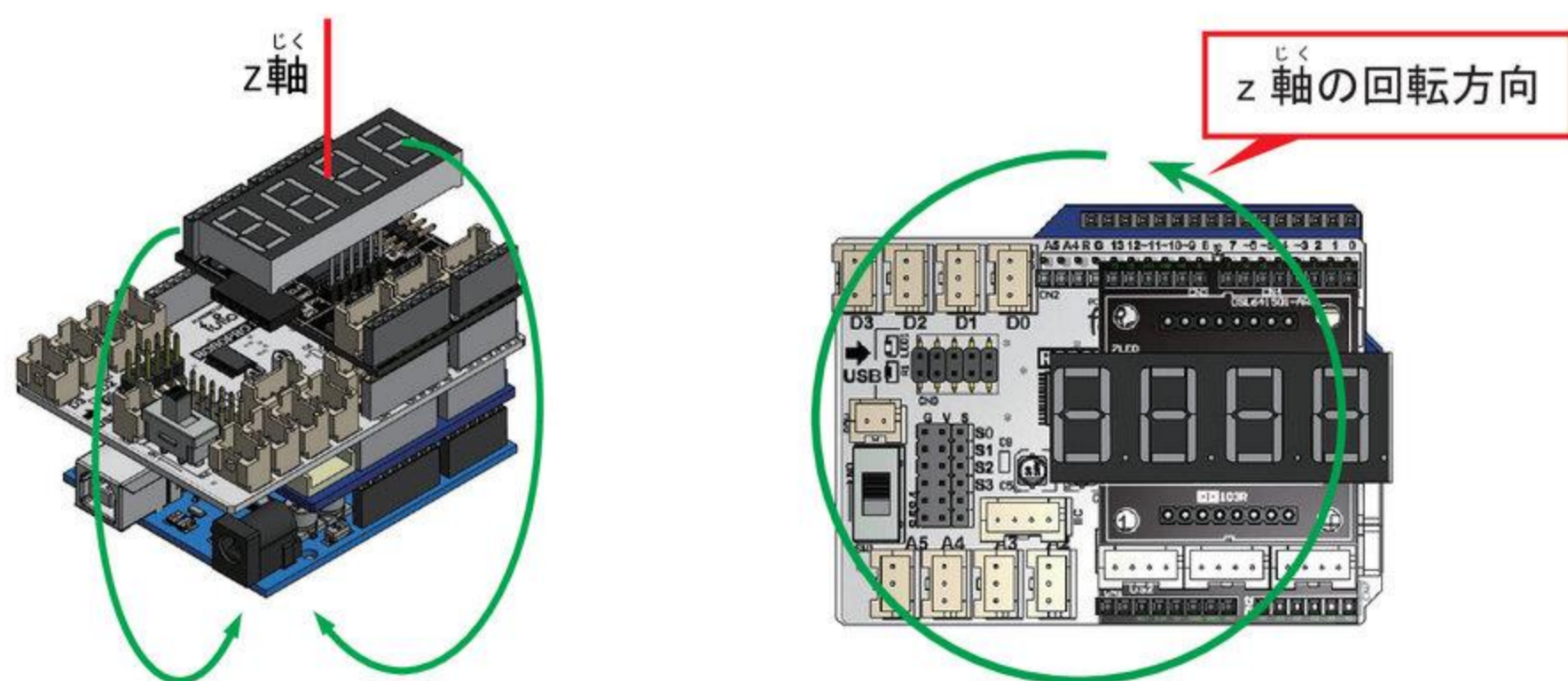


図 2-0 姿勢センサーの回転方向

まずは、以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse2 > Inverted2 > AngularLED

実行結果：はじめは7セグメントLEDのまわりを光がまわる（センサーが初期値を探しています）。動かさずにしばらく待ち、数値が表示されてからz軸を中心に回転させると、7セグメントLEDの数値が変化します。

水平方向の角度が表示されました。どうやっているのでしょうか？ ちょっと、速度と距離の関係にもどって考えましょう。速度と動いた時間がわかれば、移動距離は計算できますよね。



POINT

「距離 = 速度 × 時間」と同じ考えで、「角度 = 角速度 × 時間」という式ができます。

サンプルプログラムでは10ミリ秒の間隔 (`delay()` の時間) ですから、「角度 [deg] = 角速度 [deg/s] × 0,01[s]」の計算をしています。

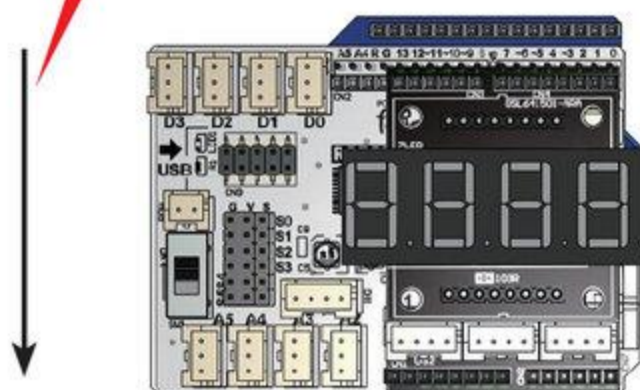
出てきた角度をすべて足せば、回転した角度を求められます。こうすると鉛直、水平方向に関係なく角度をはかることができるのです。ただし、実際はもうちょっとややこしいです。詳しくは積分という、高校の数学で習います。

やってみよう！

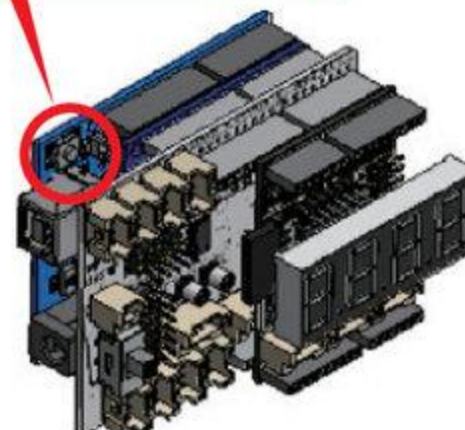
プログラム「AngularLED」で角度検出^{けんしゅつ}の実験をしてみよう。

図のようにテーブルに姿勢^{しせい}センサーを立て、マイコンボードのリセットボタンを押して初期化しよう。

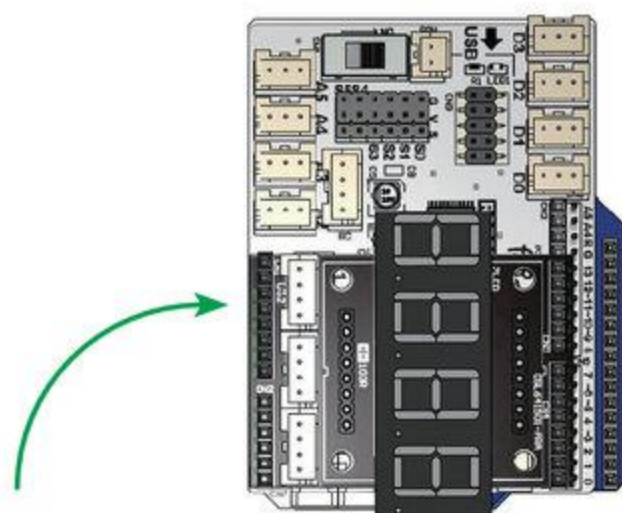
テーブルに立てる



リセットボタン

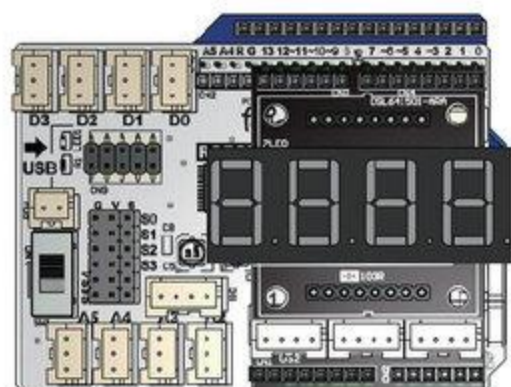


右にコロリと回し、このときの値（角度）を読み取ろう。



約 90 度

再び元の状態にもどし、このときの値（角度）を読み取ろう。



約 0 度

かなり精度よく値が出るよね。くり返していくとどうなるか、観察してみよう。
どんな結果になったかな？

講

概ね 10 回ほどくり返させてください。
結果、くり返すうちに徐々に値に誤差が出ることを確認してください。

10回くらいくり返していると、最初0だったはずの角度がずれてきたのではないのでしょうか？
 実は、同じ場所に放っておいても、時間が経つとずれてきます。根本的に角速度センサーは、ずれてくるものです。このように、測定器で測定する対象や条件を固定しても、時間とともに、示される値がしだいにずれることを、専門用語で「ドリフト」といいます。
 もちろんセンサー自体に問題もあるのですが、場合によっては地球の自転の角速度を検出してずれていく場合もあります。これを「コリオリの力」といいます。



豆知識

コリオリの力

私たちがいる地球は地軸を中心に東向きに自転しています。その上に暮らす私たちは地球の重力によって回転運動をさせられています。つまり外力を受けているのです。したがって、私たちが地球上で行う観測には、かならず地球の回転運動による「みかけ」の力が加わっているように見えることとなります。この「みかけ」の力を転向力、あるいはこの力を発見し体系づけた科学者の名前を取って「コリオリの力」といいます。

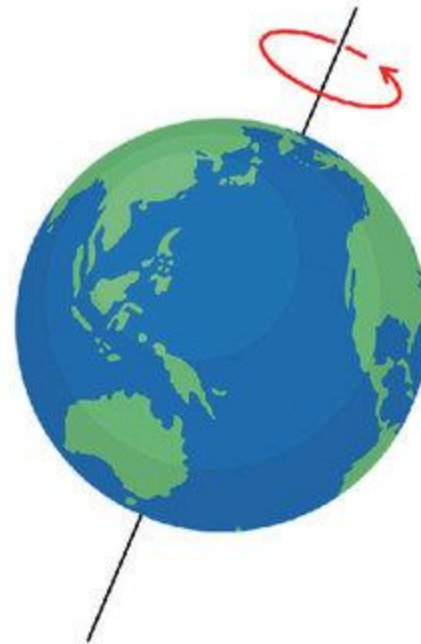


図 2-1 地球の自転

前回の重力加速度を利用した傾斜計にはなかった欠点です。なかなかうまくいかないものですね。では、逆に前回の傾斜計の欠点はわかりますか？
 もう一度、前回のプログラム「inclinometer」を実行して観察して考えてみましょう。ヒントは、速い動きをすると原理的に問題があるわけです。

プログラムの書き込み

RoboticsProfessorCourse2 > Inverted1 > inclinometer

どのような欠点があるのでしょうか？

重力加速度以外の加速度がかかる場合（激しい動きや、振動のある場所での検知など）は、重力加速度の検出に誤差が出てしまうため、対応できない。

では今度は、今回使った角速度を利用した測定器の利点を考えてみましょう。

速い動きや水平方向の動きに対応できる。

2.1. 角速度を音で体感する

加速度のときと同じように角速度をわかりやすく体感してみましょう。
スピーカーを [D1] に接続し、以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > Inverted2 > GyroTone

実行結果：スピーカーから音が出る。x 軸まわりの回転角速度によって音が変化する。
音の変化を、注意して聴いてみましょう。

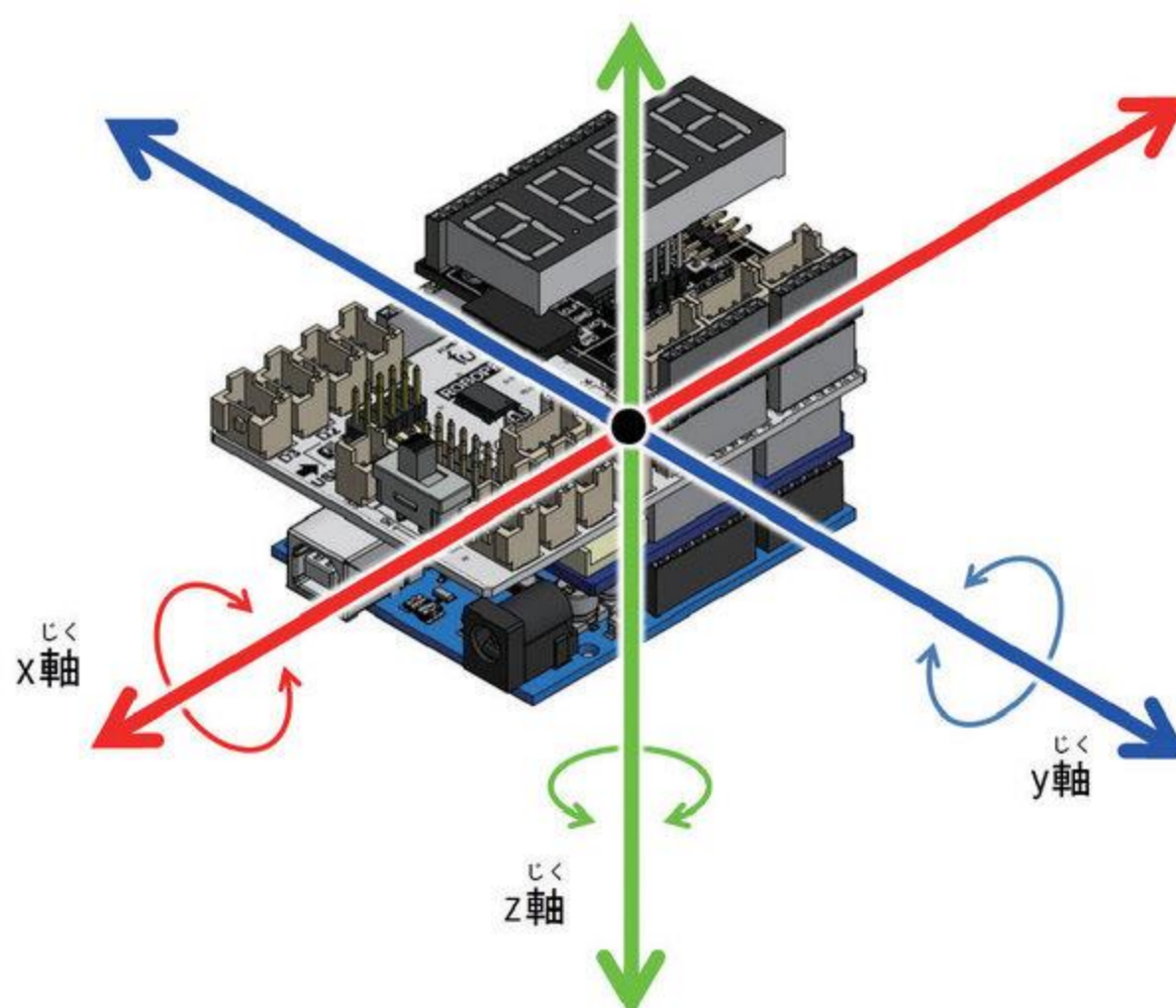


図 2-2 軸の回転方向

チャレンジ課題

プログラム「GyroTone」では、x 軸まわりの回転角速度によって音が変化したね。では、プログラムを改造し、y 軸まわりの回転角速度でテンポが変化するようになしてみよう。

講

解答プログラムは以下となります。

RoboticsProfessorCourse2 > Inverted2 > GyroToneChallenge

解答例プログラムでは変数 [gx] を 16.4 で割っていますが、これは角速度を deg/s へ変換するためです。ただこの処理が無くても「加速度が大きいほどテンポが遅くなっていく」という動作になることは変わりませんので、必須ではありません。

2.2. 角速度を図として表示させる

続いては、図2-3のように、7セグメントLEDをマトリクスLEDに付けかえます。

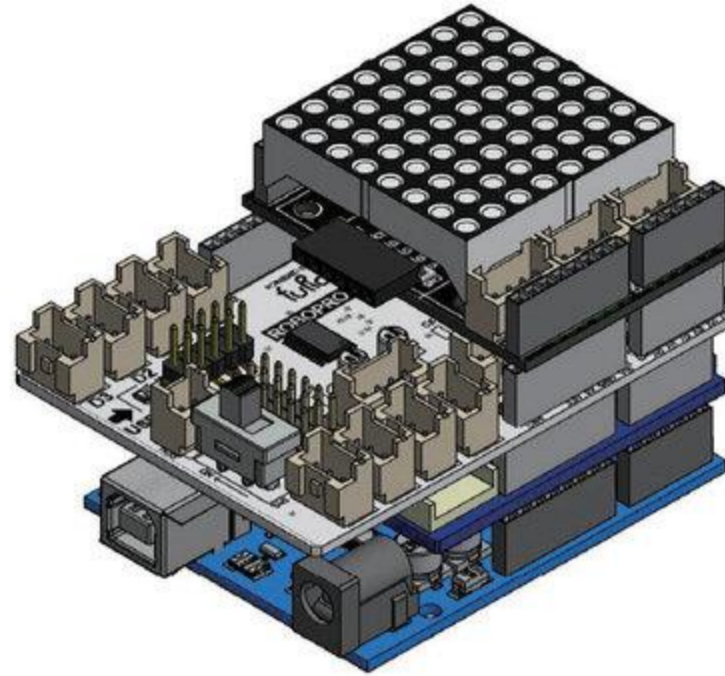


図2-3 マトリクスLEDシールドの付けかえ

準備ができたなら、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 > Inverted2 > MatrixLineTurnGZ

実行結果：はじめはマトリクスLED上をラインがスクロールする（初期化中です）。初期化が終わると、マトリクスLED上をドットの線が回転する。z軸を中心として姿勢検出ユニットの角度を変えると線の回転スピードも変わる。

傾けたり、横にふったりしてみましょう。

やってみよう！

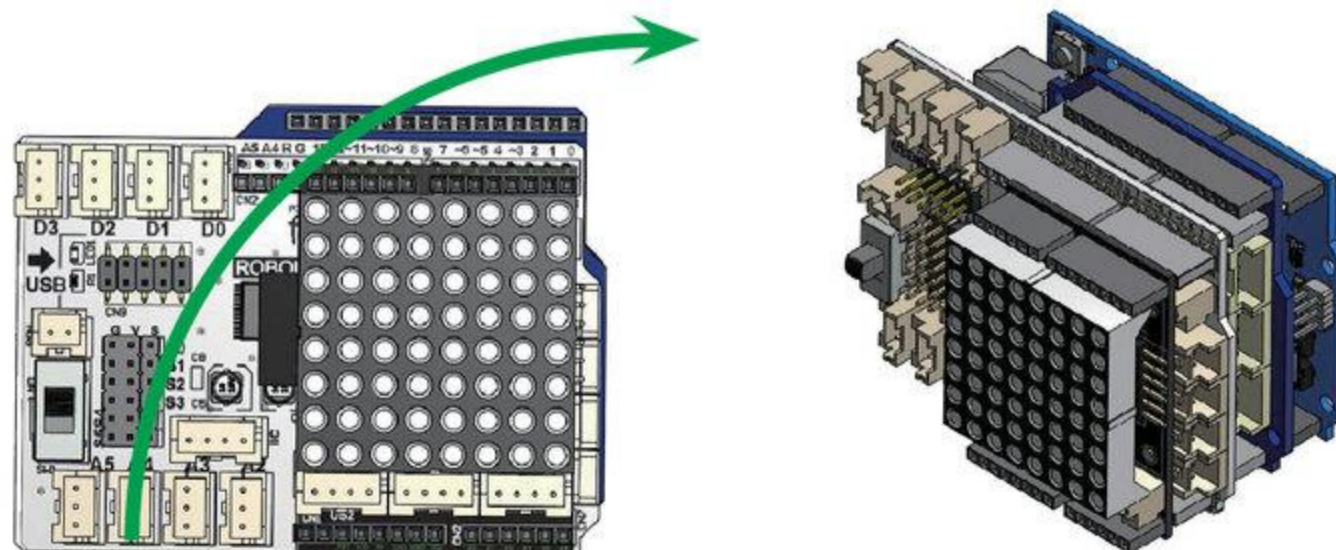
以下のプログラムを実行してみよう。

講 初期化中は動かさないでください

∞プログラムの書き込み

RoboticsProfessorCourse2 > Inverted2 > MatrixLineTurnDP

以下の図のように机に立てた状態で、初期化のためにマイコンボードのリセットボタンを押してみよう。すると、ラインがスクロールし、次に横にラインが入るよ。（ただし、きれいな直線にはならず、真ん中の部分で段差ができるよ。）



z軸を中心として横方向にコロリと回してみよう。どうなるかな？

ある一定の範囲内で回転させると、ラインが水平を保ったままになるよね。

チャレンジ課題

プログラム「MatrixLineTurnDP」は、まだ未完成だよ。ラインが動く範囲が狭かったよね。では、z軸を中心とした回転方向一周（360度）までラインが動くように完成させよう。

 ヒント

プログラム「MatrixLineTurnDP」では、if文が使われているよ。

0度から90度まで、0度から-90度まで、といった具合に180度の範囲が設定されているんだ。

```
if(rang >= 0 && rang < 90){  
    if(rang >= 0 && rang < 10) y = 7;  
    if(rang >= 10 && rang < 20) y = 6;  
    if(rang >= 20 && rang < 30) y = 5;  
    if(rang >= 30 && rang < 45) y = 4;  
    if(rang >= 45 && rang < 55) y = 3;  
    if(rang >= 55 && rang < 70) y = 2;  
    if(rang >= 70 && rang < 80) y = 1;  
    if(rang >= 80 && rang < 90) y = 0;  
    // y軸に沿ってラインが動く  
    x1 = 0;  
    x2 = 7;  
    y1 = y;  
    y2 = 7 - y;  
}
```

さて、360度の範囲で表示させるには、どうしたらよいか考えよう。基本的にはコピーでできるから、コピー後、数字だけ変更しよう。

講

解答プログラムは以下となります。

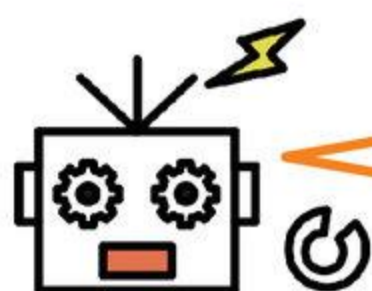
RoboticsProfessorCourse2 > Inverted2 > MatrixLineTurnDP2
プログラムに追加でかく解答例部分は巻末に記載します。

3. まとめ（目安5分）

今回は、角速度について勉強してみました。また、実際にセンサーを使いながら、加速度と角速度のそれぞれのセンサーの利点と欠点も体験できましたね。

これらをうまくロボットに取り入れていきたいですね。事実、これらのセンサーが後に作る倒立振り子の肝とうりっしんし きもになってきます。こういった癖くせがあるのかよく知っておくことはとても大切です。世の中にはいろいろなセンサーがありますが、それぞれ一長一短があります。各センサーのいいところ取りができるようになりましょう。各センサーを工夫して同時に使えるようになると、最終的に自分がやりたい結果を一番すばらしい形で得ることができます。

前回と今回ではセンサーの機能を一個ずつみてきました。次回以降は、組み合わせに関しても勉強していきましょう。



次回は、測った値の活用方法を考えよう

講

○以下の理解度を確認します。

- ・角速度を知る
- ・角速度を体感する

○次回のテーマは、「ロボットの『入力』と『出力』」です。

今回使用した基板ユニットをそのまま使いますが、追加でスピーカーが必要です。

《次回必要なもの》

- ・次回は、今回使ったパーツの他に、以下のパーツを持ってきてください。

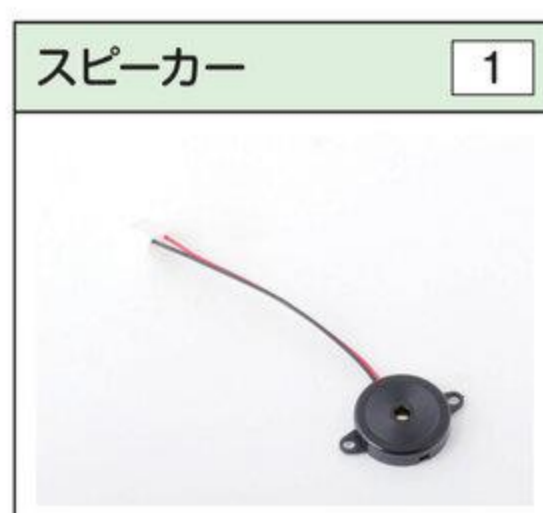


図 3-0 次回必要なもの

P.15 チャレンジ課題 解答例 (追加部分のみ記載)

```
if(rang < -90 && rang > -180){
    rang += 180;
    if(rang >= 0 && rang < 10) y = 7;
    if(rang >= 10 && rang < 20) y = 6;
    if(rang >= 20 && rang < 30) y = 5;
    if(rang >= 30 && rang < 45) y = 4;
    if(rang >= 45 && rang < 55) y = 3;
    if(rang >= 55 && rang < 70) y = 2;
    if(rang >= 70 && rang < 80) y = 1;
    if(rang >= 80 && rang < 90) y = 0;
    //y軸に沿ってラインが動く
    x1 = 0;
    x2 = 7;
    y1 = y;
    y2 = 7 - y;
}

if(rang > 90 && rang < 180){
    rang -= 180;
    if(rang < 0 && rang > -10) x = 0;
    if(rang <= -10 && rang > -20) x = 1;
    if(rang <= -20 && rang > -30) x = 2;
    if(rang <= -30 && rang > -45) x = 3;
    if(rang <= -45 && rang > -55) x = 4;
    if(rang <= -55 && rang > -70) x = 5;
    if(rang <= -70 && rang > -80) x = 6;
    if(rang <= -80 && rang > -90) x = 7;
    //x軸に沿ってラインが動く
    y1 = 7;
    y2 = 0;
    x1 = x;
    x2 = 7 - x;
}
```

上記の部分をプログラムに追加します。