

# マイコン基礎\*

カトリ スザン

18NC021

# Contents

1	実験日の天候状態	2
2	実験の目的	2
3	使用機器	2
4	実験結果	2
4.1	実験 1 Arduino の動作確認	2
4.1.1	実験手順	2
4.1.2	実験結果	2
4.2	実験 2 デジタル出力による LED 点灯回路	3
4.2.1	実験手順	3
4.2.2	実験結果	3
4.3	実験 3 スイッチによるデジタル入力で LED 点滅	4
4.3.1	実施手順	4
4.4	実験 4 アナログ入力による LED 点灯制御	4
4.4.1	実施手順	4
4.4.2	実験結果	5
4.5	実験 5 アナログ出力で、暗いと LED を明るく点灯する	6
4.5.1	実施手順	6
4.6	実験 6 圧電スピーカーでのドレミ演奏	8
4.6.1	実施手順	8
4.6.2	実験結果	9
4.7	実験 7 フルカラー LED のデジタル駆動	9
4.7.1	実施手順	9
4.7.2	実験結果	10
4.8	実験 8 フルカラー LED のアナログ駆動	10
4.8.1	実施手順	10
4.8.2	実験結果	11
5	検討事項	11
5.1	デフォルトで保存されるファイル名の名づけについて述べよ。	11
5.2	digitalWrite、digitalRead、analogRead、analogWrite の各関数の機能	11
5.3	LED 点灯時の電流制限抵抗の電圧降下などから出力インピーダンス $Z$ を推定せよ。	12
5.4	プルダウン抵抗をプルアップ抵抗に変更した場合のプログラム変更について	12
5.5	analogRead 関数の値が 0~1023 の 1024 段階で、analogWrite 関数の値が 0~255 の 256 段階となっている理由	12
5.6	map 関数の機能について述べよ。	12
5.7	PWM の波形と音階の波形からその機能の違いについて具体的に述べよ。	13
5.8	PWM と tone 関数の関係について述べよ。	13
6	吟味	13
6.1	実験 1, 2	13
6.2	実験 3	13
6.3	実験 4	13
6.4	実験 5	13
6.5	実験 6	13
6.6	実験 7, 8	13
7	参考文献	14

## 1 実験日の天候状態

- 実験日: 2019/12/ (2 ,9) 日
- 天候 (2 日): 晴れ、24°C、40%
- 天候 (9 日): 晴れ、22°C、26%

## 2 実験の目的

ワンボードマイコンである Arduino を用いてマイコンの入力とプログラミングについて学習する。

## 3 使用機器

本実験で使用した機器を下記の表に示す。

表 1: 使用機器

機器名	型番等
マイコン	Atmel 社 Arduino
パソコン	Windows 10
オシロスコープ	IWASU DS-5104B
プローブ	SS-0112

## 4 実験結果

### 4.1 実験 1 Arduino の動作確認

#### 4.1.1 実験手順

Arduino 本体とパソコンを USB ケーブルで接続する。Arduino.exe を起動し、ボードの種類を” Arduino/Genuino Uno”, シリアルポートを COM9(Arduino/Genuino Uno) と選択する。開発環境 (Arduino.exe) で以下のプログラムを作成し、コンパイル後、Arduino 本体に転送する。

```
void setup(){
    pinMode(13, OUTPUT);
}
void loop(){
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
}
```

転送後、LED(13 番ピンランプ) が点滅することを確認し、スケッチをデスクトップ下の docs ファイルに保存する。

#### 4.1.2 実験結果

Arduino の基板上の 13 番ピン LED が点滅する様子を確認した。

## 4.2 実験 2 デジタル出力による LED 点灯回路

### 4.2.1 実験手順

ブレットボード上に LED 点灯回路を作成し、実験1のプログラムで動作させる。

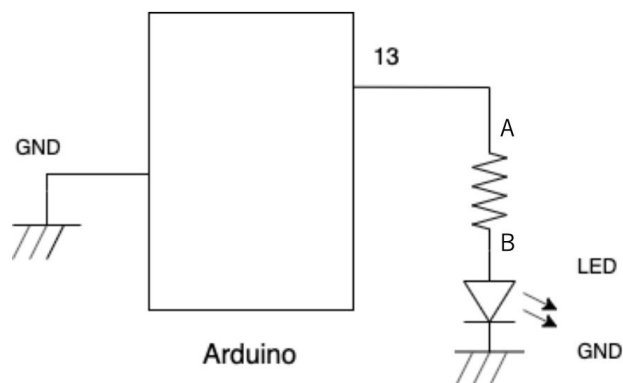


図 1: 実験 2 回路図

### 4.2.2 実験結果

今回は青の LED を使用したので、データシートより順方向電圧  $V_F = 3.1[\text{V}]$  抵抗値の計算の過程を記す。最大電流 ( $16\text{mA}$ ) の 25% である順方向電流  $I_F = 4[\text{mA}]$  にと想定した。

$$\begin{aligned} R &= \frac{E - V_F}{I_F} \\ &= \frac{5.0 - 3.1}{4} \\ &= 475[\Omega] \end{aligned}$$

E24 系列の抵抗から計算値に近い  $470\Omega$  の抵抗を使用する。  
 $470\Omega$  の抵抗を利用した場合の電流  $I$  は

$$\begin{aligned} I &= \frac{E - V_F}{470} \\ I &= \frac{5 - 3.1}{470} \\ I &= 4.0[\text{mA}] \end{aligned}$$

求めた抵抗値を用いて回路を作成して、A 点と B 点の点灯時と消灯時の電圧を測定したものを表1に示す。

表 2: 点灯時と消灯時の電圧

	消灯時の電圧 [V]	点灯時の電圧 [V]
A - GND 間	0.0013	4.71
B - GND 間	0.0012	2.68

上記の表の結果を用いて電流値を求める

$$I = \frac{V_A - V_B}{R}$$

$$I = \frac{4.71 - 2.68}{470}$$

$$I \approx 4.31[mA]$$

電流は計算した結果はより 0.31mA 大きくなっており、7.8% の誤差が生じている。この誤差は実際に計算値とは違う抵抗 (E24 系列に近い抵抗) を使ったためと Arduino の内部抵抗が理由であると考えられる。

### 4.3 実験 3 スイッチによるデジタル入力で LED 点滅

#### 4.3.1 実施手順

実験 2 で作成したブレットボード上の LED 点灯回路に、スケッチ入力回路を追加し、スイッチを押している間は LED を点灯させるプログラムで動作させた。作成した回路は図 2 に、プログラムは以下に示す。

```
int swPin = 10;
int ledPin = 13;
void setup() {
  pinMode(swPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (digitalRead(swPin) == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  delay(10);
}
```

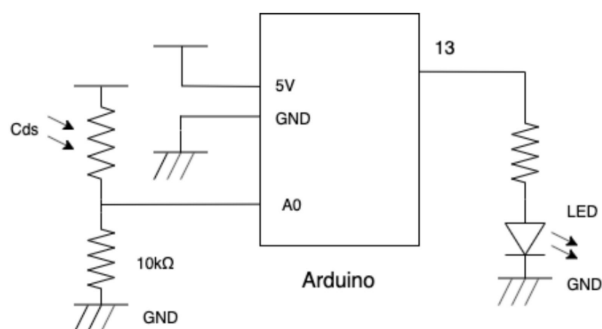


図 2: 実験 3 回路図

### 4.4 実験 4 アナログ入力による LED 点灯制御

#### 4.4.1 実施手順

実験 2 で作成したブレットボード上の LED 点灯回路に、Cds 光電セルからのアナログ入力する回路を追加し、明るいとき消灯し、暗いとき点灯させるプログラムを作成した。作成した回路図を図 3 に、プログラムを以下に記す。

```

int ledPin = 13;
int sensorPin = 0;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  int bright = analogRead(sensorPin);
  Serial.println(bright);
  if (bright < 440) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  delay(100);
}

```

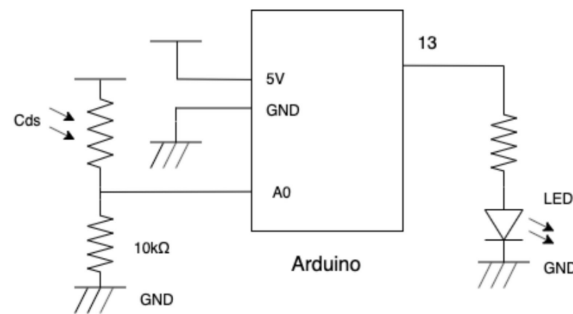


図 3: 実験 4 回路図

#### 4.4.2 実験結果

本実験で使った Arduino の AD 変換器は 10bit で量子化するため、0V から 5V の電圧を 0 から 1023 の数値に変換 (map) する。入力電圧と変換値の対応は以下の通りである。

表 3: 入力電圧と変換値の対応表

入力電圧	0V	1V	2V	3V	4V	5V
変換値	0	204	409	613	818	1023

本実験用いた Cds セルは明るい状態で  $1k\Omega$  程度の抵抗値である。Cds と抵抗にかかる電圧とそれらの抵抗値との間には、比例の関係「Cds と  $10k\Omega$  の合計抵抗値:  $10k\Omega =$  電源電圧 5V : 測定電圧  $V_{A0}$ 」がある。よって、明るい状態での測定電圧は

$$1 + 10 : 10 = 5 : V_{A0}$$

$$V_{A0} \approx 4.5[V]$$

明るい状態の Cds セルの変換値を、シリアルモニタより読み取ると 840 であった。この時の入力電圧は計算すると 4.12[V] である。Cds と抵抗にかかる電圧とそれらの抵抗値の関係より、Cds セルの抵抗値  $R_{Cds}$  は

$$R_{Cds} + 10 : 10 = 5 : 4.12$$

$$4.12R_{Cds} = 50 - 41.2$$

$$R_{Cds} \approx 2.14[k\Omega]$$

Cds セルは暗くなると抵抗値が増大する。増大すると  $A_0$  に入力される電圧が減少するので、変換値も減少する。暗いと点灯に適切な設定値は 520 であった。

## 4.5 実験 5 アナログ出力で、暗いと LED を明るく点灯する

### 4.5.1 実施手順

実験 4 で作成したブレットボード上の LED 点灯回路と、Cds セルからのアナログ入力回路は変更せずに、暗くなるにつれて LED の明るさを徐々に明るくするプログラムを作成する。LED を 13 番ピンから 11 番ピンに変更し、オシロスコープを用いて 11 番ピンの波形観測を行う。作成した回路図を図 5、波形観測したスケッチを図 4、プログラムを以下に記す。

```
int ledPin = 11;
int sensorPin = 0;
void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}
void loop() {
    int bright = analogRead(sensorPin);
    int value = map(bright, 520, 840, 255, 0);
    Serial.print(bright);
    Serial.print(" ");
    Serial.println(value);
    if (value < 0) {
        value = 0;
    }
    analogWrite(ledPin, value);
    delay(100);
}
```

# 別ページ

図 4: 実験 5 すけっち



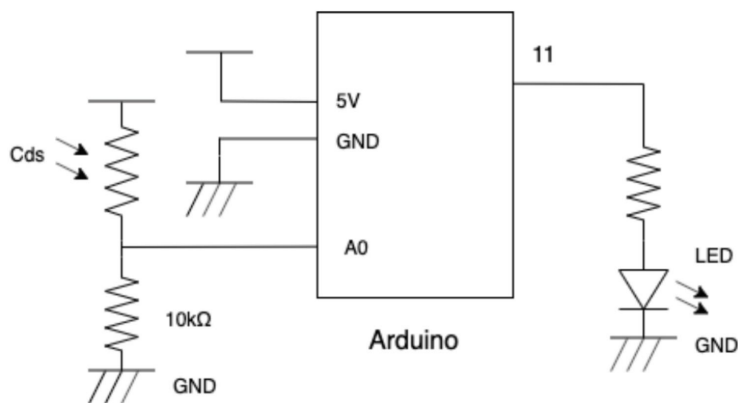


図 5: 実験 5 回路図

## 4.6 実験 6 圧電スピーカーでのドレミ演奏

### 4.6.1 実施手順

電圧で変形する圧電素子を用いた圧電スピーカーを用いて、tone 関数で音階を鳴らす。オシロスコープにより 12 番ピンの波形観測を行う。作成した回路図を図 6、実行したプログラムを以下に記す。

```
int f[] = {262, 294, 330, 349, 392, 440, 495, 523};
void setup() {
}
void loop() {
  int i;
  for(i = 0; i < 8; i++){
    tone(12, f[i], 200);
    delay(200);
  }
  delay(300);
}
```

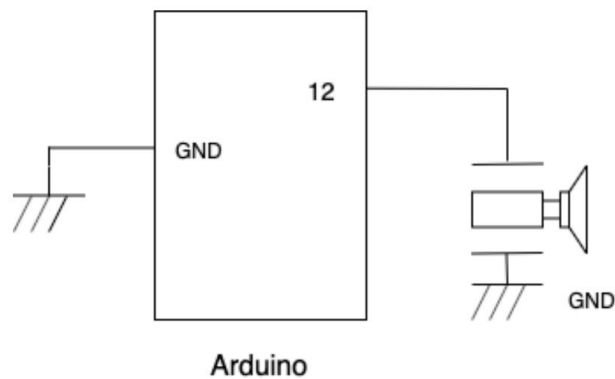


図 6: 実験 6 回路図

### 4.6.2 実験結果

波形観測より、音階が上がるにつれて、周波数が上がる。また、デューティーは常に 50% で音階が変わっても変化がない。

## 4.7 実験 7 フルカラー LED のデジタル駆動

### 4.7.1 実施手順

順方向電流は各色 20mA であるが、順方向電圧は、Red が 2.0V、Blue と Green は 3.6V である。20mA の 2 割となるように抵抗を選択する。作成した回路図を図 7、実行したプログラムを以下に記す。

```
int rPin = 11;
int bPin = 10;
int gPin = 9;
void ledD(int pin);

void setup() {
  pinMode(rPin, OUTPUT);
  pinMode(gPin, OUTPUT);
  pinMode(bPin, OUTPUT);
}
void loop() {
  ledD(rPin);
  ledD(gPin);
  ledD(bPin);
}
void ledD(int pin){
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);
}
```

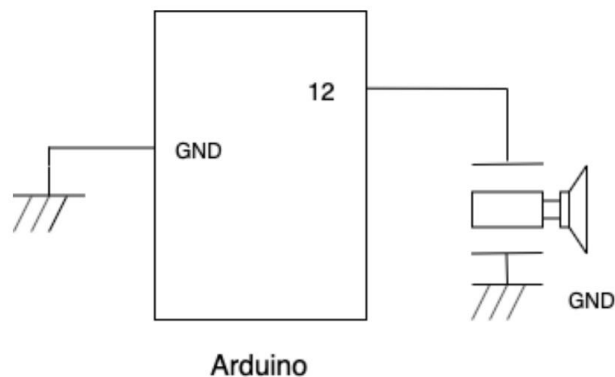


図 7: 実験 7 回路図

#### 4.7.2 実験結果

Red の抵抗値  $R_1$  の計算過程を記す。

$$R_1 = \frac{(5 - 2) * 10^3}{20 * 0.2}$$

$$R_1 = 750[\Omega]$$

計算結果に最も近い E24 系列の  $820\Omega$  の抵抗を選択した。選択した抵抗での電流  $I_1$  は

$$I_1 = \frac{5 - 2}{750}$$

$$I_1 = 4.0[\text{mA}]$$

Green, Blue の抵抗値  $R_2, R_3$  の計算過程を記す。

$$R_2 = R_3 = \frac{(5 - 3.6) * 10^3}{20 * 0.2}$$

$$R_2 = R_3 = 350[\Omega]$$

計算結果に最も近い E24 系列の  $330\Omega$  の抵抗を選択した。選択した抵抗での電流  $I_2, I_3$  は

$$I_2 = I_3 = \frac{(5 - 3.6)}{330}$$

$$I_2 = I_3 = 4.0[\text{mA}]$$

よって、各色ほぼ同じ電流値になった。

### 4.8 実験 8 フルカラー LED のアナログ駆動

#### 4.8.1 実施手順

ハードウェアは実験 7 と同一で、プログラムだけ異なる。使用した回路図を図 7 である。実行したプログラムを以下に記す。

```

int rPin = 11;
int bPin = 10;
int gPin = 9;
int r, g, b;
void setup() {
    pinMode(rPin, OUTPUT);
    pinMode(gPin, OUTPUT);
    pinMode(bPin, OUTPUT);
    r = 0;
    g = 0;
    b = 0;
}
void loop() {
    for(r = 0, b = 255; r < 255; b--, r++){
        ledA();
    }
    for(g = 0, r = 255; g < 255; r--, g++){
        ledA();
    }
    for(b = 0, g = 255; b < 255; g--, b++){
        ledA();
    }
}
void ledA(){
    analogWrite(rPin, r);
    analogWrite(gPin, g);
    analogWrite(bPin, b);
    delay(10);
}

```

#### 4.8.2 実験結果

LED は「青、紫、赤、黄、緑、水色」と変化した。プログラムでは「青、赤、緑」と変化するが、徐々に変化するため実際には色が混ざりあい紫、黄色、水色といった色が確認できる。

## 5 検討事項

### 5.1 デフォルトで保存されるファイル名の名づけについて述べよ。

デフォルトでは'sketch\_dec02a' という名前で保存される。最初の'sketch' の部分は Arduino Sketch プログラムのファイル (スケッチ) であることを示す。その次にある部分'de02' はそのファイルが作成された月と日である。'dec02' は December, 02 (12 月 2 日) に作成されたファイルということになる。そして末尾のアルファベットはファイルが作られた順番である。

['sketch': 固定]\_(作成された月の英語名の先頭 3 文字: ex. dec)(作成された日: ex. 02)(日付ごとにアルファベットで作成順番: ex. a)]

### 5.2 digitalWrite、digitalRead、analogRead、analogWrite の各関数の機能

- digitalWrite : デジタルピンを'HIGH' または'LOW' にセットするための関数。

- digitalRead : 指定したデジタルピンが'HIGH' か'LOW' かを読み取る関数。戻り値は'HIGH' か'LOW'。
- analogRead : 指定したアナログピンの値 (0~5V) を読み取り、0~1023 の値を返す。
- analogWrite : 指定したアナログピンから指定したアナログ値を返す。

### 5.3 LED 点灯時の電流制限抵抗の電圧降下などから出力インピーダンス $Z$ を推定せよ。

実験 2 の結果を用いる。A 点で電圧が 4.81V だったため、

$$\frac{5}{470 + Z} \times 470 = 4.81$$

$$Z = \frac{5 \times 470}{4.81} - 470$$

$$Z \approx 18.57$$

### 5.4 プルダウン抵抗をプルアップ抵抗に変更した場合のプログラム変更について

スイッチが押されているとき LOW を出力し、放している時 HIGH を出力するよう変更する。プログラムは下記の通りになる。

```
int swPin = 10;
int ledPin = 13;
void setup() {
    pinMode(swPin, INPUT);
    pinMode(ledPin, OUTPUT);
}
void loop() {
    if (digitalRead(swPin) == LOW) {
        digitalWrite(ledPin, HIGH);
    } else {
        digitalWrite(ledPin, LOW);
    }
    delay(10);
}
```

### 5.5 analogRead 関数の値が 0~1023 の 1024 段階で、analogWrite 関数の値が 0~255 の 256 段階となっている理由

Arduino の AD 変換器の量子化ビットは 10 ビットであるため、analogRead 関数の値が 0~1023 の 1024 段階になっている。一方 analogWrite の場合は DA 変換が 8bit で行なわれているため、analogWrite 関数の値が 0~255 の数値となる。

### 5.6 map 関数の機能について述べよ。

Arduino の map 関数は数値をある範囲から別の範囲へ変換する。map(a, 0, 1023, 255, 0) ならば、a の値の範囲を 0~1023 から 255~0 に変換させている。

## 5.7 PWM の波形と音階の波形からその機能の違いについて具体的に述べよ。

PWM はデューティー比を変化させることで、Arduino の電圧を変化させている。音階の波形はデューティー比が 50% のまま変わらず、周波数が変化している。

## 5.8 PWM と tone 関数の関係について述べよ。

tone 関数は指定したピンに矩形波を出力するための関数である。また、出力信号の周波数そして長さも指定することができる。tone 関数が出力する PWM 信号のデューティー比は 50% で固定されている。tone 関数使用時には、3 番と 11 番 pin の PWM 出力ができなくなる。AVR にタイマーが 3 つあり tone 関数はこれを使用するためである。そして、tone 関数では 31Hz 以下の周波数の信号は出力できない。

# 6 吟味

## 6.1 実験 1, 2

実験1,2では Arduino をパソコンに接続する方法、そして C 言語で書かれたプログラムを基に Arduino 用のバイナリを生成するためのツールである Arduino Sketch の初期設定やプログラムを本体に書き込みする方法など、Arduino を動かすために必要最低限のことについて理解できた。また、過大電流が流れ、LED が破壊されないよう最大電流の 2~3 割が流れるような制限用の抵抗を計算する方法についても理解できた。

## 6.2 実験 3

実験3ではボタンを押すと LED が光る回路を構成した。そしてボタンが接続されたピンが HIGH になると LED が光る(LED が接続されてる pin も HIGH になる)プログラムを書いた。本実験を通して Arduino のプログラムの構成(loop, setup 関数)について詳しく理解できた。

## 6.3 実験 4

実験 4 では、Cds セルからのアナログ入力による LED 点灯制御を行った。本実験では analogRead 関数を使ってアナログピンの値を 0~1023 の離散的な値に変換して取得できることが分かった。また、シリアルモニターに任意の値や文字列を出力できる Serial.println 関数について分かった。

## 6.4 実験 5

実験 5 では Cds セルを使い明るさに合わせて、led の輝度をコントロールした。PWM のデューティー比を変えて led の明るさを変えられることを確認した。そして、デジタル値をアナログ信号に変換して出力できる analogWrite 関数について理解した。

## 6.5 実験 6

実験 6 では圧電スピーカーを使い音の波形観察をした。デューティー比は 50% のまま固定さて、周波数が変化することを確認できた。

## 6.6 実験 7,8

実験 7,8 ではフルカラー led を点灯させた。デジタル駆動とアナログ駆動で動作させた。アナログの場合は led の色が途切れることなく連続的に変化し続けた。一方デジタル駆動では led の点灯は各色同時に点灯することができず、変化が途切れていた。

## 7 参考文献

- <https://iot.keicode.com/arduino/arduino-analogwrite-pwm.php>
- <https://www.arduino.cc/reference/en/#functions>