

# 電気工作クラブ アルデュイーノ Arduinoマイコン入門

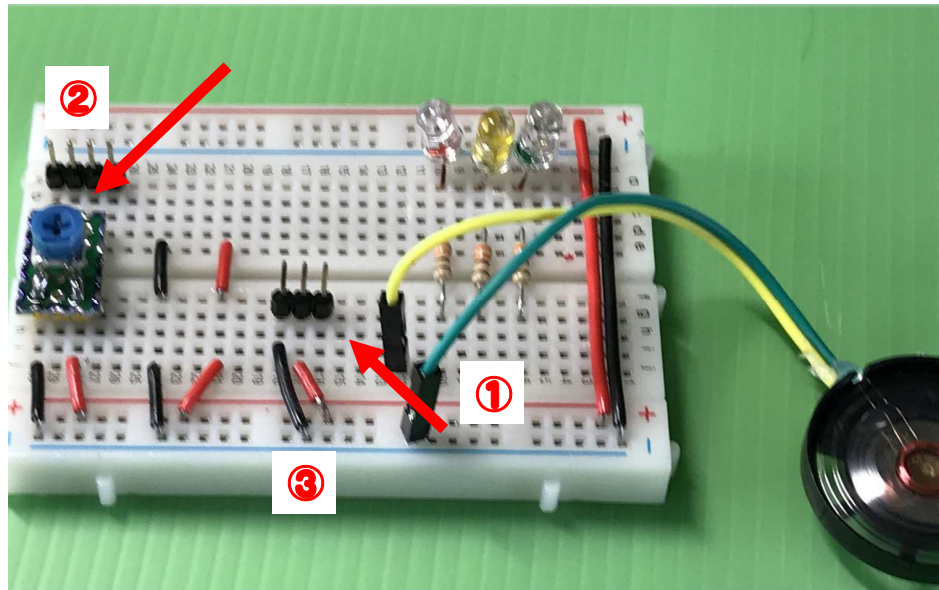
Arduino マイコンの基礎

第 4 回

## 第5回 Arduino で音を出す、サーボモーターを動かす

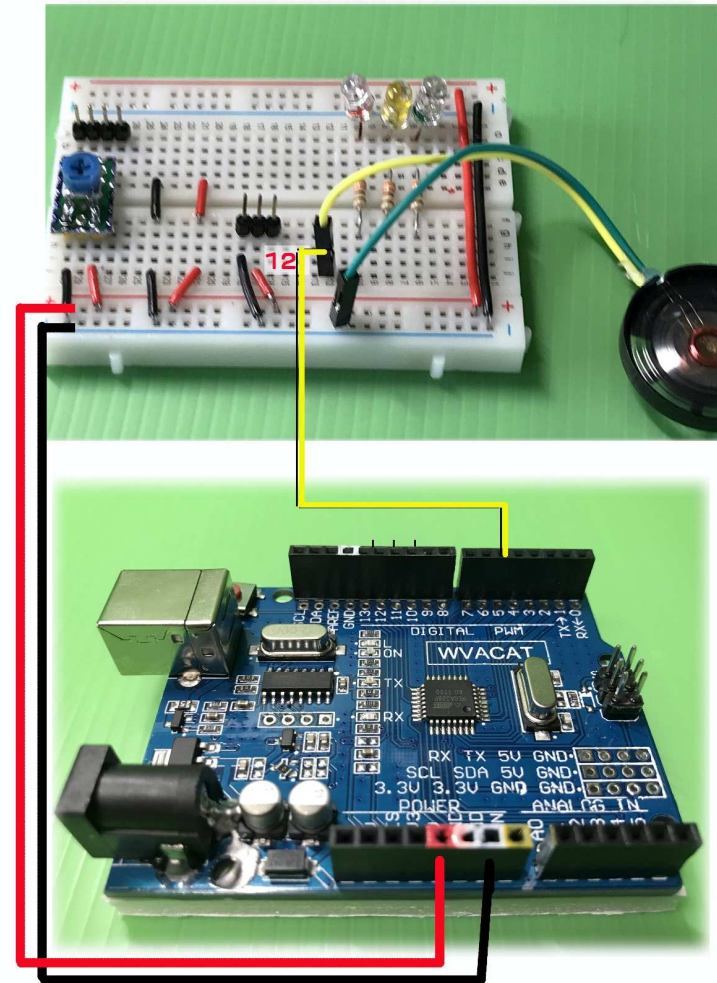
### ブレットボードの配線を変えます

- ① スピーカーの配線  
12番(黄色) - 線(緑色)
- ② ボリウムを取り付ける  
28番 g (赤色側) 30番 g(黒色側)  
29番 e
- ③ 赤・黒色ジャンパー線  
28番(赤色) 30番(黒色)  
17番赤色 16番から移動)  
3本ピン端子を付ける



## Arduino で音を出す

スピーカーの黄色線を Arduino uno 5番ピンに接続  
緑色の線は マイナス線 または 13番ピン横の  
グランドピンにつなぐ



## 音を出すスケッチ【tone1. ino】

```
void setup()
{
  pinMode(5, OUTPUT); // D5 にスピーカーつなぐ。
}
int sounds[] = {200,300,400,600,800,1200,1600,2400};
int v = 0;

void loop()
{
  tone(12,sounds[v],10);
  delay(50);
  v++;
  if(v > 7){
    v = 0;
  }
}
```

## 音を出すスケッチ【tone2. ino】

```
#define ONPU 600//四分音符
#define ONPU2 800//二分音符
#define ONPU3 100 //三連符
#define SPEAKER 5 //スピーカーの出力ピン番号 5
void setup() {
}
void loop() {
  // tone (出力ピン番号,周波数(Hz),音の長さ(ミリセカンド))
  tone(SPEAKER,523,ONPU3); // ド
  delay(200);
  tone(SPEAKER,523,ONPU3); // ド
  delay(200);
  tone(SPEAKER,523,ONPU3); // ド
  delay(200);
  tone(SPEAKER,523,ONPU2); // ド
  delay(600);
  tone(SPEAKER,415,ONPU); // ラ b
  delay(600);
  tone(SPEAKER,466,ONPU); // シ b
  delay(600);
  tone(SPEAKER,523,ONPU3); // ド
  delay(200); // 8 分音符として付与
  delay(200);
  tone(SPEAKER,466,ONPU); // シ b
  delay(200);
  tone(SPEAKER,523,ONPU2); // ド
  delay(1200);
}
```

音階と周波数の関係		
単位:ヘルツ		
ド	262 (523)	261.626
レ	294 (587)	293.665
ミ	330 (659)	
ファ	350 (698)	
ソ	392 (784)	
<b>ラ</b>	<b>440 (880)</b>	
シ	494 (987)	
ド	523 (10475)	

正確な値は調べてください

## Arduino UNO で サーボモーターを動かす

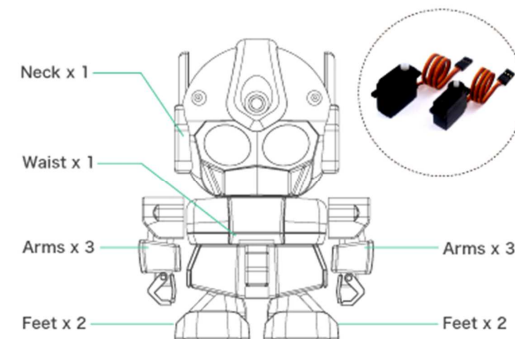
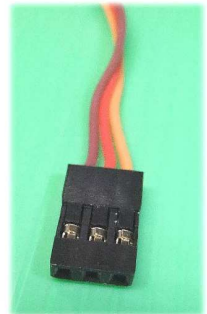


- 右から
- 1 オレンジ色 信号線
  - 2 赤色 プラス5V
  - 3 茶色 マイナス

## 「ラピロ」というロボットです

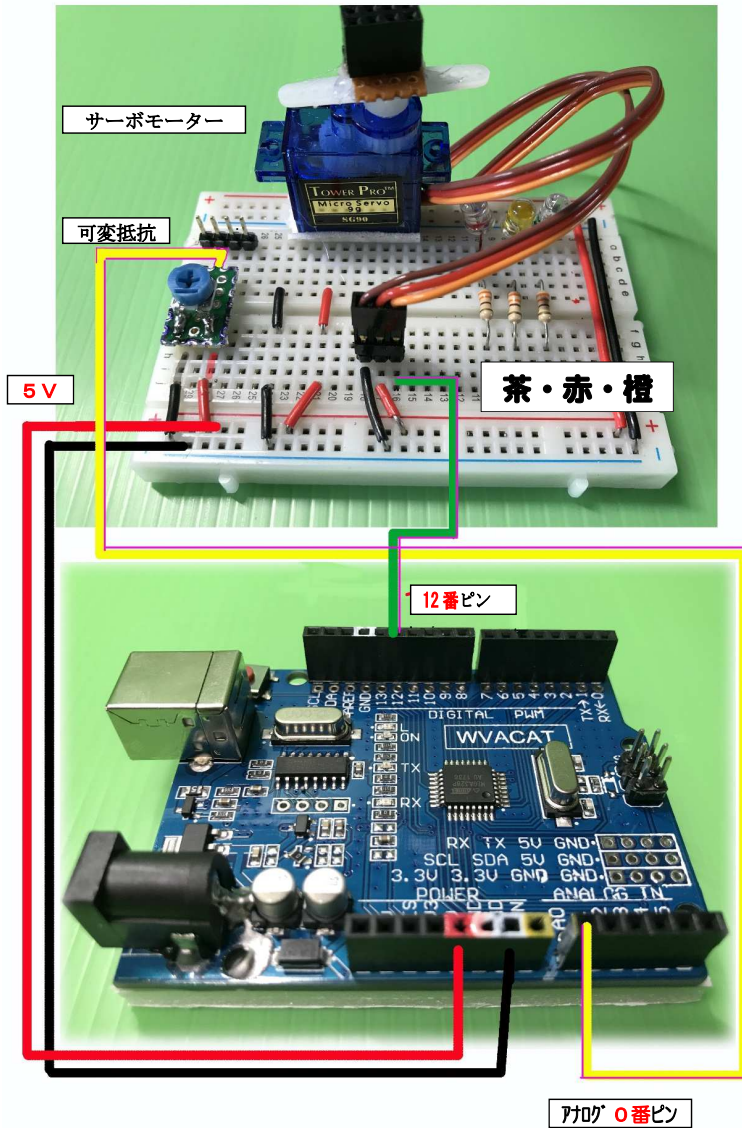
首に1つ、腰に1つ、  
両足に2つずつ、両腕に3つずつ、  
計12個のサーボモーターがついています。

それぞれが正しく動いて  
向きを変えたり手をあげたりできるのです。



## サーボモーターを ArduinoUNO につなく

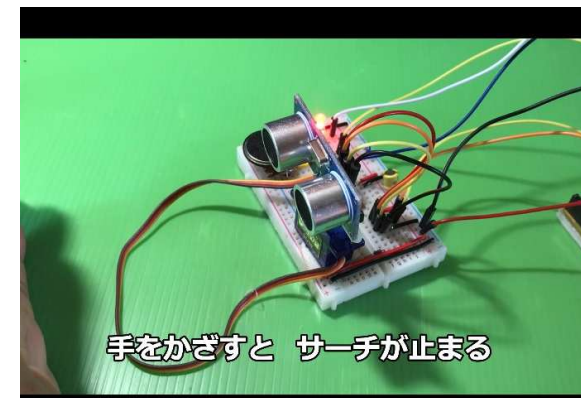
可変抵抗で サーボモーターを動かす



## スケッチ【\_2018servo1】

を使ってサーボを動かしてみましょう

```
1 #include<Servo.h>
2 Servo myservo;
3 int val;
4
5 void setup()
6 {
7   myservo.attach(12); //デジタル 12番ピンをサーボの角度命令出力ピンとして
   設定
8 }
9
10 void loop()
11 {
12   val = 0;
13   myservo.write(val); //サーボを動かす(0度)
14   delay(1000);
15   val = 90;
16   myservo.write(val); //サーボを動かす(90度)
17   delay(1000);
18   val = 180;
19   myservo.write(val); //サーボを動かす(180度)
20   delay(1000);
21   val = 90;
22   myservo.write(val); //サーボを動かす(90度)
23   delay(1000);
24 }
```



## スケッチ『\_2018servomotor\_VOL2』

みず色の部品(4.7kΩの半固定ボリューム)

プラストライバーで回すとサーボの角度が変わります

```
1 #include <Servo.h>
2
3 #define SERVOPIN    (12)
4 #define VOLUMEPIN  (0)
5
6 Servo g_servo;
7
8 void setup()
9 {
10     g_servo.attach(SERVOPIN);
11 }
12
13 int Map( int iIn, int iIn1, int iIn2, int iOut1, int iOut2, boolean bConstrain = false )
14 {
15     double dValue = (double)(iIn - iIn1) * (iOut2 - iOut1) / (iIn2 - iIn1) + iOut1;
16     int iValue = (0 < dValue) ? (int)(dValue + 0.5) : (int)(dValue - 0.5);
17     if( bConstrain )
18     {
19         int iOutMin, iOutMax;
20         if( iOut1 < iOut2 )
21         {
22             iOutMin= iOut1;
23             iOutMax= iOut2;
24         }
25         else
26         {
27             iOutMin= iOut2;
28             iOutMax= iOut1;
29         }
30         if( iOutMin > iValue )
31         {
32             return iOutMin;
33         }
34         if( iOutMax < iValue )
35         {
36             return iOutMax;
37         }
38     }
39     return iValue;
40 }
41
42 void loop()
43 {
44     int iValue = analogRead(VOLUMEPIN);
45     int iAngle = Map(iValue, 0, 1023, 0, 180, true);
46     g_servo.write(iAngle);
47     delay(15);
48 }
```



## 超音波レーダー

### スケッチ【\_2018Ultrasonic\_radar】

ブレッドボードの結線を確認してください

これまでの結線のままでは うまく動きません

超音波センサーのトリガー	8 番端子(橙色の線)
超音波センサーの受信信号	9 番端子(黄色の線)
サーボモーターの制御信号	12 番端子(緑色の線)
音 (スピーカー)	6 番端子
LED (赤・黄・青 何色でも良い)	5 番端子

※ここからスケッチ スタート

```
1 #include <Servo.h>
2 #define pinLED 5 //距離設定した LED の点灯表示
3 #define PINNO 6 // スピーカを接続したピン番号
4 #define trigPin 8 //距離センサーのトリガ信号
5 #define echoPin 9 //距離センサーのエコー信号
6 #define BEAT 300 // スピーカーから出る音の長さを指定
7 Servo myservo; // create servo object to control a servo
8 int pos = 0; // サーボモーターの位置(角度)を表す数値
9 int duration, distance; //エコー信号から計算した距離数値
10 int old_distance = 10; //設定距離 8 cm
11
12 void setup()
13 {
14     myservo.attach(12); //サーボモーター制御ピン番号
15     Serial.begin(9600); //シリアル信号スピード
16     pinMode(trigPin, OUTPUT);
17     pinMode(echoPin, INPUT);
18     pinMode(pinLED, OUTPUT);
19 }
20 void loop()
21 {
22     for (pos = 0; pos <= 180; pos += 1) //位置 0 から 180 になるまでステップ 1 で
```

```

23 digitalWrite(pinLED, LOW);
24 myservo.write(pos);
25 Eloop();
26 while (distance <= old_distance) {
27     Eloop();
28     digitalWrite(pinLED, HIGH);
29     tone(PINNO, 392, BEAT + BEAT); // ソ
30     //     delay(BEAT);
31     //     tone(PINNO, 440, BEAT); // ラ
32     //     delay(BEAT);
33     //     tone(PINNO, 494, BEAT + BEAT); // シ
34     //     delay(BEAT + BEAT);
35     //delay(20);
36 }
37 }
38 for (pos = 180; pos >= 0; pos -= 1) //位置 180 から 0 になるまでス
    テップ 1 で
39 {
40     digitalWrite(pinLED, LOW);
41     myservo.write(pos);           // サーボの位置を 'pos' に入れ
    ておく
42     Eloop();
43     while (distance <= old_distance) {
44         Eloop();
45         digitalWrite(pinLED, HIGH);
46         tone(PINNO, 494, BEAT + BEAT); // シ
47         //     delay(BEAT + BEAT);
48         //     tone(PINNO, 440, BEAT); // ラ
49         //     delay(BEAT);
50         //     digitalWrite(pinLED, HIGH);
51         //     tone(PINNO, 392, BEAT + BEAT); // ソ
52         //delay(BEAT);
53         // delay(20);
54     }

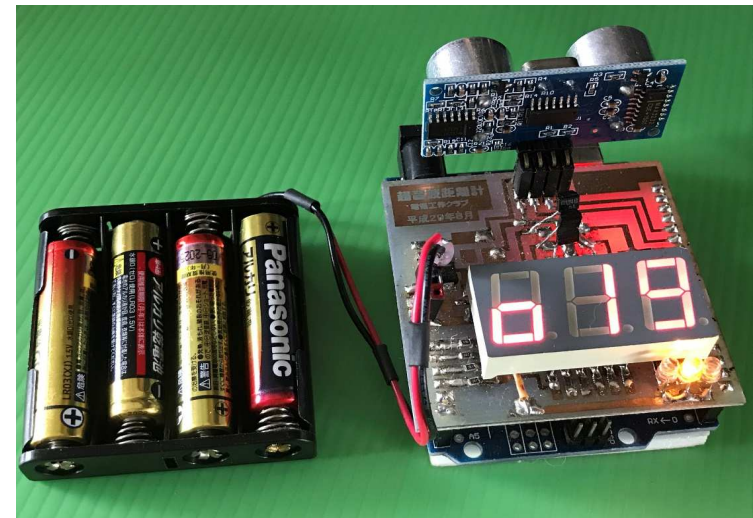
```

```

55 }
56 }
57 void Eloop() {
58     digitalWrite(trigPin, HIGH);
59     delayMicroseconds(1000);
60     digitalWrite(trigPin, LOW);
61     duration = pulseIn(echoPin, HIGH);
62     distance = (duration / 2) / 29; //音速から距離を計算する式
        // Serial.print(pos);           //シリアルモニターに POS と表示 以
        下半角(英数字しか受け付けない
63     // Serial.print(" deg ");        //角度を表示
64     // Serial.print(distance);        //distance と表示
65     // Serial.println(" cm");        //距離を表示
66     delay(2);
67 }

```

「//」を消すと 出る音が増えます



次回は第 5 回(8 月 4 日) 「超音波距離計シールド」を完成させます、  
「ArduinoNano」の使い方を実習します。

※ ArduinoNANO にピンを付けてきます。一時預かり()