7.1　センサ実験１

本実験は２週で構成されている。１週目は、超音波センサとサーボモータの動作原理と使い方を学ぶ。２週目は、それらを統合したアプリケーションシステムを作る。いずれも、これまでの実験で習得した技術を応用することが必要となる。

本実験の達成目標を以下に示す。

１週目：

- 超音波センサの動作原理を説明できる
- 超音波センサを使用できる
- 超音波センサを用いたアプリケーションシステムを作成できる
- サーボモータの動作原理を説明できる
- サーボモータを使用できる
- サーボモータを用いたアプリケーションシステムを作成できる

２週目：

- 超音波センサとサーボモータを統合したアプリケーションシステムを作成できる

7.1.1　超音波

超音波（ultrasonic）は、人間の耳では聴こえない周波数の音波である。一般的に、人間に聴こえる音の周波数は、約 20Hz～20kHz であり、これを可聴周波数という。可聴周波数より高い周波数の音波を超音波と呼ぶ。

超音波の性質を以下に示す。

（１）伝搬に媒質が必要

波とは、振動が伝搬する現象である。よって、振動する物体（媒質）がなければ、波は伝搬しない。音波は、媒質の圧力変化が波となって伝わる現象である。媒質の存在しない真空中では音は伝わらない。空気などの気体だけでなく、水などの液体、鉄などの固体も、音波の媒質となる。

（２）伝搬速度が媒質により変化

音波が伝わる速さ（音速）は、媒質によって異なる。例えば、空気中の音速は約 340m/s、水中では約 1,530m/s、鉄では約 5,000m/s である。光速は 30 万 km/s なので音波は伝搬速度の遅い波といえる。気体中ではエネルギーが減衰しやすく固体では効率よく伝搬する。媒質の状態や温度によっても音速は変化する。

（３）反射・回折

音波は物質に当たると反射し、回折して障害物の陰に回り込んで進む性質を持つ。音波の波長が長いほど回折しやすくなる。

（４）指向性が高い

音波は周波数が高いほど指向性が高くなる。指向性とは、方向に対する波の伝わり方の性質である。超音波は、可聴周波数域の音波よりも指向性が高いため、狭い範囲に波が直線的に伝搬する。

超音波の応用例を以下に挙げる。

（1）距離の測定

超音波は指向性が高いため、距離計測に用いるのに適している。超音波を発射し、反射波が返ってくるまでの時間を計測することで距離を求める。魚群探知機、潜水艦のソナーも同様の原理に基づく。

（2）非破壊検査

超音波を固体に発射し、伝搬の状態の違いを観測することにより、固体内部の状態を知ることができる。対象物を壊さなくてよいので、原子炉内部の調査や、エコー診断装置などの医療機器に利用されている。

（3）超音波洗浄機

超音波を液体に発射すると、液体が激しく振動し微小な真空の泡が発生する。この泡が破裂する衝撃で汚れを引き剥がす。超音波メガネ洗浄機や超音波歯ブラシはこの原理に基づいている。

（4）超音波カッター

刃に超音波振動を加えると、刃と切断物との摩擦が小さくなり、切れ味が増すという性質を応用した工具である。医療用に超音波メスなど。

（5）超音波加湿器

水に超音波振動を加えて細かく粉砕し、扇風機で空気中に拡散して加湿する。蒸気を発生させるスチーム方式と比べて水の粒子が小さいので拡散しやすく、加熱しないので省電力という利点がある。

7.1.2　距離計測の原理

超音波による距離計測の原理を以下に示す。

（1）発信機から超音波を送波

（2）超音波は対象物に反射

（3）受信機で反射波を受信

（4）超音波の送波から反射波の受信までに要した時間を距離に変換

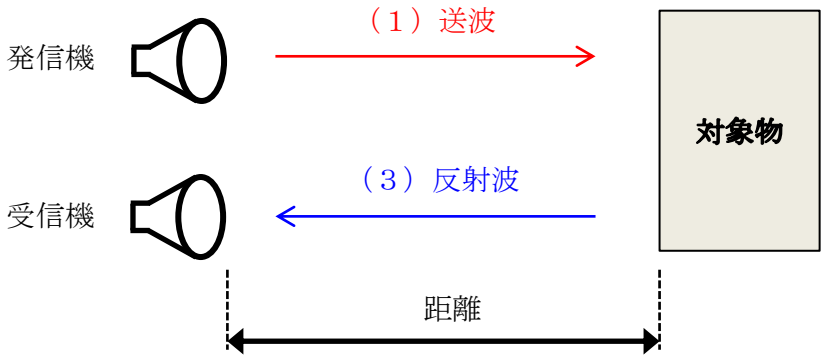

図 7.1.1：超音波による距離計測

超音波の発信機から対象物までの距離 d[m] は、音速を c[m/s]、発信から受信までの時間を t[s] とすると、式（7.1.1）で求めることができる[1]。

$$\text{距離 } d[\text{m}] \;=\; \text{音速 } c[\text{m/s}] \;\times\; \text{時間 } t[\text{s}] \;/\; 2 \qquad \cdots (7.1.1)$$

ここで、音速 c は媒質の状態によって変化することに注意しなければならない。

標準状態の乾燥空気における、音速に対する温度の影響を図 7.1.2 に示す。温度が上昇するのに比例して、音速も上昇する。

音速は、簡便的に温度のみの一次式(7.1.2)で近似することができる。k は摂氏（℃）である。

$$\text{音速 } c[\text{m/s}] \;=\; 0.61 \;\times\; k[\text{℃}] \;+\; 331.5 \qquad \cdots (7.1.2)$$

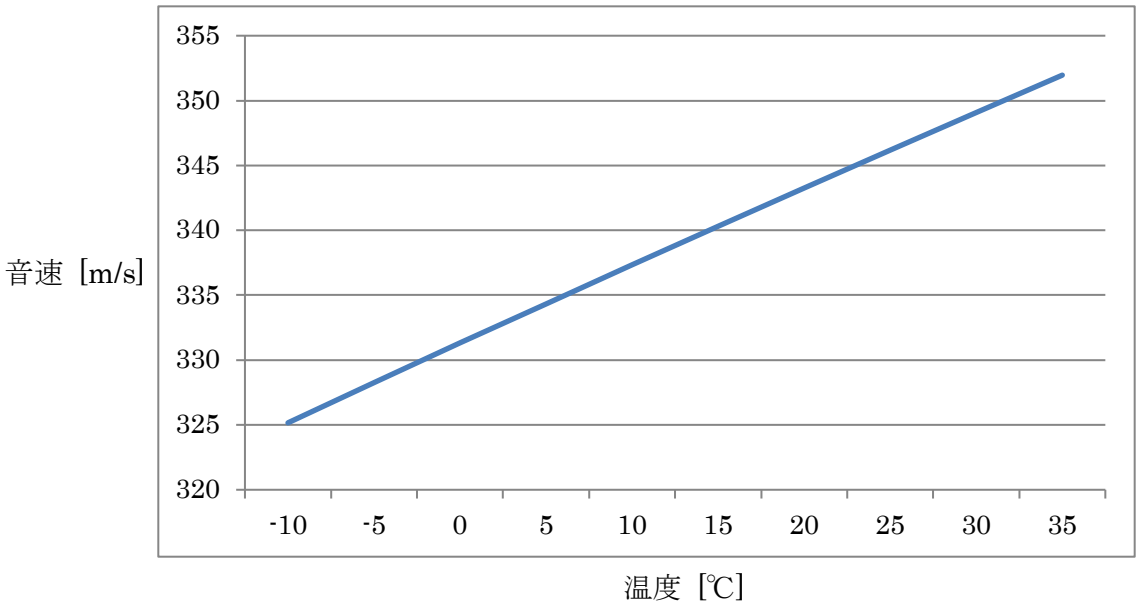

図 7.1.2：音速の温度依存性

7. 1. 3　超音波センサ（HC-SR04）の仕様

本実験で用いる超音波センサ（HC-SR04）を図 7.1.3 に示す。左側の円筒が超音波の発信機（超音波振動子）、右側の円筒が受信機である。HC-SR04 は４つのピン（Vcc, Trig, Echo, GND）を備える。

各ピンの役割は以下のとおりである。

- Vcc：電源電圧 5V
- Trig：トリガー信号入力
- Echo：エコー信号出力
- GND：グランド電圧 0V

[1] 時間 t は往復に要する時間であるため２で割る

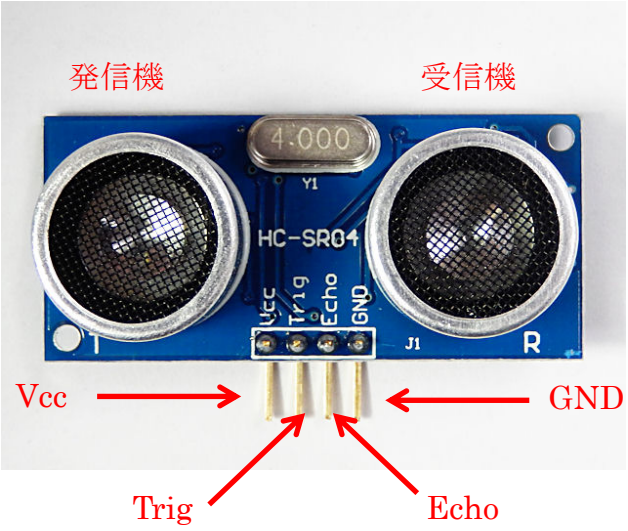

図 7.1.3：HC-SR04 のピン配置

表 7.1.1：HC-SR04 の仕様

項目	内容
電源電圧	DC 5V
動作電流	15 mA
超音波周波数	40 kHz
測定距離	2 cm ～ 400 cm
測定角度	基板正面を中心に 15 度
分解能	0.3 cm
トリガー信号	10μs
エコー信号	反射（往復）時間
サイズ	45mm ×20 mm × 15 mm

HC-SR04 は、以下のように動作する。図 7.1.4 に動作の概要を示す。

（1）Trig 信号を 10μs 以上 HIGH レベルに保った上で、LOW レベルに下げる。

（2）（1）の終了と同時に、8 個のピークをもつ 40kHz の超音波が発信機から送波される。

（3）（2）の終了と同時に、Echo 信号は HIGH レベルを出力する。

（4）対象物に反射した超音波を受信機が受信すると、Echo 信号は LOW レベルを出力する。

つまり、「Echo 信号が HIGH である時間」が、「超音波センサから対象物までの距離を、超音波が往復した時間」に等しくなる。

正確に動作させるため、以下の点に注意すること。

- 再び Trig 信号を HIGH レベルにする際、60ms 以上の間隔をあけること（HC-SR04 の仕様）
- 対象物（距離の検出対象）は、硬い反射面を持つ平面のものが望ましい

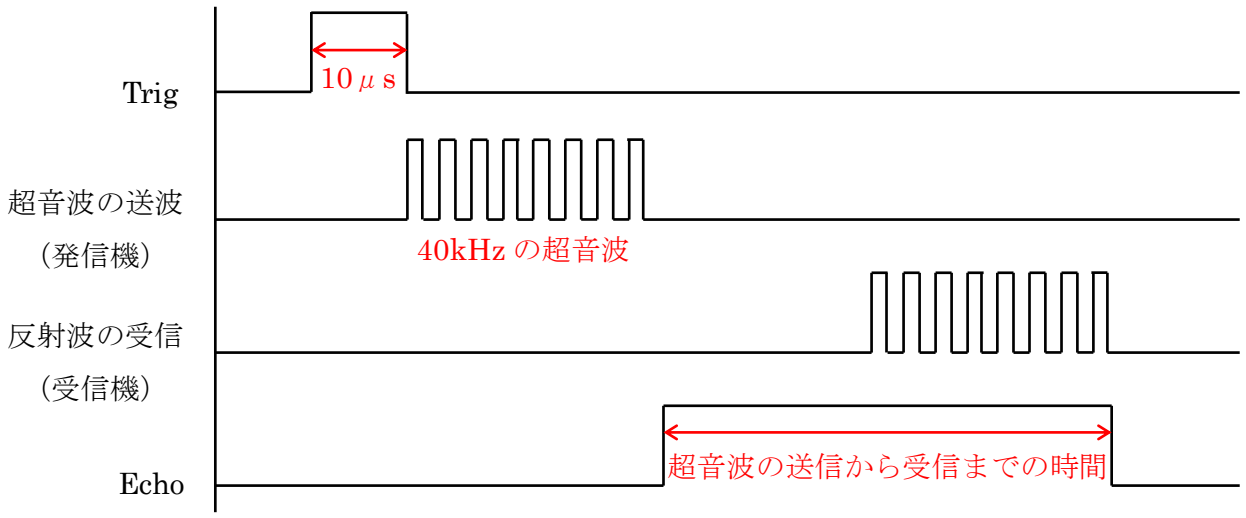

図 7.1.4：HC-SR04 のタイミングチャート

警告：

基板の裏の金属部分を、指や金属で触れないこと。配線がショートし故障する。机に置く際は、前面（発信機と受信機がある面）を底にせよ。超音波センサを破壊した場合は、弁償の対象となる。

＜演習 7. 1. 1＞

「超音波センサから発射された超音波が、対象物に反射して返ってくるまでの時間」を計測してみよう。

（1） 図 7.1.5 の回路をブレッドボードに組む。作図の都合上、超音波センサが正面を向いているが、実際に配線するときは、裏側に向けたほうがよい（配線に超音波が反射してしまうかもしれない）。

※Arduino と PC はまだ接続しない

- 超音波センサの Trig ピンは Arduino のデジタル 7 番ピンに接続(デジタルピンならどこでも可)
- 超音波センサの Echo ピンは Arduino のデジタル 8 番ピンに接続(デジタルピンならどこでも可)
- 超音波センサの Vcc ピンを Arduino の 5V ピンに接続
- 超音波センサの GND ピンを Arduino の GND ピンに接続

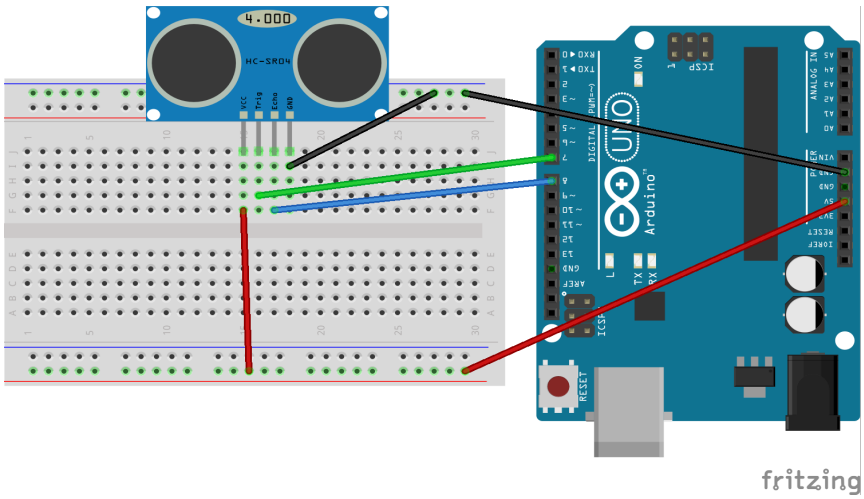

図 7.1.5 : 演習 7. 1. 1 のブレッドボード

（2） プログラム 7. 1. 1 を Arduino IDE で作成

（3） Arduino と PC を USB ケーブルで接続

（4） プログラムを「検証」→「マイコンボードに書き込む」

（5） シリアルモニタを開く

（6） 超音波センサに手をかざして、シリアルモニタの表示がどのように変わるか観察せよ

プログラム 7. 1. 1

```
const int trig = 7; //Trig ピンをデジタル 7 番に接続
const int echo = 8; //Echo ピンをデジタル 8 番に接続
unsigned long interval;   //Echo のパルス幅

void setup() {
  Serial.begin(9600);
  pinMode(trig, OUTPUT);    //7 番を出力ポートに設定
  pinMode(echo, INPUT);     //8 番を入力ポートに設定
}

void loop() {
  //10μs のパルスを Tring ピンに出力
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  //Echo 信号が HIGH である時間(μs)を pulseIn 関数で計測
  //10000μs 以上経過したら、超音波が反射して返ってこないとみなして 0 を返す
  interval = pulseIn(echo, HIGH, 10000);

  Serial.println(interval); //超音波の往復時間をシリアルモニタに表示
  delay(60);                //次の Trig 信号の出力まで 60ms 待機
}
```

プログラム 7.1.1 では、pulseIn 関数によって、Echo 信号が HIGH レベルである時間を計測している。pulseIn 関数の書式を以下に示す。

pulseIn(pin, value, timeout)

ピン(pin)に入力されるパルスを検出する。パルスの種類(value)を HIGH に指定した場合、pulseIn 関数は入力が HIGH に変わると同時に時間の計測を始め、また LOW に戻ったら、そこまでの時間(つまりパルスの長さ)をマイクロ秒単位で返す。タイムアウト(timeout)を指定した場合は、その時間を超えた時点で 0 を返す。この関数で計測可能な時間は、経験上、10μs から 3 分である。あまりに長いパルスに対してはエラーとなる可能性がある。

【引数】

pin: パルスを入力するピンの番号

value: 測定するパルスの種類。HIGH または LOW。

timeout（省略可）: タイムアウトまでの時間(μs)。省略した場合は 1 秒。

【戻り値】

パルスの長さ(μs)。パルスがスタートする前にタイムアウトとなった場合は 0 (unsigned long)。

ここで、μs（マイクロ秒）は 10^{-6} 秒（0.000001 秒）である。

7.1.4 サーボモータの動作原理

サーボ（servo）[2]とは、物体の位置、方位、姿勢などを制御量として、目標値に追従するように自動的に動作する機構である。サーボモータは、回転軸の角度を、指定した角度になるように制御させるもので、工作機械や産業用ロボットなど、様々な装置に用いられている。

サーボモータの回転制御の概要を図 7.1.6 に示す。

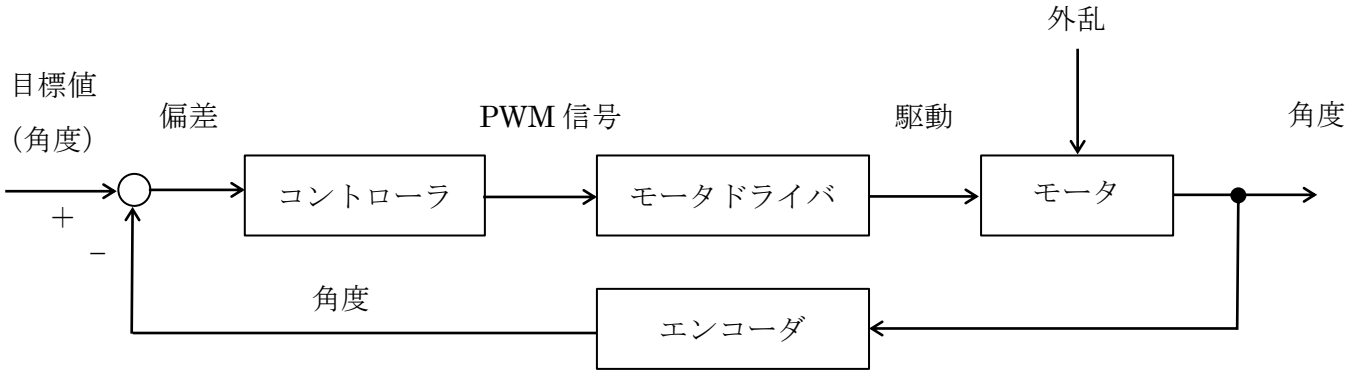

図 7.1.6：サーボモータの制御

サーボモータの制御の流れを以下に示す。

（1）サーボモータに対して、回転の目標値（角度）を入力

（2）コントローラは、角度を電圧（PWM 信号）に変換

（3）モータドライバは、モータを駆動

（4）モータが回転する。この時、外乱（電圧の変化や、回転軸に繋がっている物体の重さやバランスの変化など）の影響を受け、目標値まで回転しないことがある。

（5）エンコーダは、現在の回転角度を検出し、目標値との偏差をコントローラにフィードバックする

（6）目標値とフィードバック値の偏差が一定値以下になると回転が止まる

このような制御の方法を、サーボ制御という。

7.1.5 サーボモータ（SG90）の仕様

本実験で用いるサーボモータ(SG90)を図 7.1.7 に示す。SG90 には 3 本のサーボホーンが付属しており、回転軸に取り付けることができる。コネクタのピン配置を図 7.1.8 に示す。ピンの役割は以下のとおりである。

- Vcc 線（赤）：電源電圧 4.8 V (～5.0 V)
- GND 線（茶）：グランド電圧 0 V
- 信号線（黄）：PWM 信号

[2] サーボ（servo）の語源は servant（召使い）

SG90 は信号線に PWM 信号を入力することによって、任意の角度に回転軸をコントロールできる。

図 7.1.7：SG90

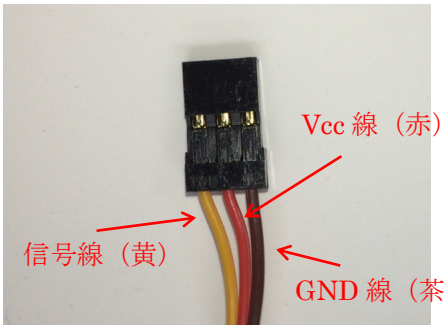

図 7.1.8：SG90 のピン配置

図 7.1.9 に SG90 の制御信号の概要を示す。SG90 は 50Hz(20ms)の PWM 信号で動作する。Arduino UNO の PWM 出力ピン 3, 9, 10, 11 番は 490Hz, ピン 5, 6 番は 980Hz の PWM を出力するので、これを直接使うことはできない。

本実験では、後述する Servo ライブラリを用いる。Servo ライブラリは 50Hz の PWM を出力できる。

サーボモータの回転軸を 0 度～180 度まで回転するためには、パルス幅を 1～2ms（duty 比 5%～10%）まで変化させる。パルス幅 1ms（duty 比 5%）がサーボモータの 0 度に、2ms（duty 比 10%）が 180 度に、おおよそ対応している。

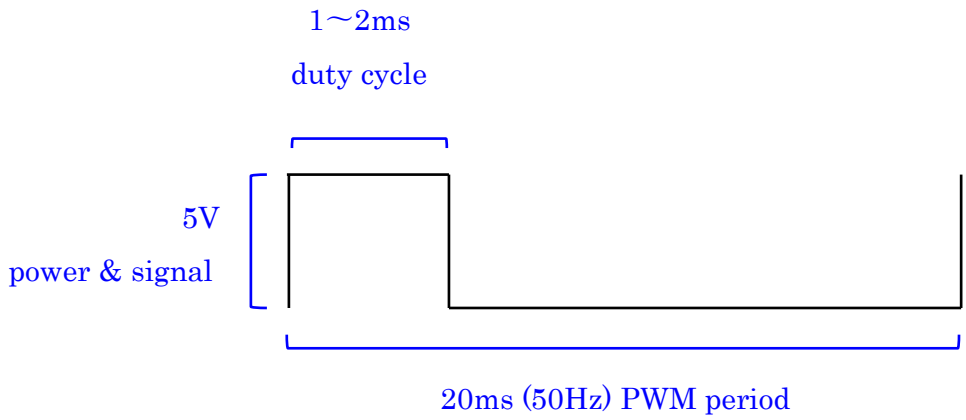

図 7.1.9：SG90 の制御信号

表 7.1.2 に、SG90 の仕様を示す。回転角度は 0～180 度である。トルクとは回転の強さであり、回転軸から 1cm の距離で最大 1.8kg の物体を動かすことができることを意味する。駆動速度とは、回転の速さであり、60 度回転するのに 0.1s かかる。

表 7.1.2：SG90 の仕様

項目	内容
回転角度	0 ～ 180 degree
トルク	1.8 kg/cm
駆動速度	0.1 s / 60 degree
定格電圧	5.0V
適正温度	0 ～ 55 ℃
寸法	23 mm × 12.1 mm × 29 mm
重さ	9 g

警告：

サーボモータの回転軸を無理に手で回してはいけない。内部の歯車が欠けて、故障の原因となる。

故障した場合は弁償の対象となる。

＜演習 7.1.2＞

サーボモータを動かしてみよう。Arduino 開発環境には、サーボモータを簡単に動作させるための Servo ライブラリが用意されている。

（1）サーボモータの回転軸にサーボホーンを図 7.1.10 のように 90 度の位置に取り付ける。

後で取り外して角度を調整するので、軽く取り付けておくとよい。

（2）サーボモータのコネクタに、図 7.1.11 のようにジャンパワイヤを接続する。

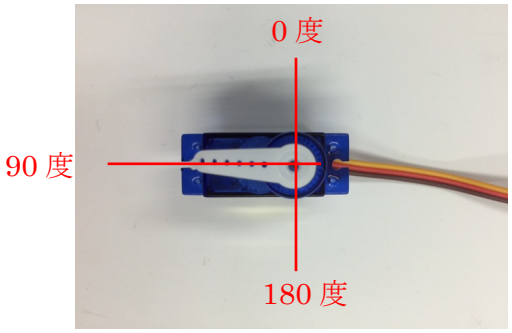

図 7.1.10：サーボホーンの取り付け

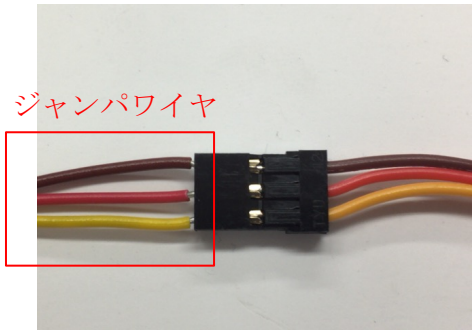

図 7.1.11：コネクタとジャンパワイヤの接続

（3）図 7.1.12 の回路をブレッドボードに組む（Arduino と PC はまだ接続しない）

- サーボモータの信号線（黄）を Arduino のデジタル 10 番ピンに接続（デジタルピンならどこでも可）

- サーボモータの電源線（赤）を Arduino の 5V ピンを接続
- サーボモータの GND 線（茶）を Arduino の GND ピンに接続

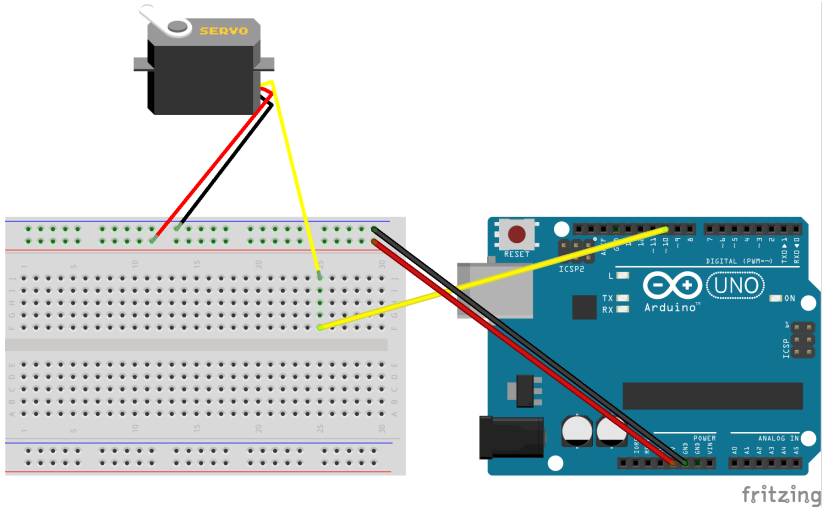

図 7.1.12：演習 7. 1. 2 のブレッドボード図

（4）次のプログラム 7. 1. 2 を Arduino IDE で作成する

プログラム 7. 1. 2

```
#include <Servo.h>    // サーボライブラリのインクルード

Servo servo;  //Servo クラスのインスタンスを生成

const int servoPin = 10; //サーボモータの信号線をデジタル 10 番ピンに接続

void setup() {
  servo.attach(servoPin);  //制御の対象を 10 番に割り当て
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    char val = (char)Serial.read();  //文字の読み込み
    switch (val) {
      case 'a':
        servo.write(0);   //0 度まで回転
        break;
      case 'b':
        servo.write(90);  //90 度まで回転
        break;
      case 'c':
        servo.write(180); //180 度まで回転
        break;
      default:
        servo.write(0);
    }
  }
}
```

（3）Arduino と PC を USB ケーブルで接続

（4）プログラムを「検証」→「マイコンボードへ書き込む」

（5）シリアルモニタを開き、送信欄に a, b, c のいずれかを入力

（6）サーボホーンが 0 度、90 度、180 度に回転することを確認せよ

（7）サーボホーンの向きを調節して、0 度、90 度、180 度になるようにせよ

Servo ライブラリはサーボモータの制御に用いられる。Arduino UNO で Servo ライブラリを使用した場合、analogWrite 関数によるデジタル 9 番と 10 番ピンの PWM 出力が無効になるので注意せよ。

以下に、プログラム 7.1.2 で使われている Servo ライブラリの関数を紹介する。

`attach(pin, min, max)`

サーボ制御の対象を `pin` に割り当てる。

【引数】

`pin`: サーボ制御の対象となるピンの番号

`min`(オプション): サーボの角度が `0` 度のときのパルス幅（μ s）。デフォルトは `544`。

`max`(オプション): サーボの角度が `180` 度のときのパルス幅（μ s）。デフォルトは `2400`。

`write(angle)`

回転角度をセットし、回転軸をその方向に向ける。

【引数】

`angle`: 回転させる角度（`0`〜`180`）

その他の関数については、Arduino のリファレンス（http://www.musashinodenpa.com/arduino/ref/）を参照せよ。

＜課題 7.1.1＞

超音波センサを用いて対象物までの距離を計測し、シリアルモニタに表示するプログラムを作成せよ。

① 気温を 25℃と仮定し、式 7.1.1 (p.206) および 式 7.1.2 (p.206) を用いて距離を算出せよ。距離の単位は cm とする。

② pulseIn 関数の第 3 引数を変更し、100cm 以上（>=100）の距離を計測しない（タイムアウトさせて 0 を返す）ようにせよ。

③ 対象物を超音波センサから前後させたとき、計測値が適切に増減するか確認せよ。

距離が適切に計測できない場合、以下の原因が考えられる。

- 布や綿など柔らかく凸凹した物は、超音波を乱反射し、適切な出力を得られないことがある。学生証など表面が硬く平面な対象物を使うとよい。
- 他の超音波センサが発射した超音波を、受信機が拾っていることがある。他の超音波センサと距離をあけたり、向きを変えてみるとよい。

＜課題 7.1.2＞

超音波センサによる距離計測を、タイマ割り込みで 100ms 毎に行うプログラムを作成せよ。

- 距離の単位は cm, 100cm まで計測、気温は 25℃と仮定する。
- タイマ割り込みの詳細は、テキスト「基礎実験 4　割り込み」を参照せよ。

＜課題 7.1.3＞

超音波センサの測定精度を確認してみよう。

距離の測定精度の確認

① 図 7.1.13 の位置に超音波センサと同心円図を設置せよ。

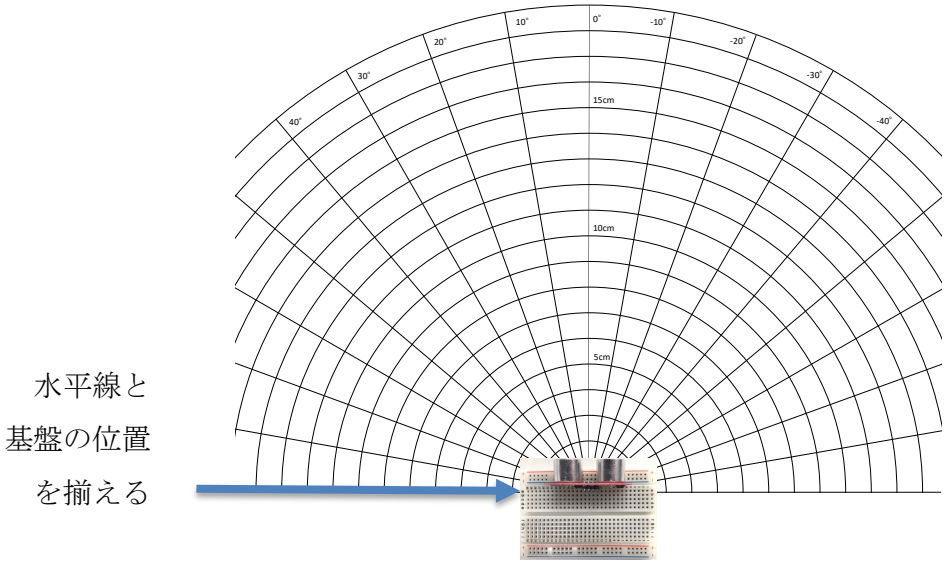

図 7.1.13：超音波センサの配置

② 対象物を 0° の方向（基盤正面）に置き、2cm～19cm の目盛りまで 1cm ずつ遠ざけながら、超音波センサで測定した距離を記録せよ。対象物は学生証など、表面が硬く平面な対象物を使うとよい。

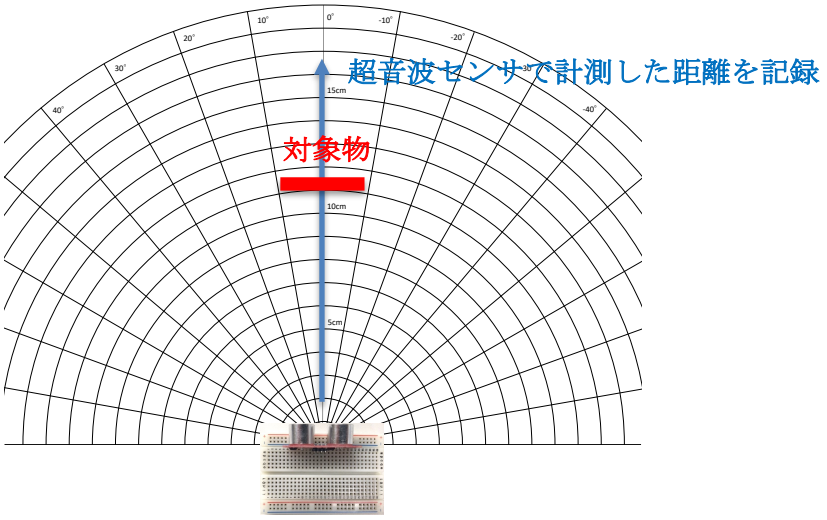

図 7.1.14：距離に対する測定精度の確認

③ 実際の距離を横軸、測定距離を縦軸とする折れ線グラフを作成せよ（LibreOffice Calc などを用いるとよい）。

④ 対象物を 0° の方向、15cm の位置に置き、少しずつ回転させて、正確に計測できなくなる角度を記録せよ。

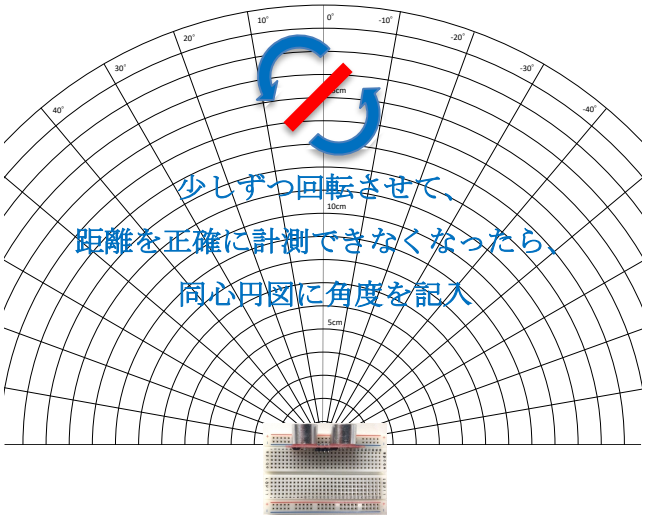

図 7.1.15：角度に対する測定精度の確認

指向性の確認

⑤ 超音波センサによる測定距離が 15cm を示す点を、−90° から 90° まで 10° 毎に、同心円図へプロットせよ。15cm を示す点が存在しなければプロットしなくてよい。

＜課題 7.1.4＞

衝突警報装置を作ってみよう。

- 超音波センサで距離を計測せよ。
- 圧電ブザーを用いて、測定距離が短くなるに従って高い音、長くなるに従って低い音を出せ。
  - 圧電ブザーの使い方はテキスト「基礎実験３　AD 変換」を参照せよ。
  - 測定距離が 0cm の時、最も高い音として 1000Hz の音を鳴らせ。
  - 測定距離が 30cm 以上（>=30)の時、最も低い音として 100Hz の音を鳴らせ。
  - 0cm～30cm の間の音の周波数は線形に変化させよ。
  - タイマ割り込みと tone 関数を同時に使うことはできない。ポーリングで処理せよ。
  - 抵抗は 330Ω を用いよ（抵抗を入れ忘れないこと！大電流により圧電スピーカが壊れる！）
- 測定距離が 10cm 以内（<=10）なら赤色 LED を点灯させ、10cm より上(>10)なら消灯させよ。
  - LED の使い方はテキスト「基礎実験１　電子回路の基礎」および「基礎実験２　Arduino 開発環境とディジタル IO」を参照せよ。
  - 抵抗は 330Ω を用いよ（抵抗を入れ忘れないこと！大電流により LED が壊れる！）

圧電ブザーを任意の音高（周波数）で鳴らすには、tone 関数を用いると便利である。tone 関数は、任意の Arduino のピンに対して、デューティー比 50%の矩形波を生成する。圧電ブザーに対して、tone 関数で生成した矩形波を入力すれば、任意の音高で鳴動させることができる。

tone 関数の書式を以下に示す。

```
tone(pin, frequency, duration)
```

指定した周波数の矩形波（デューティ比 50%）を生成する。duration を指定しなかった場合、noTone 関数を実行するまで動作を続ける。この関数はデジタル 3 番と 11 番の PWM 出力を無効にするので注意。

【引数】

pin: トーンを出力するピン

frequency: 周波数(Hz)

duration（オプション）: 出力する時間(ms)

```
noTone(pin)
```

tone 関数で開始された矩形波の生成を停止する。

【引数】

pin: トーンの生成を停止するピン

tone 関数の使用例を以下に示す。

```
const int pin = 9;  //デジタル 9 番ピンをトーン出力先として割り当て

void setup() {}
void loop() {
   tone(pin, 100);  //デジタル 9 番ピンに 100Hz の矩形波を出力し続ける
}
```

ある範囲の数値を別の範囲の数値に置き換える処理を行うには map 関数が便利である。map 関数の詳細については、テキスト「可視化実験２」を参照せよ。

map 関数の書式を以下に示す。

map(value, fromLow, fromHigh, toLow, toHigh)
数値 value をある範囲(fromLow～fromHigh)から別の範囲(toLow～toHigh)に変換する。map 関数は整数だけを扱う。変換の結果、小数が生じた場合は、小数点以下は切り捨てられる。
【引数】
value: 変換する値
fromLow: 現在の範囲の下限
fromHigh: 現在の範囲の上限
toLow: 変換後の範囲の下限
toHigh: 変換後の範囲の上限
【戻り値】
変換後の値(long)

Arduino の関数については、リファレンス（http://www.musashinodenpa.com/arduino/ref/）を参照せよ。

＜課題 7.1.5＞

サーボモータを「0 度から 180 度まで一定速度で回転させ、再び 0 度まで一定速度で戻る」を繰り返すようにプログラムせよ（扇風機の首振りと同様の動き）

- サーボモータの回転速度を半固定抵抗で変えられるようにせよ。
  - ✧ 半固定抵抗は 10kΩを用いよ。
  - ✧ 0 度～180 度まで回転し、再び 180 度から 0 度まで戻ってくるのに約 5 秒～約 18 秒の間に設定できるようにすること。厳密に 5 秒～18 秒である必要はない（ライブラリ関数の処理時間などが無視できないため）。
  - ✧ 抵抗値が 0Ωのとき 5 秒、10kΩのとき 18 秒となるようにせよ。

＜発展課題 7.1.6＞

通行人カウント装置を作ってみよう。

- 超音波センサの前を人が通過した（横切った）回数をカウントし、シリアルモニタに表示せよ。
- 同時に横切る人数は１名とする（同時に複数人が横切らない）
- 横切る方向（左／右／斜め）や速さ（素早く／ゆっくり／超音波センサの前で一旦立ち止まる）を変えても、できるだけ正確にカウントできるよう工夫せよ。

＜発展課題 7.1.7＞

対象物までの距離を超音波センサで計測し、距離を Processing で可視化して表示するプログラムを作成せよ。ただし、距離を数字（text 関数）で表示するのではなく、空間での位置関係を分かりやすく図示すること。

【レポート 7.1（2019 年 6 月 28 日出題、7 月 5 日 12:50 締切）】

プログラムにはコメントを記述すること。コメントがないものは減点する。

レポート 7.1.1

課題 7.1.1 で実装したブレッドボード図とプログラムを報告せよ。

pulseIn 関数の第 3 引数を大きくすることによる長所と短所を考察せよ。

レポート 7.1.2

課題 7.1.2 で実装したプログラムを報告せよ。

距離の計測において、ポーリングによる計測と、タイマ割り込みによる計測は、どのような状況で使い分けるべきか考察せよ。

レポート 7.1.3

課題 7.1.3 で作成した折れ線グラフと同心円図を報告せよ。

同心円図はスキャナで読み込んだり、カメラ撮影した画像を用いるなど、デジタル化してレポートに掲載すること。

正確に距離計測するには、どのような点に注意すればよいか、実験結果から考察せよ。

レポート 7.1.4

課題 7.1.4 で実装したブレッドボード図とプログラムを報告せよ。

レポート 7.1.5

課題 7.1.5 で実装したブレッドボード図とプログラムを報告せよ。

サーボモータを指定した速度で回転させるために考慮した点を解説せよ。

発展課題レポート 7.1.6

発展課題 7.1.6 で実装したブレッドボード図とプログラムを報告せよ。

適切にカウントするために工夫した点について解説せよ。

発展課題レポート 7.1.7

発展課題 7.1.7 で実装したブレッドボード図とプログラム（Arduino と Processing）を報告せよ。実行結果のスクリーンショットも示せ。

距離を分かりやすく可視化するために考慮した点について解説せよ。

使用部品一覧

部品	個数	備考
Arduino UNO	1	
USB ケーブル	1	
ブレッドボード	1	
ジャンパワイヤ	1	
赤色 LED	1	
抵抗 330Ω	2	
半固定抵抗 10kΩ	1	
圧電ブザー	1	
同心円図	1	
超音波センサ HC-SR04	1	貸出（7/12 回収）
サーボモータ SG90	1	貸出（7/12 回収）

貸出部品は次回も使用するので忘れずに持ってくること。

後日回収するので、破損や紛失のないように注意して取り扱うこと。

参考図書

[1] 日本音響学会：音のなんでも小事典，講談社（2005).

[2] Massimo Banzi，船田巧：Arduino をはじめよう，オライリージャパン（2014).

[3] 都筑卓司：なっとくする音・光・電波，講談社（2004).

[4] システム設計および実験運営委員会:システム設計および実験テキスト,徳島大学工学部知能情報工学科(2014).