

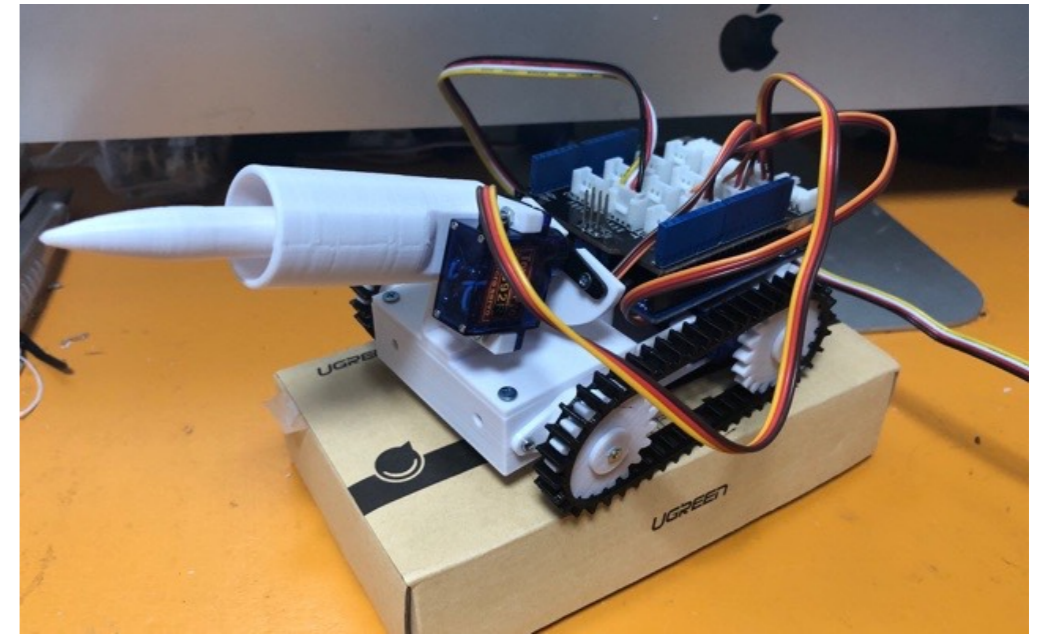
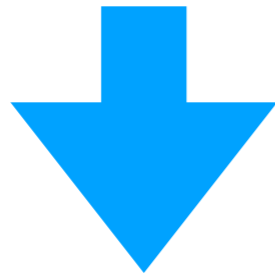
クローラロボットの改良

Arduinoで実践的なプログラムを学ぶ

クローラロボットの今のプログラムは？

- できること

- 1) ボリュームで前進・後退ができる
- 2) プッシュスイッチでミサイルの発射ができる



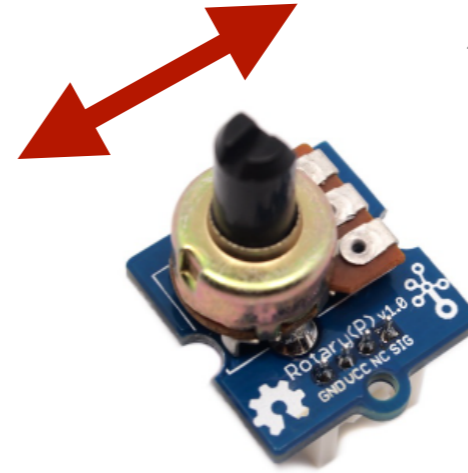
- できるようにすること

- 1) ジョイスティックで前進・後退・旋回できるようにする
- 2) スイッチ以外でミサイルを発射できるようにする
- 3) ミサイルを発射するときにLEDを点滅させる。

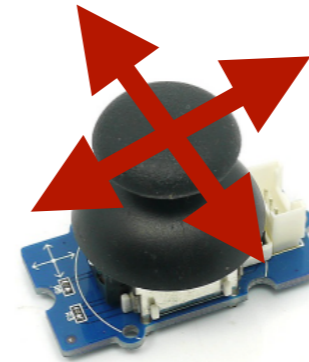
クローラロボットはどのように動いているのか？

クローラロボットは左右のモーターを回転させて動いています。

| | 左モータ | 右モータ |
|-----|------|------|
| 停止 | 停止 | 停止 |
| 前進 | 正転 | 逆転 |
| 後退 | 逆転 | 正転 |
| 左旋回 | 正転 | 正転 |
| 右旋回 | 逆転 | 逆転 |



ボリュームは1個分の大きい
小さいしか設定できない。
(回転が1方向なので)



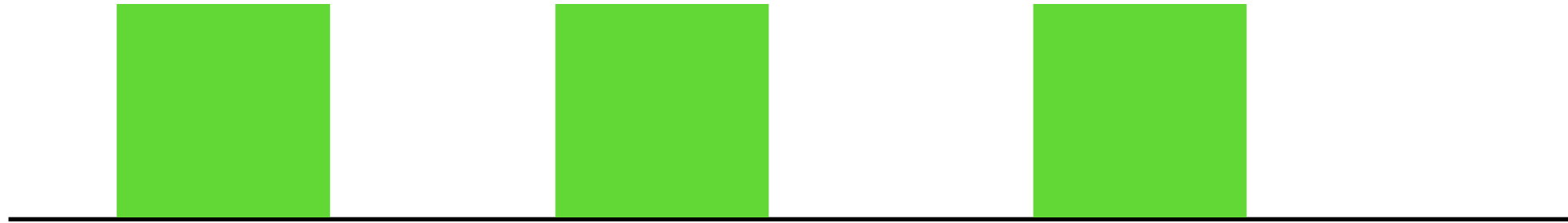
ジョイスティック前後
左右の2個分のボリューム
でできている。

ジョイスティックを使って、
前後に倒したら、それぞれ前進・後退。左右に倒したらその場で左右の旋回ができれば
自由自在にコントロールできるようになる。

サーボモータ（模型用）はどうやって制御されているのか？

入力される信号の幅で、0~180度もしくは時計回り・停止・半時計周りの回転をしています。

- ・ 基準の信号幅（1.5ミリ秒）の時は90度（もしくは停止）



- ・ 信号幅が狭い（1.5ミリ秒以下）の時は0度方向に移動（もしくは時計周りに回転）



- ・ 信号幅が広い（1.5ミリ秒以上）の時は180度方向に移動（もしくは半時計周りに回転）



信号の幅を1.5ミリ秒から細かく変化させることで、微妙な角度や速度のコントロールができる！

Arduinoにはサーボモータを簡単に制御する**ライブラリ**があります。

ライブラリとは、プログラムを機能ごとにまとめて、自分のプログラムから利用できるようにしたプログラムのこと。

サーボモータを動かすためには**わずか4行を書くだけでOK!**

`#include <Servo.h>` ← このプログラムでサーボモータのライブラリを使うよ!と宣言

`Servo motorA;` ← int や charのように Servoという変数を使うよ!と設定

`motorA.attach(2);` ← 2番ピンにServo変数のmotorAを関連付けるよ!と設定

`motorA.write(100);` ← motorAを100の位置に移動しなさいと命令

fig-1 を書き込んで、クローラーロボットの動きを見てみよう

fig1

```
#include <Servo.h>

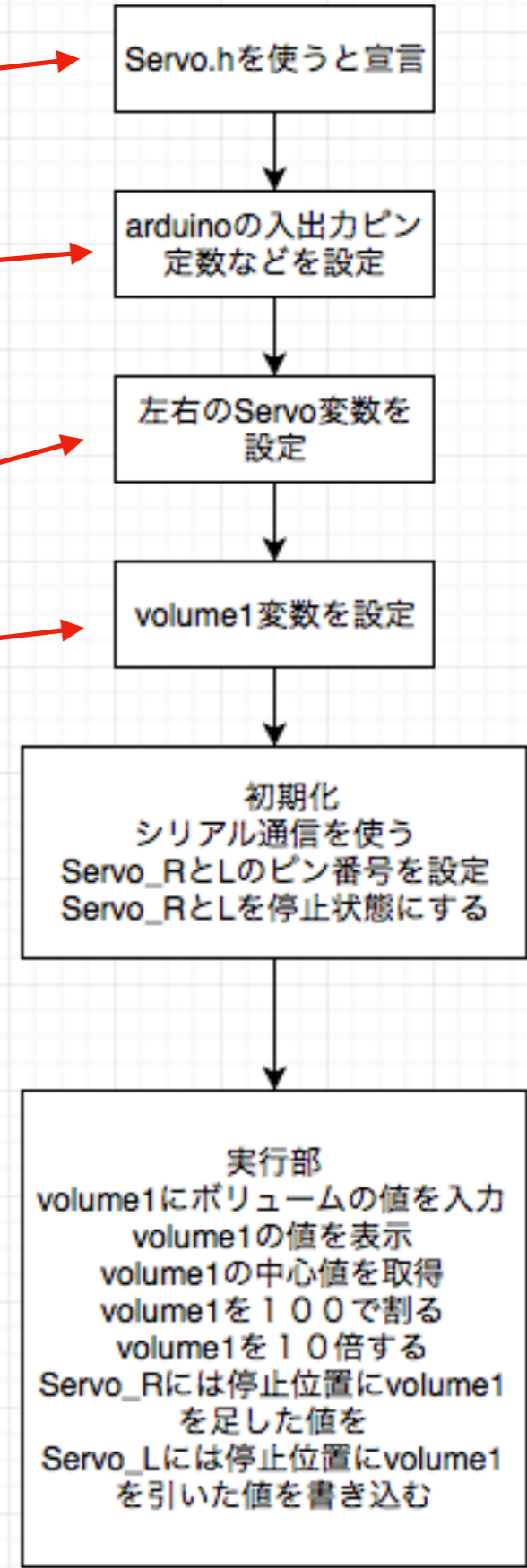
const int SERVO_L=2;
const int SERVO_R=3;
const int STOP_L=92;
const int STOP_R=90;
const int CENTOR=511;
const int VOLUME1=A0;
const int STEP=100;
const int ADD=10;

Servo Servo_R,Servo_L;

int volume1;

void setup() {
  Serial.begin(9600);
  Servo_R.attach(SERVO_R);
  Servo_L.attach(SERVO_L);
  Servo_R.write(STOP_R);
  Servo_L.write(STOP_L);
}

void loop() {
  volume1=(analogRead(VOLUME1));
  Serial.println(volume1);
  volume1=volume1-CENTOR;
  volume1=volume1/STEP;
  volume1=volume1*ADD;
  Servo_R.write(STOP_R+volume1);
  Servo_L.write(STOP_L-volume1);
}
```



ボリュームは0~1023まで値が変化するが、Servo.writeで書き込めるのは0~180まで。

ボリュームの値をそのまま0~1023

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  Serial.println(volume1);  
  volume1=volume1-CENTOR;  
  volume1=volume1/STEP;  
  volume1=volume1*ADD;  
  Servo_R.write(STOP_R+volume1);  
  Servo_L.write(STOP_L-volume1);  
}
```

中心位置の値を引いて-511~0~+511

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  volume1=volume1-CENTOR;  
  Serial.println(volume1);  
  volume1=volume1/STEP;  
  volume1=volume1*ADD;  
  Servo_R.write(STOP_R+volume1);  
  Servo_L.write(STOP_L-volume1);  
}
```

扱いやすいように100で割る(-5~0~+5)

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  volume1=volume1-CENTOR;  
  volume1=volume1/STEP;  
  Serial.println(volume1);  
  volume1=volume1*ADD;  
  Servo_R.write(STOP_R+volume1);  
  Servo_L.write(STOP_L-volume1);  
}
```

10倍する(-50~0~+50)

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  volume1=volume1-CENTOR;  
  volume1=volume1/STEP;  
  volume1=volume1*ADD;  
  Serial.println(volume1);  
  Servo_R.write(STOP_R+volume1);  
  Servo_L.write(STOP_L-volume1);  
}
```

演算することで、扱いやすい値にする！

ボリュームからジョイスティックに変更してみよう

A0に繋いでいるボリュームをジョイスティックに変更し、volume1の値がどのように変化するか見てみよう。

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  Serial.println(volume1);  
  volume1=volume1-CENTOR;  
  volume1=volume1/STEP;  
  volume1=volume1*ADD;  
  Servo_R.write(STOP_R+volume1);  
  Servo_L.write(STOP_L-volume1);  
}
```

ボリュームは0~1023の変化だったけど、
ジョイスティックは240~760までしか
変化しない??



つなぐものによって値の範囲は変わる。
(プログラムを作る前に確認しておこう)

ボリュームと同じように変化させるためには、`const int STEP=100;`を
`const int STEP=50;`にする。

ジョイスティックの左右の信号はどこに入るの？

A0にジョイスティックの前後が割り当てられ、A1にジョイスティックの左右が割り当てられる。

fig1をfig3の様に変更して確認してみよう！

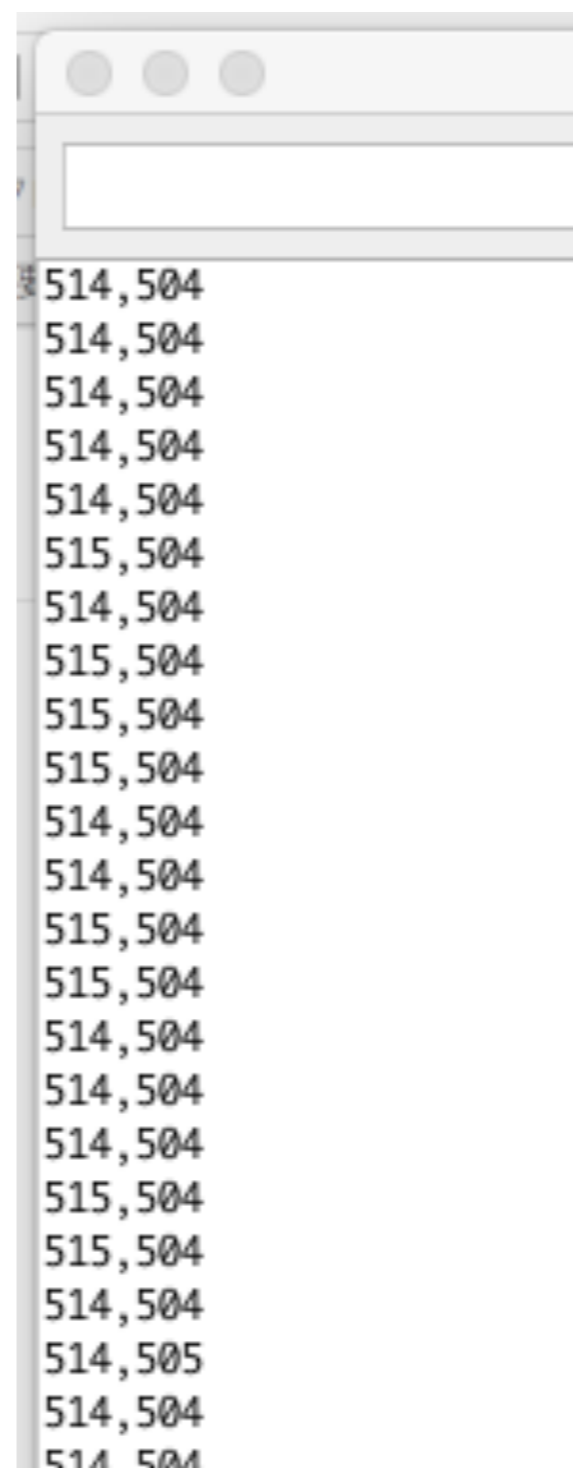
```
const int CENTOR=511;
const int VOLUME1=A0;
const int VOLUME2=A1; ←
const int STEP=100;
const int ADD=10;

Servo Servo_R,Servo_L;

int volume1,volume2; ←

void setup() {
  Serial.begin(9600);
  Servo_R.attach(SERVO_R);
  Servo_L.attach(SERVO_L);
  Servo_R.write(STOP_R);
  Servo_L.write(STOP_L);
}

void loop() {
  volume1=(analogRead(VOLUME1)); ←
  volume2=(analogRead(VOLUME2)); ←
  Serial.print(volume1); ←
  Serial.print(","); ←
  Serial.println(volume2); ←
}
```



514,504
514,504
514,504
514,504
514,504
515,504
514,504
515,504
515,504
515,504
514,504
514,504
515,504
515,504
514,504
515,504
515,504
514,504
514,505
514,504
514,504

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

```
const int SERVO_L=2;  
const int SERVO_R=3;  
const int STOP_L=92;  
const int STOP_R=90;  
const int CENTOR=511;  
const int VOLUME1=A0;  
const int VOLUME2=A1;  
const int STEP=100;  
const int ADD=10;
```

```
Servo Servo_R,Servo_L;
```

```
int volume1,volume2;
```

```
void setup() {  
  Serial.begin(9600);  
  Servo_R.attach(SERVO_R);  
  Servo_L.attach(SERVO_L);  
  Servo_R.write(STOP_R);  
  Servo_L.write(STOP_L);  
}
```

```
void loop() {  
  volume1=(analogRead(VOLUME1));  
  volume2=(analogRead(VOLUME2));  
  Serial.print(volume1);  
  Serial.print(",");  
  Serial.println(volume2);
```

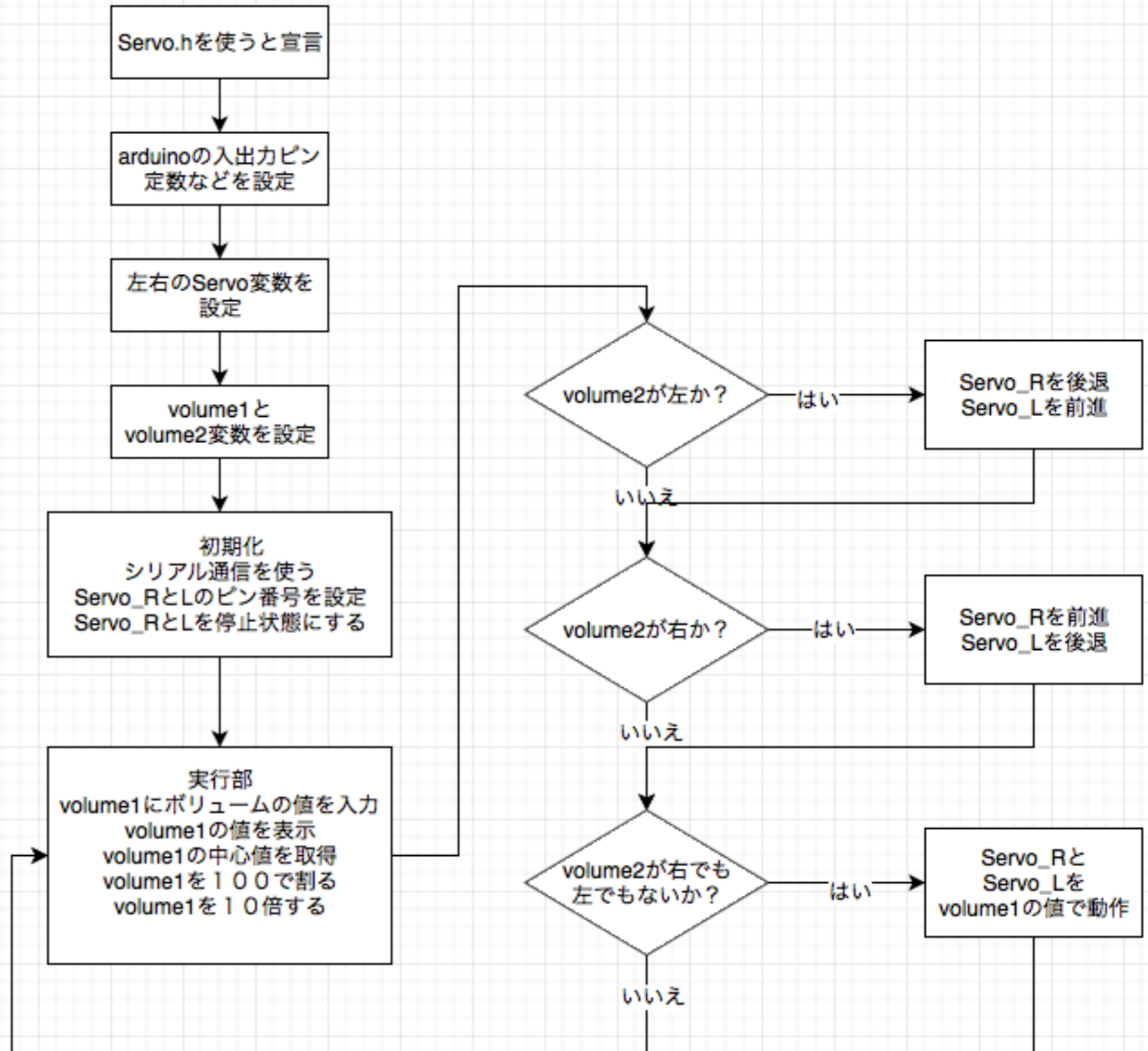
```
  volume1=volume1-CENTOR;  
  volume1=volume1/STEP;  
  volume1=volume1*ADD;
```

```
  if (volume2<[redacted]) {
```

```
  if (volume2>[redacted]) {
```

```
  if (volume2>[redacted] && volume2<[redacted]) {  
    Servo_R.write(STOP_R+volume1);  
    Servo_L.write(STOP_L-volume1);
```

fig-3を改造して、旋回動作をするようにプログラムを変更してみよう



ミサイルを発射できるようにしてみよう！

ミサイルのサーボはD4に、ミサイルの発射スイッチはD5に設定してプログラムを考えてみよう。

【必要なこと】

- `const int` でスイッチのD5、ミサイルのサーボD4や停止状態位置、発射位置の数値を割り付ける。
- ミサイルのサーボ変数を追加する。
- ミサイル発射スイッチのint変数を追加する。
- ミサイルのサーボ変数をD4にattachする。
- ミサイルのサーボ位置を停止位置に書き込む。
- if文を使って、ミサイルスイッチが押されたか判定し、押されたら、
 - ミサイルサーボを発射位置に移動。
 - 1秒待つ（動作を確実にするため）
 - ミサイルサーボを元の停止位置に戻る

**これで今回のセミナーは終わりです。
気づいたこと、疑問点などをみんなで話し合おう。**