

3 ブレッドボードによる電気回路の作り方

Arduinoは電気回路（以下、回路）や部品に接続して制御する。回路は基板に部品を直接はんだ付けして製作するよりも、ブレッドボード（図2）上に組む方が容易である。ブレッドボードには無数の穴が規則正しく空いており、裏面で穴同士が電氣的に接続された構造である（指導資料情報教育第149号にも記載）。そのため、スイッチやセンサなどの入力装置やLED、アクチュエータ*3などの出力装置、部品同士をつなぐジャンパー線をボード上に配置し結線することで回路を組むことができる。

これらのことを踏まえ、図2はLEDを点灯する回路を組んだものである（回路図は図3）。LEDには極性*4があり、電池の+側に長い線の方を接続し（図4）、直列に電流制限用の抵抗（極性なし）を入れる必要がある。このように容易に回路を組めるので、電気の単元で扱うことでも教育効果が期待できる。

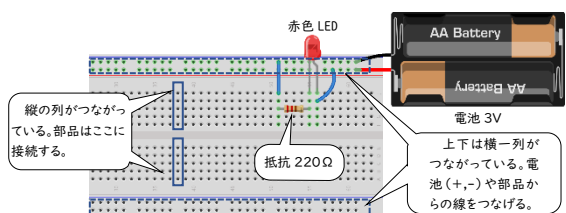


図2 ブレッドボードを用いたLED点灯回路 fritzing

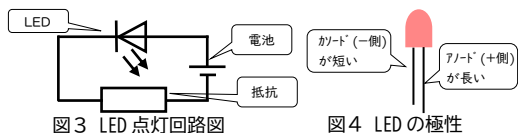


図3 LED点灯回路図

図4 LEDの極性

4 Arduinoにおける制御の流れ

赤色のLEDをArduinoにより点滅させる例（図5）で制御の流れを説明する。

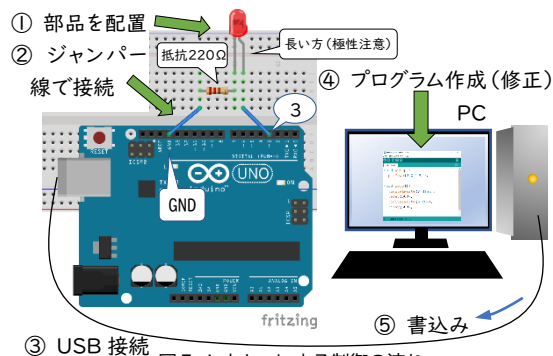


図5 Arduinoによる制御の流れ

- ① Arduinoに接続する前にブレッドボード上に、部品を配置し取り付ける。
- ② 次にジャンパー線で部品同士や電源と接続する。Arduinoへは図5のように3番ピンとGND*5に接続する。
- ③ 回路を完成させ電気を流すため、PCとArduinoをUSBケーブルで接続する。
- ④ プログラミングに移行するため、PCのアプリケーション「Arduino IDE」を起動する。インストールしていない場合は公式サイト*6よりダウンロードする。起動したエディタ*7上で図6左のスケッチ*8のとおり入力する。

<pre> void setup() { pinMode(3, OUTPUT); } void loop() { digitalWrite(3, HIGH); delay(500); digitalWrite(3, LOW); delay(500); } </pre>	<ul style="list-style-type: none"> setup() 関数*9で{}内を最初の1回だけ実行する。 pinMode(3, OUTPUT); 3番ピンを出力として設定する。 loop()関数で{}内を繰り返し実行する。 digitalWrite(3, HIGH); 3番ピンをHIGHにする(点灯)。 delay(500); 何もせず500ミリ秒待つ。 digitalWrite(3, LOW); 3番ピンをLOWにする(消灯)。 delay(500); 何もせず500ミリ秒待つ。
---	--

図6 LED点滅回路のスケッチ（左）と説明（右）

- ※ 入力に際して次のことに注意する。
- ・ 半角英数入力モードで入力する。
 - ・ 大文字・小文字を区別して入力する。
 - ・ 行末にセミコロン（;）を入力する。
 - ・ （）や {} の括弧は必ず対で用いる。

- ⑤ メニューバー「ツール」の「ボード」という項目で「Arduino UNO」, 「シリアルポート」で「COM※(Arduino UNO)*10」を選択する（この操作は、最初の1回だけ行う通信の設定である）。さらにメニューバーにある ボタンを押して、マイコンボードに書き込む。0.5秒間隔でLEDが点滅を繰り返せば成功である。

もしスケッチにエラーがある場合は、正しく動作するまで④と⑤を繰り返し、修正する。

以上が制御の流れだが、これを基に異なる色のダイオードを用いて車両用・歩行者用信号機を製作させたり、点滅の間隔をどのくらいまで縮めるとずっと点灯して見えるか検証させたりすることで、課題を解決する力や観察する力を養うことができる。

*3) モータなどの駆動装置 *4) 一方方向にしか電流が流れない性質より接続する端子の向きが決まっていること *5) 「グランド」と呼び、一般的に電源のマイナス側 *6) <https://www.arduino.cc/en/software> *7) プログラミングの編集作業を行うためのアプリケーションのこと *8) Arduino IDEではプログラムのことをスケッチと呼んでいる。 *9) ある処理をするための命令をまとめたものを関数という。 *10) ※には任意の数字が入る。

5 スイッチで入力を検知する方法

前項ではLEDが自動的に点滅を繰り返したが、タクトスイッチを用いることで人間が意図したタイミングで動作させることができる。回路作成では、図7の外観の⑦、⑧と回路図の⑦、⑧が対応することに注意し配置する(図8)。

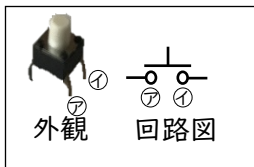


図7 タクトスイッチ

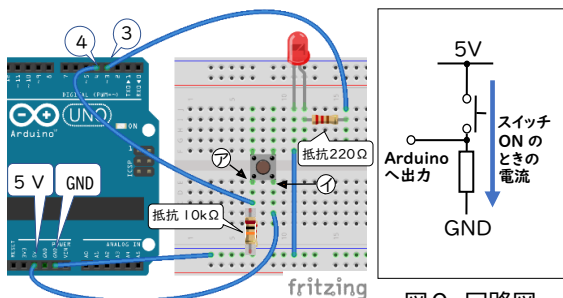


図8 スイッチを用いた回路

スイッチの取付には、図9のように抵抗を取り付ける必要がある。スイッチONのとき電流が流れ、Arduinoへの出力はHIGHとなり点灯し、OFFのとき電流が流れないのでGNDから抵抗を介して出力はLOWとなり消灯する。このような仕組みでArduinoの端子へHIGHまたはLOWを入力する。その際のスケッチと説明は図10のとおりである。さらに、複数のスイッチを使用することで、可能になるようなアイデアを生徒に考察させることもできる。

```

int b=0;

void setup(){
  pinMode(3,OUTPUT);
  pinMode(4,INPUT);
}

void loop(){
  b=digitalRead(4);
  if(b==HIGH){
    digitalWrite(3,HIGH);
  }else{
    digitalWrite(3,LOW);
  }
}

```

bは変数^{*11}と呼ばれ、値を代入することができる。intはその変数に整数を格納するという型宣言である。

4番ピンを入力して設定する。

4番ピンで読み込んだ値(スイッチON→HIGHまたはスイッチOFF→LOW)をbに代入する。

if文は条件分岐の構文であり、()内は条件式である。b==HIGHは、bの値がHIGHに等しい場合を表し、{}内は条件式が成立する場合の処理(点灯)である。

elseの後の{}内は条件式b==Hが成立しない場合の処理(消灯)である。

図10 スイッチ ON, OFF の回路のスケッチ(左)と説明(右)

*11) 変数の名前は一定の規則に従えば自由に決められる。*12) ArduinoからPC側へデータが送られ、ディスプレイ画面上にデータ表示させる機能 *13) 1秒当りに転送できるビット数の単位 *14) μ(マイクロ)は、10⁻⁶を表す接頭語

6 超音波センサで距離を測定する方法

超音波が物体に反射して返ってくる時間を測定するセンサHC-SR04(図11)を用いて、人や動物、物体が近付いたり、遠ざかったりするのを検知でき、2cm~4mまでの距離を測定する回路(図12)を製作する。

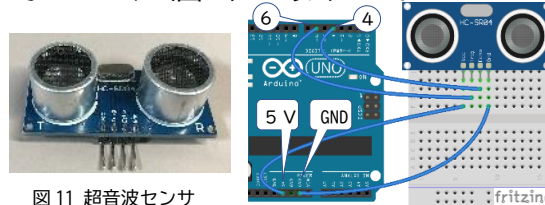


図11 超音波センサ

図12 超音波センサ接続図

スケッチと説明は図13のとおりである。出力は、Arduino IDEの右上のボタン **シリアルモニタ** を押し、さらに右下の通信速度 **19200 bps** を選択すると図14のように物体との距離が表示される。また、距離を広げたり縮めたりすることで値が変わることが確認できる。そのため、車が壁に近付いたら自動的に停止するなど様々な場面での利用を生徒に考察させることもできる。

```

void setup() {
  Serial.begin(19200);
  pinMode(6, OUTPUT);
  pinMode(4, INPUT);
}

void loop() {
  int d;
  float t;
  digitalWrite(6, HIGH);
  delayMicroseconds(10);
  digitalWrite(6, LOW);
  t=pulseIn(4, HIGH);
  d=*340*100/1000000/2;
  Serial.print(d);
  Serial.print("cm");
  delay(500);
}

```

シリアルモニタ^{*12}に19200bps^{*13}の速さの出力をするという設定である。

何もせず10マイクロ^{*14}秒待つ。

6番ピンがHIGHの信号になったとき、超音波が発せられ、物体から反射された超音波を4番ピンのセンサで受信し、その時間をt(マイクロ秒)に代入する。

物体までの距離を表す変数dにt*340*100/1000000/2を代入する。
→往復時間t(マイクロ秒)×音速(cm/s)÷10⁶÷2

cm単位でモニタ出力する。

図13 距離を測定する回路のスケッチ(左)と説明(右)

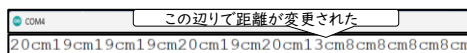


図14 シリアルモニタの出力例(距離)

センサは他にも、傾きや明るさ、気温、湿度、圧力、方位、衝撃、色、水位、動き、押された位置など様々な情報を検知し、データの取得が可能で、様々な出力と組み合わせて使用することにより更に応用範囲が広がる。

7 動きをつくるサーボモータの制御

目的とする角度に回転できるサーボモータを用いると、二足歩行ロボットの関節など様々な制御を行うことができるが、定格電流*15に気を付ける必要がある。SG90を図15のように接続し、図16を参考にして実行する。

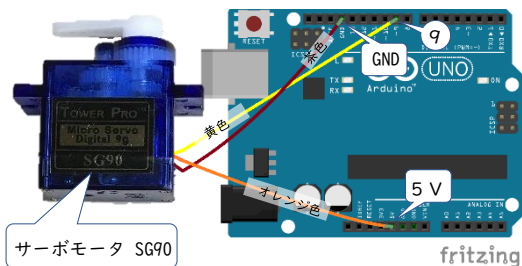


図15 サーボモータ駆動回路

<pre>#include<Servo.h> Servo sv; void setup() {sv.attach(9);} void loop() { sv.write(0); delay(1000); sv.write(45); delay(1000); sv.write(90); delay(1000); sv.write(180);delay(2000); }</pre>	<p>サーボモータのライブラリを追加する。使用時は最初に記載する。</p> <p>サーボモータのクラスを定義する。これも使用時は記載する。</p> <p>サーボモータの信号線が9番ピン*16であることを表す。</p> <p>サーボモータを基準から45°回転し、1000ミリ秒(=1秒)待つ。</p>
--	---

図16 サーボモータ駆動のスケッチ(左)と説明(右)

サーボモータの動きは図17のように繰り返す動作となる。スイッチやセンサによって、予め決められた動きをする場面での利用が考えられる。生徒に用途等のアイデアを自由にさせることで、課題を見付け解決する力を育むことができる。

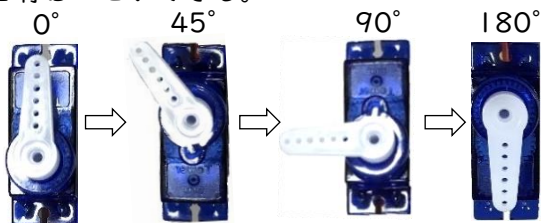


図17 サーボモータの動き

8 スピーカを鳴らす方法

音階は周波数の高さによって決まっている。例えば、圧電スピーカに断続的に電圧を加え、内蔵する振動板を1秒間に262回振動させれば、空気の振動により、262Hzのドの音として

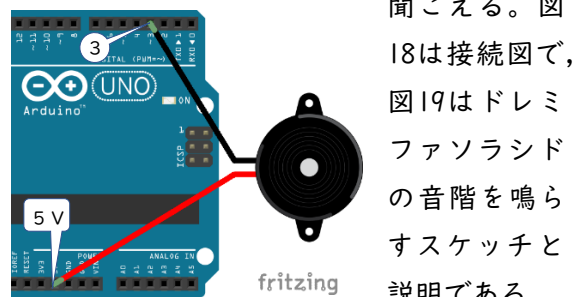


図18 圧電スピーカを鳴らす回路

聞こえる。図18は接続図で、図19はドレミファソラシドの音階を鳴らすスケッチと説明である。

<pre>void setup(){} void loop(){ tone(3,262); delay(500); tone(3,294); delay(500); tone(3,330); delay(500); tone(3,349); delay(500); tone(3,392); delay(500); tone(3,440); delay(500); tone(3,494); delay(500); tone(3,523); delay(500); noTone(3); delay(500); }</pre>	<p>tone(3,262)は3番ピンに262Hzの周波数の出力を500ミリ秒間行う関数である。つまり、ドの音を0.5秒間スピーカから鳴らすということ。以下、同様にレからドまでの音階を0.5秒間ずつ順番に鳴らす。</p> <p>noTone(3)は3番ピンの音を消すことを表す。</p>
---	---

図19 音階を圧電スピーカで鳴らすスケッチ(左)と説明(右)

これを応用して、人の検知によりメロディを流すような場面など、生徒に自由にアイデアを出させると創造力を育むことができる。

9 おわりに

Arduinoを授業等で活用し、試行錯誤しながら自然と思考力が身に付いていくところに、フィジカルプログラミングのよさがある。

さらに、今まで見過ごしてしまっていた身の回りのモノに関心を持ち、創意工夫しながら生徒の様々なアイデアを形にすることが可能になるだろう。

—参考文献—

- 鹿児島県総合教育センター「指導資料情報教育第149号」平成31年4月
- 平原真『実践Arduino!』2017年、オーム社
- 福田和宏『これ1冊でできる! Arduinoではじめる電子工作超入門』2020年、ソーテック社
(情報教育研修課 肥後 諭)
- ※ Fritzingは、CCライセンスに基づく表記です。
- ※ 本資料は、UDフォントを使用しています。

*15 流せる電流の上限のこと。Arduinoの定格電流は1Aまでと決まっており、この定格を上回る電子回路や機器を動かすときには、別電源から電気を送る必要がある。通常、大きな電流を要する電子機器類などは、トランジスタやFET、制御用ICと呼ばれる物を利用して接続する必要がある。 *16) 使用できるピンはPWM出力に対応したピンである必要があり、Arduino基板上の数字の横に波線(～)が記載されている。