

2.3 AD 変換

ロボットの各部の温度や光，加速度などの状態を知るには，それらの状態を検出して電気信号に変換するセンサが必要である。センサの出力がアナログ電圧の場合には，前回実験を行ったマイコンのデジタル入力ポートには直接入力できない。そこで，アナログ電圧をデジタル値に変換するためのAD変換（Analog-to-Digital Conversion）が必要になる。なお，AD変換する装置(回路)をAD変換器（Analog-to-Digital Convertor）と呼ぶ。

本実験では，AD変換の動作原理を理解した上で，マイコン内蔵のAD変換器を用いて半固定抵抗，温度センサ，照度センサの電圧変化をデジタル値としてマイコンに取り込み計測する実験を行う。さらに，ロボットのモータやブザーなどのアクチュエータを動かすためのPWM（Pulse Width Modulation：パルス幅変調）の動作原理を知る実験も行う。

実験目標：

- ・ AD 変換の動作原理を理解する。
- ・ アナログポート（AD 変換）を使いこなす。
- ・ センサ（アナログ）の動作原理を知る。
- ・ PWM の動作原理を理解する。
- ・ PWM によりブザーの音や LED の点灯制御を行う。

2.3.1 AD 変換

マイコンが直接処理できるのはデジタル値（0 または 1）である。一方，センサで検出できる温度や光，加速度などは，図 2.31 に示すような連続的な値 $f(t)$ として出力される。これは，時間軸が連続で値も連続なアナログ信号である。マイコンでこのようなアナログ信号を処理するには，AD 変換によってアナログ信号をデジタル信号に変換する必要がある。AD 変換は，概ね，

- ・ 一定の時間間隔でデータを取り込むサンプリング（sampling：標本化）する機能
 - ・ サンプリングした値をデジタル値に変換（量子化）する機能
- より構成される。次に，それぞれの機能について説明する。

・サンプリング（標本化）

図 2.31 のアナログ信号を時間間隔 T でサンプリングすると図 2.32 の○印のようになる。この一定の時間間隔 T を，サンプリング周期と呼ぶ。「アナログ信号をサンプリングする。」といった場合には，あらかじめ決められたサンプリング周期 T でアナログ信号を取得することを意味する。また，サンプリング周期 T の逆数を取り，サンプリング周波数 $f_s(=1/T)$ として表すこともある。なお，サンプリングした値は時間間隔 T の間は変化しないとする。アナログ信号（曲線）とサンプリングされた値との間には隙間があるが，サンプリング点の前後を差し引きすると概ねサンプリング点での値となる。実際の AD 変換においては，サンプリング処理のために入力値を一定時間保持（hold）する必要がある，これらを纏めてサンプル・ホールドと称する場合がある。

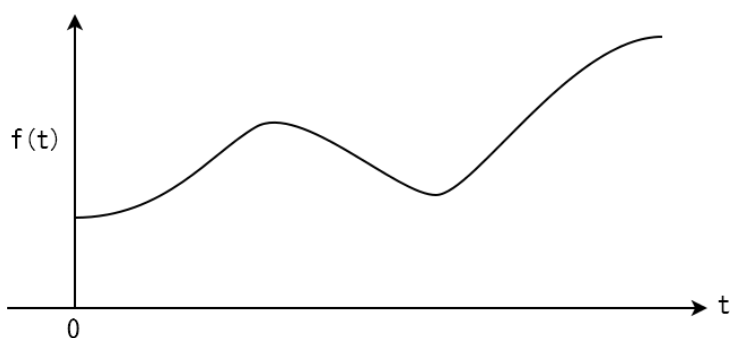


図 2.31 センサから出力されるアナログ信号（縦軸：電圧，横軸：時間）

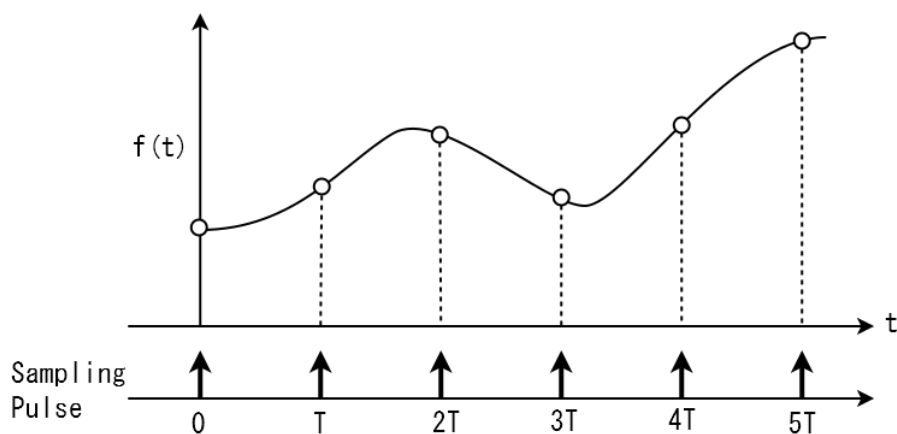


図 2.32 アナログ信号のサンプリング

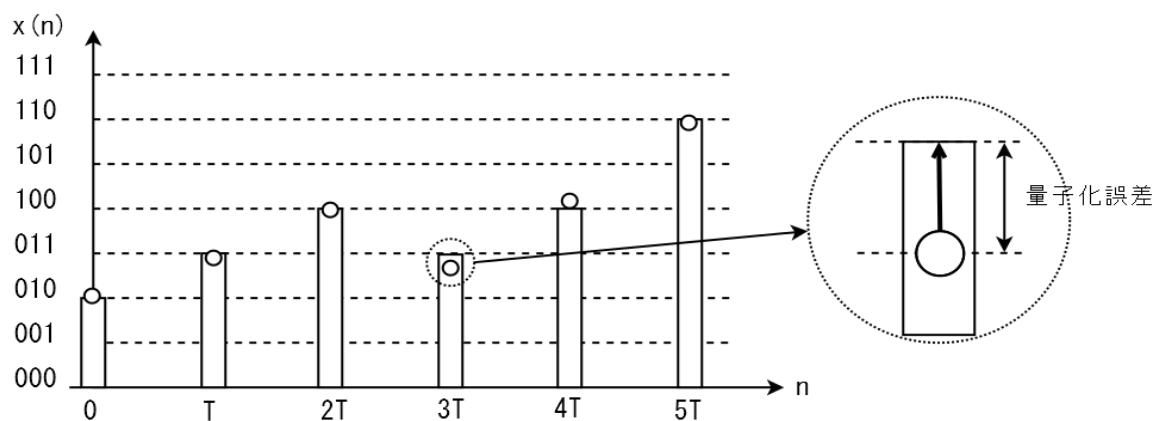


図 2.33 アナログ電圧の 3 ビット AD 変換回路による量子化

図 2.31 のアナログ信号をサンプリングするとき，サンプリング周期の決め方が重要になる。サンプリング周期 T が短い（時間間隔が狭い）方が，サンプルされる点が多くなり元のアナログ信号をより正確に表せることが直感的に理解できる。逆に，サンプリング周期 T が長い（時間間隔が広い）と元の信号とはかけ離れた信号になる。そこで，元のアナログ信号（原信号）を再現するために，適切なサンプリング周波数を与える「サンプリング定理」がある。サンプリング（標本化）定理は，「原信号に含まれる最高の周波数 f_c に対して，サンプリング周波数 f_s が，

$$f_s \geq 2 f_c \quad \text{あるいは} \quad f_s/2 \geq f_c$$

を満たす条件でサンプリングすることによって原信号を再現できる」とされている。したがって，対象となるセンサの値をサンプリングするときには，どの程度の時間で値が変化するか，ノイズ

がどの程度含まれるかなど，センサの出力特性をよく調べてからサンプリング周波数を決める必要がある。なお， $f_s/2$ をナイキスト周波数と呼ぶ。

・量子化

サンプリングはアナログ信号の時間軸における標本化であった。このサンプリングした値をデジタル値に変換することを量子化と呼ぶ。

説明を簡単にするため，ここでは 3 ビットの AD 変換器を例にする。図 2.33 に，アナログ信号の電圧を量子化する様子を示す。3 ビットの 2 進数は，000～111 の 8 通り (2 の 3 乗) のデジタル値で表せる。図 2.32 に示すように，サンプリングしたアナログ信号の電圧 (○印) を量子化すると，3 ビットのデジタル値のうち近い値に変換される。このとき生じる誤差は「量子化誤差」と呼ばれ，AD 変換において必ず発生する誤差である。AD 変換回路により変換できる電圧の最大振幅を FS (Full Scale) とすると，1 ビットあたりで表せる最小値 (1LSB(Least Significant Bit)) が求まる。

3 ビットの AD 変換器の場合には，

$$1\text{LSB} = \text{FS} / 2^n$$

の関係から， n はビット数なので $n=3$ として $\text{FS} / 8$ の分解能で電圧を量子化できる。図 2.33 を見ると，時刻 $n(n = 0, T, 2T, \dots)$ における量子化後のデジタル信号 $x(n)$ は，

$$x(0)=010, x(T)=011, x(2T)=100, x(3T)=011, x(4T)=100, x(5T)=110$$

と読み取れる。これを，2 進数から 10 進数に変換すると，

$$x(0)=2, x(T)=3, x(2T)=4, x(3T)=3, x(4T)=4, x(5T)=6$$

となる。マイコンではこれらの値が AD 変換結果として得られる。

AD 変換結果からアナログ電圧 $f(n)$ を求めるには，

$$f(n) = x(n) * (\text{FS} / 2^n) \quad \dots (2.1)$$

により電圧を算出できる。ここで，FS は AD 変換器のリファレンス (reference) 電圧である。リファレンス電圧とは AD 変換する際の基準となる電圧を示す。

たとえば，3 ビット AD 変換器，リファレンス電圧 5.0V のとき， $\text{FS}=5.0$ (V)， $n=3$ (Bit) から，

$$f(n) = x(n) * (5.0\text{V} / 8) = x(n) * 625 \text{ mV}$$

となる。つまり，AD 変換結果のデジタル値に 625mV (0.625V) を掛けるとアナログ電圧を求めることができる。したがって，マイコンの AD 変換結果から，

$$x(0)=1.25\text{V}, x(T)=1.875\text{V}, x(2T)=2.5\text{V}, x(3T)=1.875\text{V}, x(4T)=2.5\text{V}, x(5T)=3.75\text{V}$$

の電圧を計測したことになる。

2.3.2 Arduino UNO の AD 変換器

実験において使用する Arduino UNO に搭載されたマイコンには 10 ビット AD 変換器が 1 台内蔵されており、センサなどのアナログ信号を直接接続できる「アナログポート」を経由して使用できる。図 2.34 にデジタルポートとアナログポートの電圧範囲とデジタル値に変換した結果の比較を示す。2.1 節で演習を行ったとおり、マイコンのデジタルポートでは、 $-0.5\sim 0.3\text{V}$ の範囲を“0”， $0.6\sim V_{cc}+0.5\text{V}$ の範囲を“1”，として処理した。一方、アナログポートでは、内蔵 AD 変換器によって、 $0\sim 5\text{V}$ の電圧を 10 ビットの分解能でデジタル値“0~1023”に変換される。このとき、最小分解能は、前述の式(2.1) より約 4.9 mV ($FS=5.0\text{V} / 2^{10}=1024$)である。

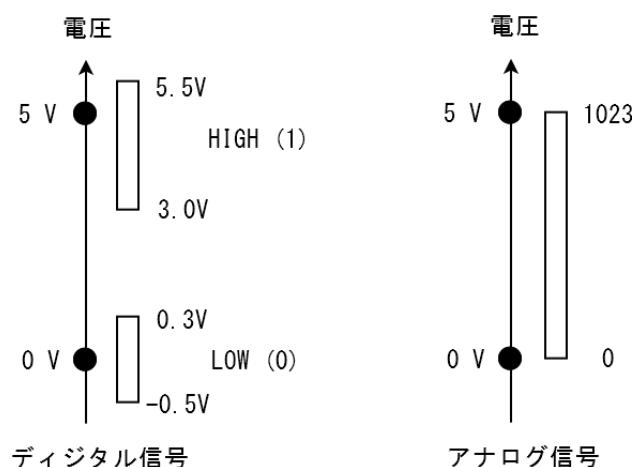


図 2.34 デジタル信号とアナログ信号の電圧範囲とデジタル変換結果の比較
(AVR ATmega328P マイコン，電源電圧 5.0V，AD 変換 10 ビットするとき)

2.3.3 Arduino UNO のアナログ入力ポート

Arduino UNO には、図 2.35 に示すように 6 つのアナログ入力ポート (A0~A5) がある。Arduino UNO の構造上、複数のアナログ入力ポートから 1 つを選択して AD 変換するため、複数のアナログ入力を AD 変換する場合には、次節で説明する **analogRead 関数** でアナログ入力ポートを切り替えながら AD 変換することになる。

2.3.4 AD 変換実験で使用する API

Arduino UNO のアナログ入力ポートを使用するための Arduino API 関数について説明する。**analogRead 関数**：指定したアナログ入力ポートの電圧 ($0\sim 5\text{V}$) を AD 変換し、デジタル値 (0~1023) として読み取る。2.3.2 節に記載したとおり AD 変換器の最小分解能は 4.9mV である。標準設定では AD 変換処理に 103μ 秒かかり、サンプリング周波数の上限は 9.7 kHz となる。このとき、2.3.1 節で説明した「サンプリング定理」から、入力されるアナログ信号の周波数の上限は半分の約 4.8kHz 以下とすること。複数のアナログ入力ポートを用いる場合は更に低下する。

書式：val = analogRead(pin);

引数：pin ピン番号 アナログ入力ポートの 0 から 5 ピン (A0, A1, A2, A3, A4, A5)

戻り値：0 から 1023 の整数値

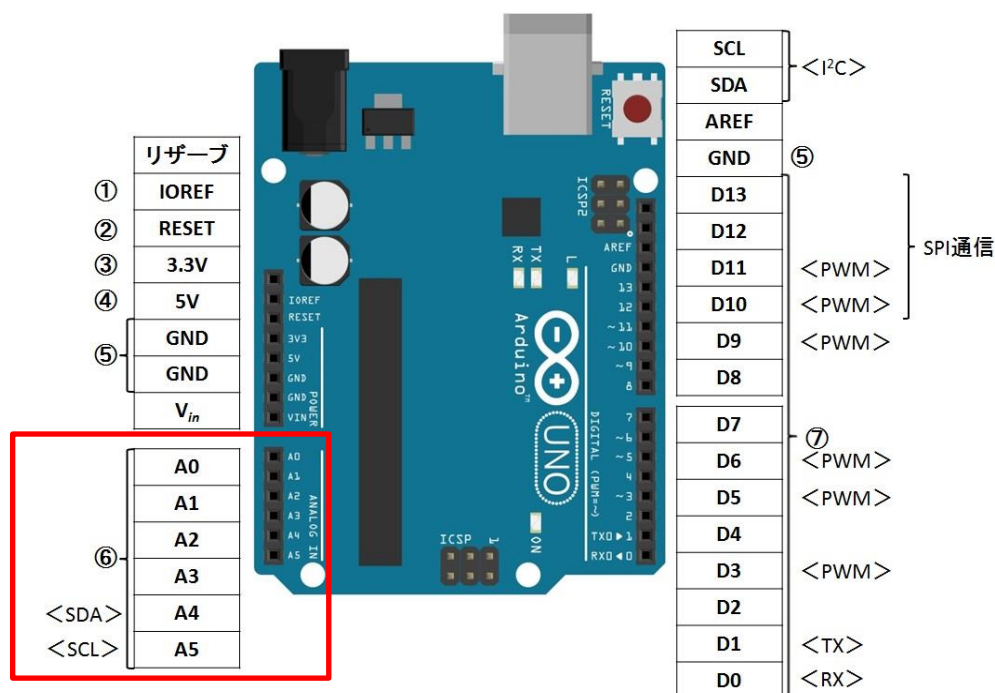


図 2.35 Arduino UNO マイコンボード（アナログ入力ポートのピン配置）

<演習 2.3.1> AD 変換器による抵抗により分圧された電圧の測定

マイコンの AD 変換器によってアナログ電圧がデジタル値に変換される過程を調べるため、ここでは、Arduino UNO のアナログ入力ポート（A0）に入力された抵抗の分圧による電圧を計測する。図 2.36 の回路(a)，(b)，(e)を順に作成して、それぞれアナログ入力ポートのデジタル値を調べ、表 2.15 を完成させよ。

マイコンのポート	ポートの入出力	接続先
A0	アナログ入力	抵抗の分圧

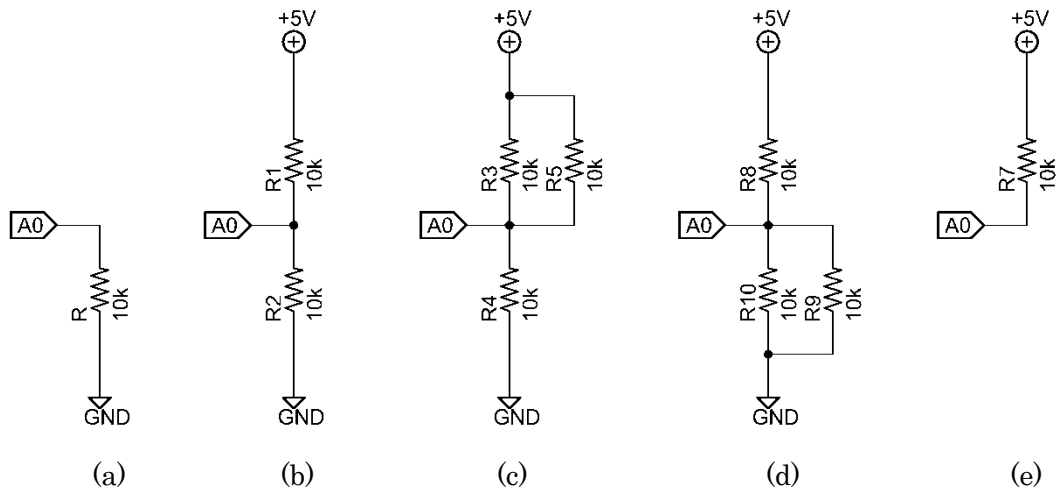


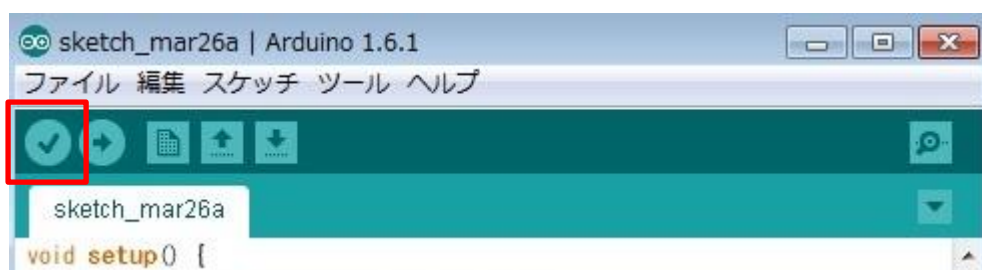
図 2.36 アナログポート(A0)に抵抗により分圧した電圧を加える回路

- (1) (a), (b), (e) のアナログポート(A0)に入力される電圧 V_0 をそれぞれ計算により求め、表 2.15 に記入せよ。ただし、電圧は小数第 2 位まで求めることとし、電源電圧のノイズや抵抗の誤差は考慮しなくてよい。
- (2) 図 2.36(a), (b), (e) の各回路について (3)以下を順次実施する。
- (3) **Arduino と PC を外す (USB 接続を抜く)。**
- (4) 回路を作成し、間違いがない事を確認する。
- (5) **Arduino と PC を接続 (USB 接続) する。**
- (6) サンプルプログラムを書き込み、電圧 V_0 の AD 変換結果のデジタル値を調べる。
なお、下記①～③は最初の 1 回のみで良い。
- ① サンプルスケッチを開く。
ファイル→スケッチの例→01:Basics→AnalogReadSerial を開く。

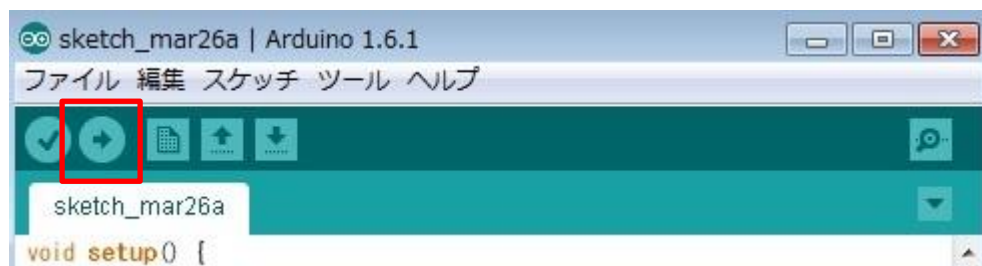
```
void setup() {
  Serial.begin(9600); // シリアル通信を 9600bps で初期化
}

void loop() {
  int sensorValue = analogRead(A0); // A0 ピンの AD 変換結果を取得する
  Serial.println(sensorValue); // シリアル通信により AD 変換結果を PC へ送信
  delay(1); // 待ち時間
}
```

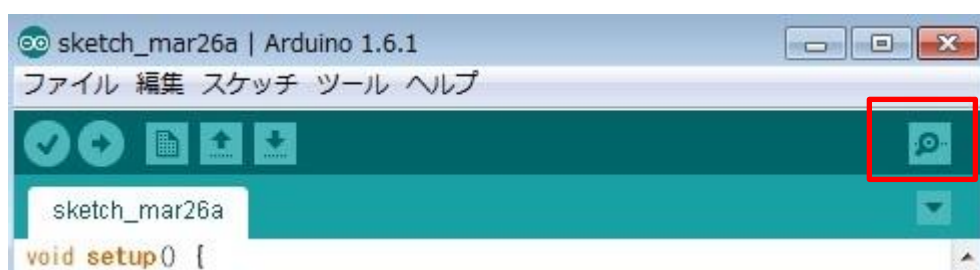
- ② 検証する (スケッチ→検証・コンパイル)。エラーがない事を確認する。



- ③ マイコンボードに書き込む（スケッチ→マイコンボードに書き込む）。



- ④ シリアルモニタを表示（ツール→シリアルモニタ）して，A/D 変換結果を確認する。



- (7) (a), (b), (e)のAD変換結果を表に記入し，式(2.1)から電圧を計算により求める。電圧は小数第2位までとする。

表 2.15：マイコンのAD変換器による抵抗の分圧による電圧の計測

回路番号	A0 の理論電圧 V_0 [V]	AD 変換結果	AD 変換結果から求めた電圧 [V]
(a)			
(b)			
(e)			

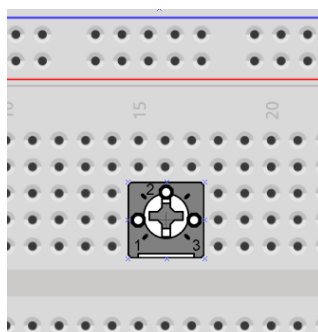
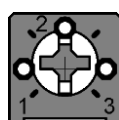
2.3.5 半固定抵抗

抵抗を半固定抵抗に変えてAD変換の実験を行う。実験に使用する半固定抵抗を図 2.37(a)に示す。ドライバーで中央+部分を回すことで抵抗値を変更できるため，テレビ，ラジオ，アンプなどの音量調整等によく利用される。半固定抵抗には3つの端子があり，図 2.37(b)のようにブレッドボードに配置して使用する。半固定抵抗の内部は，図 2.38(a)のように端子1と3を抵抗幕でつながるように配置され，つまみを回すことで端子2の摺動子が移動し，端子1と端子2間の抵抗値および端子2と端子3間の抵抗値が変わる。図 2.38(b)に半固定抵抗の等価回路を示す。半固定抵抗の端子1-3間に電圧 V_i を加え，端子2-3間を出力 V_o とすると，オームの法則より

$$V_o = (R_2 / (R_1 + R_2)) * V_i$$

の関係がある。

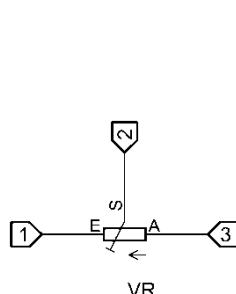
たとえば、半固定抵抗の抵抗値が $10\text{ k}\Omega$ の場合には、 $R1 + R2 = 10\text{ k}\Omega$ となり、可変抵抗のボリュームをまわすと $R2$ の値が 0Ω から $10\text{ k}\Omega$ の範囲で変化する。



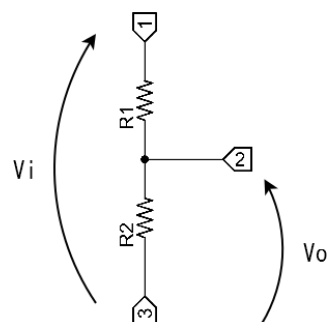
(a) 半固定抵抗

(b) ブレッドボード配置 (向きに注意)

図 2.37 半固定抵抗とブレッドボード配置



(a) 内部構成



(b) 半固定抵抗の等価回路

図 2.38 半固定抵抗の内部構成と等価回路

<演習 2.3.2> AD 変換器による半固定抵抗の電圧変化の測定

図 2.39 の回路をブレッドボード上に実装し、半固定抵抗の抵抗値を調整することで、シリアルモニタに表示される電圧が変わることを確認する。

マイコンのポート	ポートの入出力	接続先
A0	アナログ入力	半固定抵抗

- (1) **Arduino と PC を外す (USB 接続を抜く)。**
- (2) 図 2.39 の回路をブレッドボード上に実装する。間違いがないことを確認する。
- (3) **Arduino と PC を接続 (USB 接続) する。**
- (4) サンプルプログラムを書き込み AD 変換結果から電圧を求める。
 - ① サンプルスケッチを開く。
ファイル→スケッチの例→01:Basics→AnalogReadSerial を開く。
 - ② サンプルスケッチに、式(2.1)より電圧を求める処理を追加する。


```
float vo = sensorValue * (5.0 / 1024.0);
```

```
Serial.println(vo);          // Serial.println(sensorValue); を置換
```

- ③ スケッチの保存（ファイル→名前を付けて保存）（適当な名前をつける）
 - ④ 検証する（スケッチ→検証・コンパイル）。エラーがない事を確認する。
 - ⑤ マイコンボードに書き込む（スケッチ→マイコンボードに書き込む）。
 - ⑥ シリアルモニタを表示する（ツール→シリアルモニタ）。
- (5) 半固定抵抗をまわして、シリアルモニタにて電圧 V_o の変化を確認する。

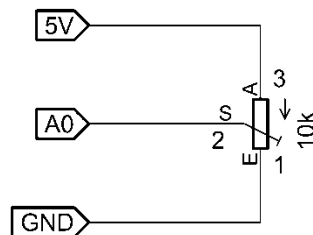


図 2.39 半固定抵抗のアナログ入力ポート（A0）への接続

<課題 2.3.1> 半固定抵抗による LED の点灯・消灯

図 2.40 の回路をブレッドボード上に実装し、演習 2.3.2 で作成したプログラムを用いて半固定抵抗からの入力電圧の範囲をシリアルモニタにより確認せよ。また、電圧 $2.5V$ を閾値として、図 2.41 のように半固定抵抗からの入力電圧が閾値以上のとき LED を点灯、閾値未満で消灯するプログラムを作成せよ。

マイコンのポート	ポートの入出力	接続先
A0	アナログ入力	半固定抵抗
D3	デジタル出力	LED（赤）

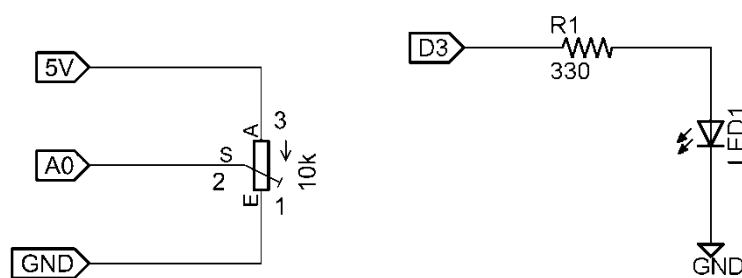


図 2.40 半固定抵抗と LED 回路

- (1) Arduino と PC を外す（USB 接続を抜く）。
- (2) 図 2.40 の回路をブレッドボード上に作成する。間違いがないことを確認する。
- (3) Arduino と PC を接続（USB 接続）する。
- (4) 演習 2.3.2 で作成した電圧変化を測定するプログラムを書き込み、実行する。
 - ① 演習 2.3.2 で作成し保存したスケッチを開く。

- ② 検証する（スケッチ→検証・コンパイル）。エラーがない事を確認する。
 - ③ マイコンボードに書き込む（スケッチ→マイコンボードに書き込む）。
 - ④ シリアルモニタを表示する（ツール→シリアルモニタ）。
 - ⑤ シリアルモニタを見ながら半固定抵抗をまわして電圧が変化することを確認する。
- (5) 図 2.41 のように、電圧 2.5V を閾値（threshold）として、
- ・ 閾値以上のとき LED を点灯、
 - ・ 閾値未満のとき LED を消灯するプログラムを作成して(4)と同様の手順で実行する。
- (6) スケッチの保存（ファイル、名前を付けて保存）

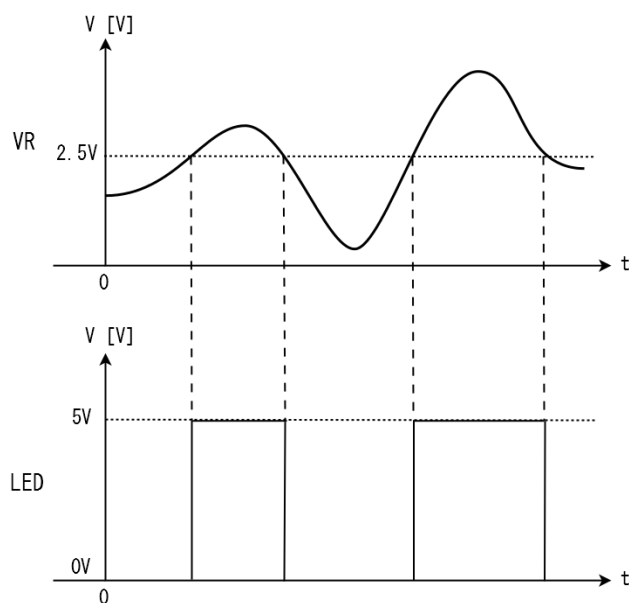


図 2.41 半固定抵抗による電圧変化と LED の点灯・消灯

2.3.6 センサ

人間は、目、鼻、耳、口、手のもつ感覚、五感（視覚、嗅覚、聴覚、味覚、触覚）によって周囲の状態を知ることができる。一方、マイコンで周囲の環境を計測するには、表 2.16 に挙げるセンサが用いられる。センサは、いろいろな物理量を検出して電気信号に変換する電子部品であり、センサによって検出できる温度や光、圧力などはアナログ値である。そのため、マイコンで処理するには AD 変換器を用いてディジタル値に変換する必要がある。このディジタル値をセンサの特性に応じた変換式を適用して物理量に変換することで、実際の照度、温度、距離、加速度などを知ることができる。この変換式は、センサのデータシートに記載されたグラフや式から算出し、プログラムとして実装する（センサによっては直接ディジタル値を出力するものもある）。

表 2.16 : センサの種類

検出対象	センサの種類
光	照度センサ、カラーセンサ、イメージセンサ
音、振動	マイクロフォン、超音波センサ
温度、湿度	温度センサ、サーミスタ、熱電対、湿度センサ

圧力	圧力センサ，歪みゲージ
加速度，地磁気	加速度センサ，地磁気センサ
角速度	ジャイロセンサ
磁力	ホールセンサ

2.3.7 実験に用いるセンサ

実験では，照度センサを用いた計測を行う。照度センサは光の強さを測定するセンサである。ここでは，照度センサの測定原理について説明する。

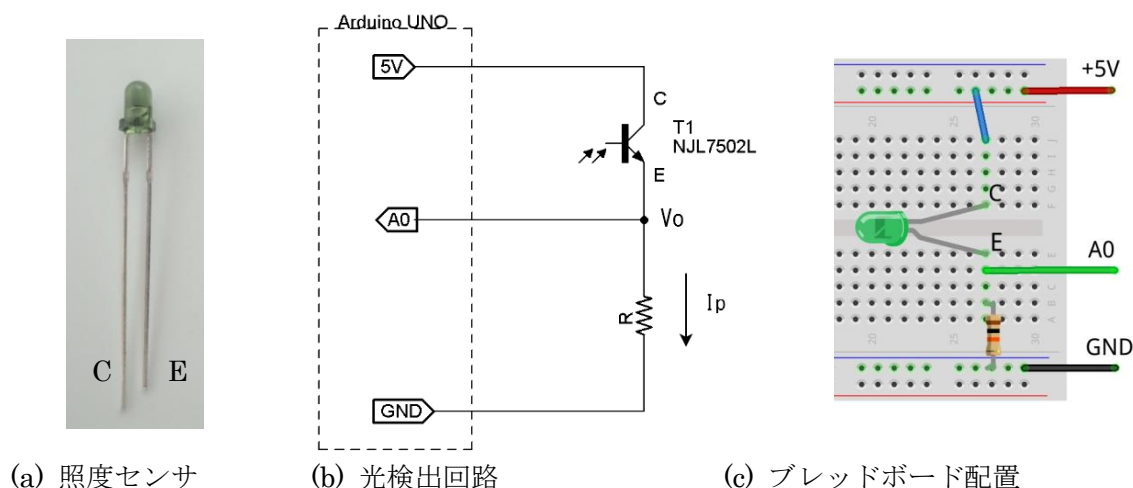


図 2.42 照度センサ NJL7502L のピン配置と光検出回路

(1) 照度センサの測定原理

照度センサ(NJL7502L)は，人間の視感度に近い特性を示す光を検出するセンサである。一般にフォトトランジスタと呼ばれる光の強さを電流に変換するフォトダイオードと電流を増幅するトランジスタで構成される。図 2.42(a)に照度センサの外観を示す。照度センサのリード線の長い方がコレクタ(C)，短い方がエミッタ (E)である。図 2.42(b)のように照度センサを接続すると，センサが光を受けると，受けた光の強さに応じた光電流 I_p がコレクタ・エミッタ間で流れる。光電流 I_p は接続された抵抗 R によって電圧 V_o に変換され，

$$V_o = R * I_p \quad \dots (2.2)$$

の電圧が発生する。すなわち，光が強いと電圧が高くなり，光が弱いと電圧が低くなる。この電圧をマイコンのアナログ入力ポートにて AD 変換することで光の強弱を計測できる。

一方，光電流と照度は，データシートより図 2.43 の関係があり，照度を L [lx]とすると，

$$L = 2.22 I_p \quad \dots (2.3)$$

により求められる。ここで， I_p の単位は μA である (μ は 10 の -6 乗)。また，照度は平面状の物体に照射された光の明るさを表す物理量で単位は lx (ルクス) である。式(2.3)の関係から光電流 I_p が計測できると，直ちに照度 L を計算により求めることができる。

たとえば，フォトトランジスタに接続した抵抗 R を $10\text{ k}\Omega$ として照度の計測を行う。マイコンの AD 変換器で計測したデジタル値 $x(n)$ から照度 L を求めるには，式(2.2)から光電流 I_p は，

$$I_p = V_o / R = 100 V_o [\mu A] \quad (\text{電流の単位変化に注意})$$

となり、式(2.3)の関係から照度 L は、

$$L = 222 V_o \quad \dots (2.4)$$

と求めることができる。これに電圧 V_o を代入すれば良い。電圧 V_o は AD 変換結果のデジタル値を $x(n)$ として、

$$V_o = x(n) * (V_{ref} / 1024) \quad \dots (2.5)$$

である。ここで、AD 変換器のリファレンス電圧 V_{ref} は 5.0 V である。これらを纏めると、

$$L = 222 * x(n) * (5.0 / 1024) \quad \dots (2.6)$$

となる。

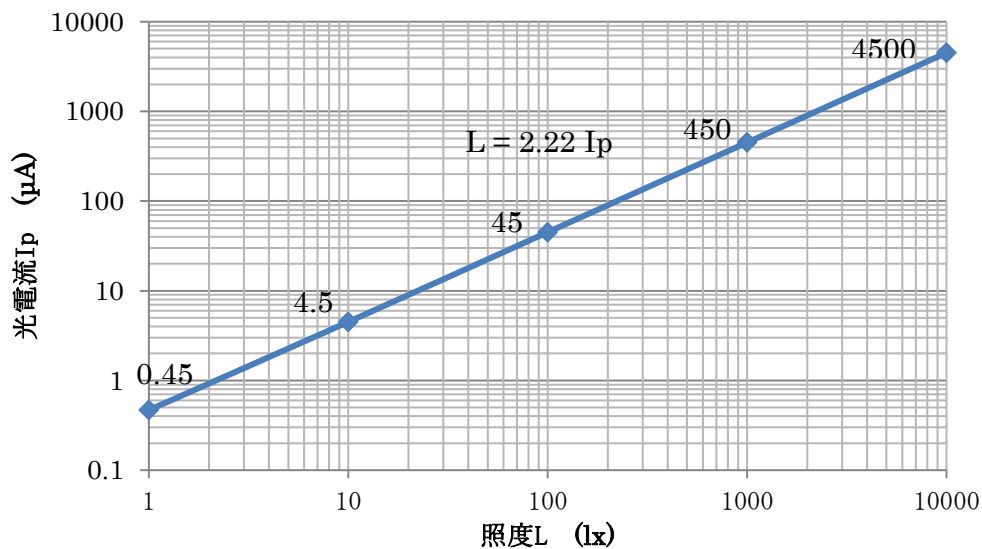


図 2.43 照度と光電流の関係(NJL7502L)

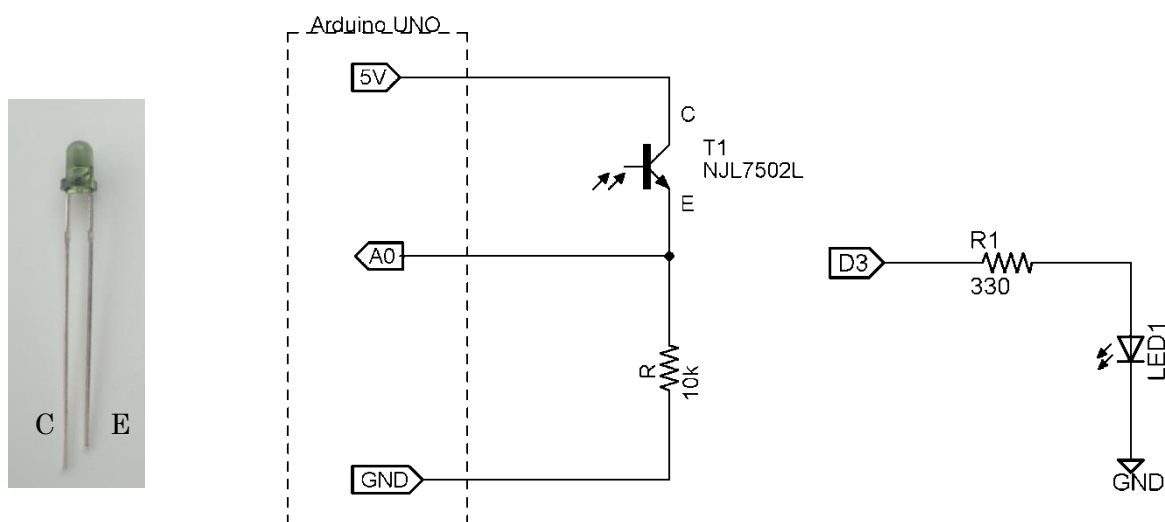
<演習 2.3.3> 照度センサによる照度の計測

図 2.44(b)の光検出回路をブレッドボード上に作成し、照度センサからの電圧入力を計測して照度を求めるプログラムを作成する。

マイコンのポート	ポートの入出力	接続先
A0	アナログ入力	照度センサ

- (1) 図 2.44(b)の回路をブレッドボード上に作成して、照度センサをマイコンのアナログ入力ポート (A0) に接続せよ。照度センサは深緑色のパッケージのため、LED の緑と間違えないこと。また、ピン配置には、コレクタ (C)、エミッタ (E) の向きがあるので注意すること。
- (2) 演習 2.3.2 の AD 変換結果から電圧を求めるプログラムに、電圧を照度に変換する式 (2.4) を追加して、照度が表示されるプログラムを作成する。

- (3) プログラムを実行して、シリアルモニタにより照度値を確認する。



(a) 照度センサの外観 (b) 光検出回路 (c) LED 回路

図 2.44 照度センサと回路図 ((a), (b)は図 2.41 から再掲載)

<課題 2.3.2> 照度センサによる照度の計測, LED の点灯・消灯

図 2.44(b)(c)の光検出回路と LED 回路をブレッドボード上に実装し、照度に応じて LED が点灯、消灯するプログラムを作成せよ。

マイコンのポート	ポートの入出力	接続先
A0	アナログ入力	照度センサ
D3	デジタル出力	LED (赤)

- 図 2.44(b)(c)の回路をブレッドボード上に作成して、照度センサをマイコンのアナログ入力ポート (A0) に、LED をデジタル出力ポート (D3) にそれぞれ接続せよ。
- 演習 2.3.3 で作成したプログラムを実行して、次の状態の照度をシリアルモニタにより確認せよ。

照度センサの状態	照度 (lx)
まわりに障害物がない状態	
手で覆った状態	

- まわりに障害物がない状態を照度の最大値、手で覆ったときを最小値として中間値を求めよ。
- 中間値を閾値(th_val)として設定し、閾値以上るとき “HIGH”，閾値未満のとき “LOW” を返す lux_threshold 関数を作成せよ。例) int lux_threshold(float lux, int th_val);
- (4)で作成した関数を用いて、閾値以上るときに LED が消灯、閾値未満のとき LED が点灯す

るプログラムを作成して実行せよ。

2.3.7 PWM

Arduino のデジタルポートは 5V または 0V の電圧しか出力できない。しかし、5V と 0V を繰り返す PWM (Pulse Width Modulation : パルス幅変調) という方法を用いてアナログ値相当を出力することができる。本実験では、LED やブザー、サーボモータを駆動するために使用する。

PWM の概要を図 2.45 に示す。PWM 波は、振幅 V 、一定周期 T のパルス波の繰り返しにより生成される。パルス波 1 周期の ON 時間と OFF 時間の割合はデューティ比と呼ばれる。図 2.46 のようにデューティ比を変化させることは平均電圧 V_0 を変化させることに相当する。つまり、デューティ比が小さいと平均電圧は低くなり、デューティ比が大きいと平均電圧は高くなる。これにより、LED の明るさやモータの回転速度の調整に応用できる。

デューティ比 D は、PWM 波の周期を T 、ON 時間を τ とすると、

$$D = \tau / T \quad \dots (2.6)$$

と定義できる。デューティ比は、0~1.0 または 0%~100% の範囲で表される。また、PWM 波の振幅 V から見かけの平均電圧 V_0 は、

$$V_0 = D * V \quad \dots (2.7)$$

となる。たとえば、PWM 波の振幅が 5V、周期 $T=2.0[\text{ms}]$ 、ON 時間 $\tau=0.5[\text{ms}]$ のとき、デューティ比 $D=0.25$ となり平均電圧 $V_0=5\text{V} \times 0.25=1.25\text{V}$ となる。課題 2.1.3 では、LED の明るさ調整を抵抗の変更により行っていたが、PWM 波を用いると抵抗は一定値のままで高速に電源を ON,OFF して、ON 時間を調整するだけで明るさを変えられる。さらに、1 周期の OFF 時間には電力がほとんど消費されないため効率的な制御方法である。

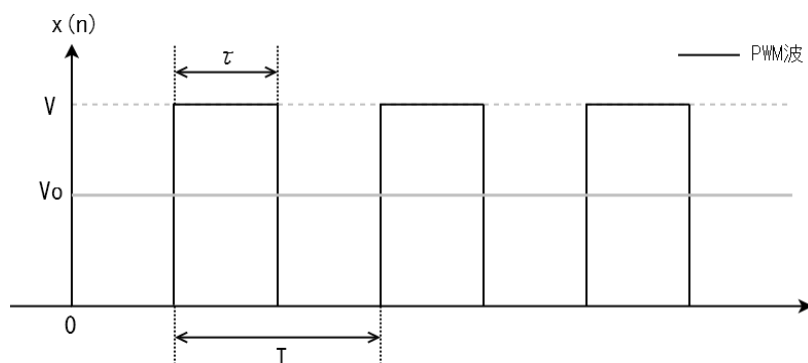


図 2.45 PWM の概要

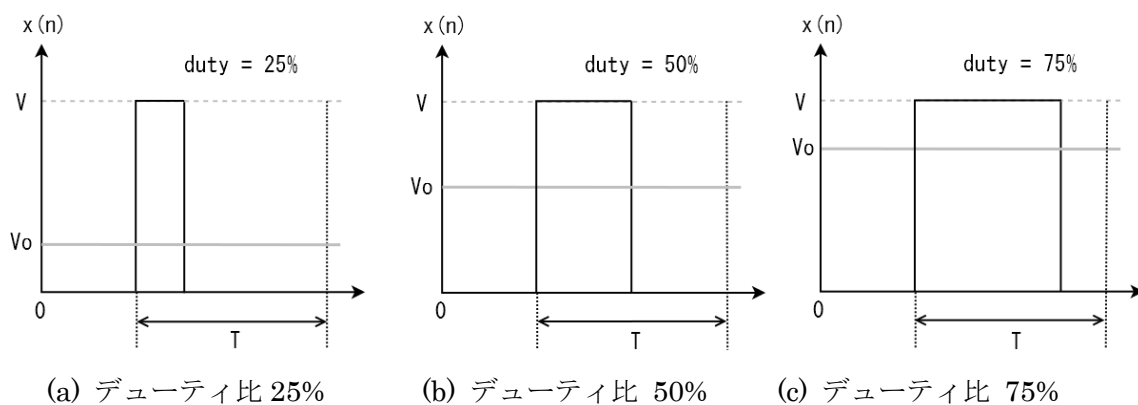


図 2.46 デューティ比を変化させたときの PWM 波形 (1 周期)

2.3.8 Arduino UNO のデジタル PWM 出力ポート

マイコンからモータやブザー, LED などのアクチュエータを駆動するにはデジタル PWM 出力ポートを用いる。Arduino UNO には, 図 2.47 に示すように 6 つのデジタル PWM 出力ポート (D3,D5,D6,D9,D10,D11) がある。デジタル PWM 出力ポートは, それぞれマイコン内蔵のタイマにより生成される PWM 出力回路に接続されており, 前節で説明した PWM 波を出力できる。なお, PWM 出力を実際にアナログ電圧として利用するためには出力に平滑回路を接続して, 1 周期分の電圧変化を平滑化する必要がある (今回の実験ではアナログ電圧値としては用いない)。

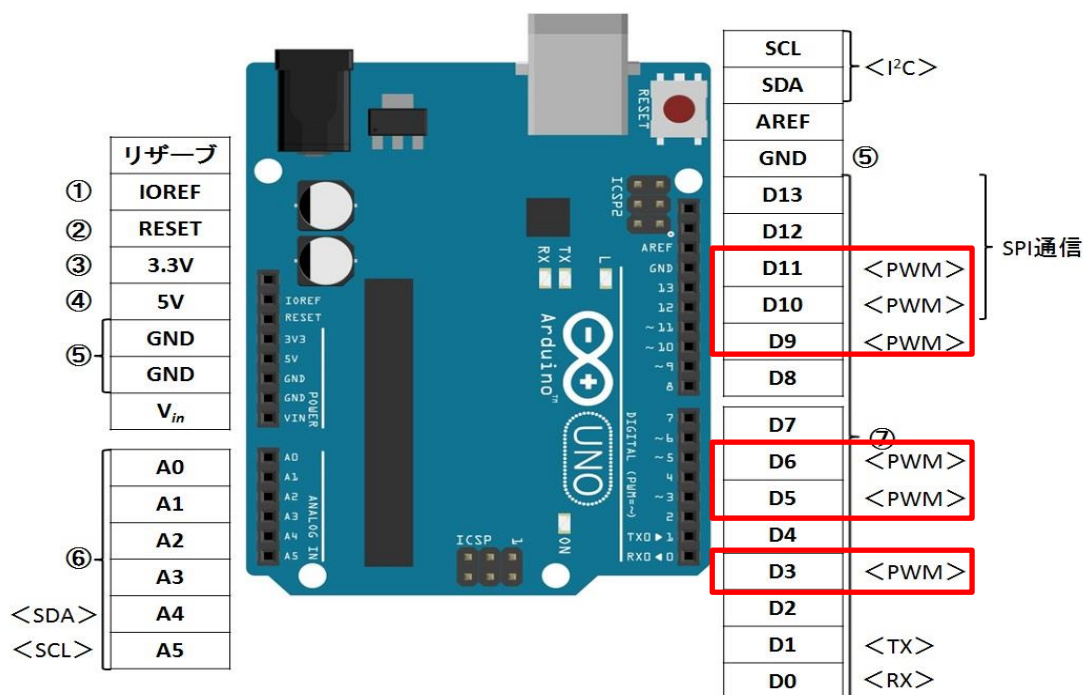


図 2.47 Arduino UNO マイコンボード (デジタル PWM 出力ポートのピン配置)

2.3.8 実験に使用する API

Arduino UNO のデジタル PWM 出力ポートを使用するための Arduino API 関数について説明する。

analogWrite 関数 : 指定したデジタルピンから PWM 波を出力する。Arduino UNO ボードの

基板上に～マークがあるデジタルポート（D3, D5, D6, D9, D10, D11）に対してピン設定ができる。analogWrite 関数によって出力できる PWM 波の周波数は、約 490 Hz（D3, D9, D10, D11）と約 980 Hz（D5, D6）である。

analogWrite 関数の引数 val に 0 を指定すると、0V（デューティ比 0%）が出力され、255 を指定すると 5V（デューティ比 100%）が出力される。引数 val に 128 を指定すると約 50%のデューティ比の PWM 波が出力され、デジタルポートから出力される 1 周期の平均電圧は 2.5V となる。つまり、analogWrite に指定する val は、PWM 波のデューティ比を D（0%から 100%）として式(2.8)より求められる。

$$\text{val} = D * 255 \quad \dots (2.8)$$

書式：analogWrite(pin, val);

引数 1：pin ピン番号（3, 5, 6, 9, 10, 11）

引数 2：val デューティ比の ON 時間 τ （0 から 255）

戻り値：なし

<演習 2.3.4> デジタル PWM 出力による LED の明るさ調整 1

LED の光の強さを抵抗値により調整できることは既の実験を行った。ここでは、デジタル PWM 出力ポートから PWM 波を出力することで LED の光の強さを制御する。analogWrite の引数を変えることで、指定のデューティ比の PWM 波を生成したときの LED の状態を確認せよ。

マイコンのポート	ポートの入出力	接続先
D9	デジタル出力（PWM）	LED（赤）
D11	デジタル出力（PWM）	LED（黄）

(1) 表の analogWrite の引数 val を求める。

デューティ比	analogWrite の引数(val)
0%	
25%	
50%	
75%	
100%	

(2) 図 2.48 の回路をブレッドボード上に実装する。

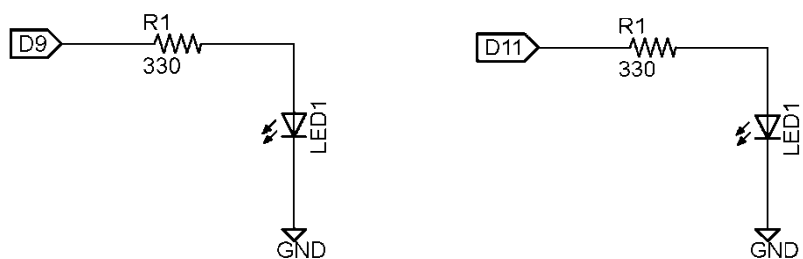


図 2.48 PWM 波による LED の明るさ制御回路(LED (赤) (D9), LED (黄) (D11))

- (3) デューティ比が 25%の PWM 波を LED (赤) に, デューティ比が 75%の PWM 波を LED (黄) (D11) に出力するプログラムを, `analogWrite` 関数を用いて作成し実行せよ。
- (4) LED (赤) と LED (黄) の明るさを見比べ, PWM 波のデューティ比の違いを確認せよ。

<課題 2.3.3> デジタル PWM 出力による LED の明るさ調整 2

図 2.49 の回路をブレッドボード上に作成あるいは演習 2.3.4 の回路 (図 2.48 左) をそのまま利用し, PWM 波のデューティ比を時間とともに変化させて, LED の明るさを調整するプログラムを作成し, 動作させよ。

マイコンのポート	ポートの入出力	接続先
D9	デジタル出力 (PWM)	LED (赤)

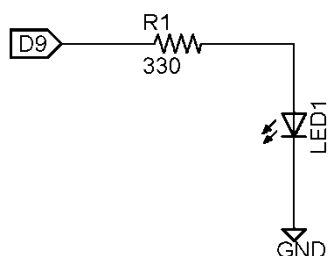


図 2.49 LED 回路

- (1) 図 2.49 の LED 回路をブレッドボード上に作成あるいは演習 2.3.4 の回路を利用する。

- (2) 図 2.50 を参考に、PWM 波のデューティ比をステップ状に指定時間で変化するプログラムを作成して実行せよ。そのとき LED の状態を確認せよ。(ヒント：for ループ，delay 関数を使用する。)

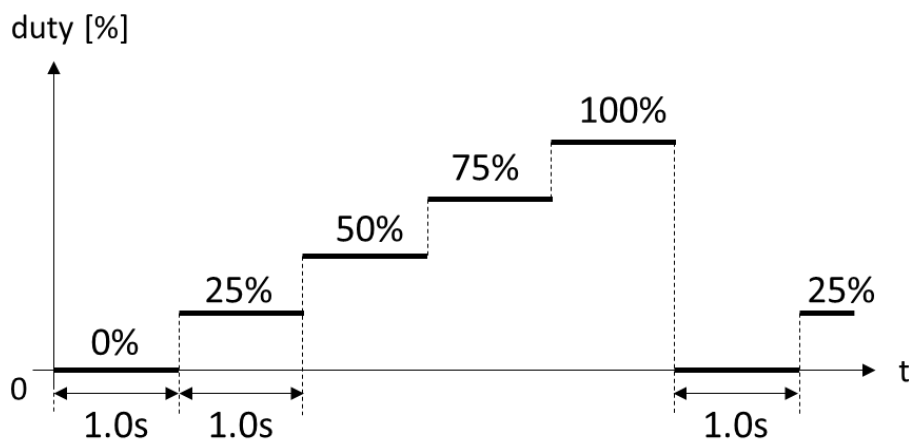
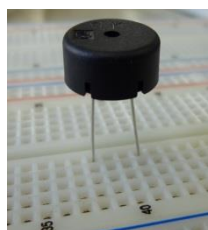


図 2.50 PWM 波のデューティ比変化波形

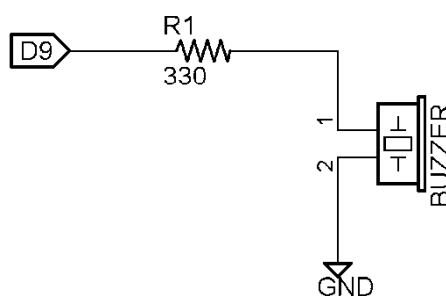
<発展課題 2.3.1> デジタル PWM 出力によるブザーの駆動

図 2.51(b)の圧電ブザーの回路をブレッドボード上に作成して、課題 2.3.3 の PWM 波を圧電ブザーに加えて音を鳴らせ。

- (1) 図 2.51(b)の圧電ブザーの回路をブレッドボード上に作成する。ただし、圧電ブザーは図 2.51(a)の向きに配置すること。
- (2) 課題 2.3.3 のプログラムをそれぞれ実行する。
- (3) 圧電ブザーの音を鳴らして音の変化を確認する。



(a) 圧電ブザー



(b) 回路図

図 2.51 圧電ブザーと回路図

<発展課題 2.3.2> アナログ入力とデジタル PWM 出力 2：照度センサによる LED の明るさ調整

課題 2.3.2 の回路の照度センサを用いた、照度に連動して LED の明るさを調整するプログラムを作成せよ。また、シリアルモニタでデジタル PWM 出力の値が変化することも確認せよ。ただし、課題 2.3.2 の結果から得られる照度の最小値 (lux_min)，最大値 (lux_max) を用いて、照

度値を LED のデジタル PWM 出力の最小値 0, 最大値 255 に対応させよ。

このとき, 照度値からデジタル PWM 出力値に変換する関数を作成せよ。

```
int convert(int lux, int lux_min, lux_max);
```

<発展課題 2.3.3> デジタル PWM 出力による LED の明るさ調整 2

図 2.52 の回路をブレッドボード上に作成し, PWM 波のデューティ比を時間とともに連続的に変化させて, LED の明るさを調整するプログラムを作成せよ。

マイコンのポート	ポートの入出力	接続先
D9	デジタル出力 (PWM)	LED (緑)

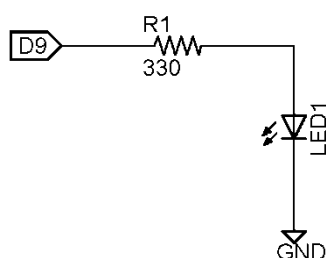


図 2.52 LED 回路

- (1) 図 2.52 の LED 回路をブレッドボード上に作成する。
- (2) (a)のこぎり波(図 2.53)および(b)の三角波(図 2.54)を参考に, PWM 波のデューティ比を指定の時間間隔で変化するプログラム 1, 2 をそれぞれ作成して実行せよ。そのとき LED の状態を確認せよ。(ヒント: for ループ, delay 関数を使用する。)

(a) のこぎり波

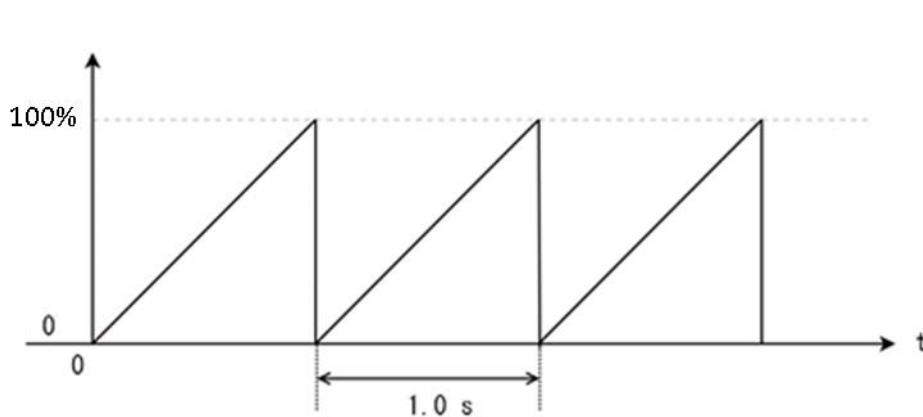


図 2.53 PWM 波のデューティ比変化 (のこぎり波)

(b) 三角波

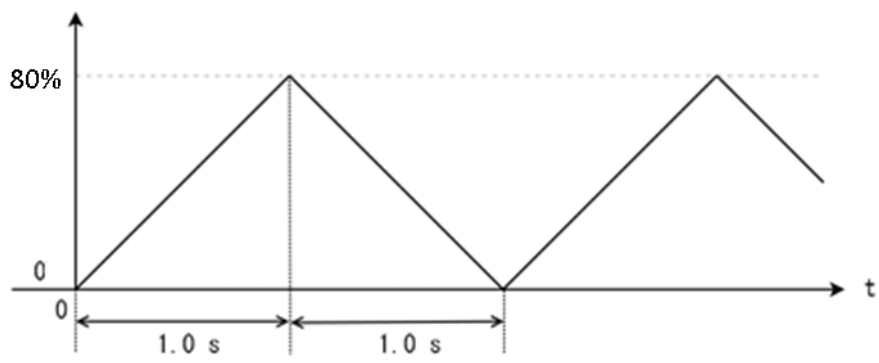


図 2.54 PWM 波のデューティ比変化（三角波）

【レポート 2.3 締切：次回実験日の 12:50 】

※スケッチには詳細のコメントを記述すること。コメントが無いものは採点対象外とする。

レポート 2.3.1 電圧、デジタル値との関係

レポート概要：演習 2.3.1 の結果を報告し、その結果から AD 変換結果（デジタル値）と電圧のグラフを作成せよ（一つのグラフに記載すること、縦軸： V_0 、横軸：AD 変換結果）

- ・ AD 変換結果（デジタル値）－ $A0$ の計算上の電圧 V_0 [V]
- ・ AD 変換結果（デジタル値）－AD 変換結果から求めた電圧 [V]

レポート 2.3.2 半固定抵抗による LED の点灯・消灯

レポート概要：課題 2.3.1 において実装、ブレッドボード配線図およびプログラムを報告せよ。シリアルモニタで確認した値と LED の点灯・消灯の状況を報告せよ。さらに、プログラム作成時に工夫した点を記せ。また、抵抗値を調整することで、なぜ図 2.40 の $A0$ の電圧が変化するのか、 $A0$ における電圧を図 2.38(b)を参考にして考察せよ。

レポート 2.3.3 照度センサによる LED の点灯・消灯

レポート概要：課題 2.3.2 において実装したブレッドボード配線図、プログラムおよびプログラムの実行結果を報告せよ。また、プログラム作成において工夫した点を記せ。

演習 2.3.3 で作成したプログラムによる、シリアルモニタで確認した照度の最大値および最小値を報告せよ。作成したプログラムによる、明るさの変化と LED の点灯・消灯の状況を報告せよ。

レポート 2.3.4 デジタル PWM 出力による LED の明るさ調整

レポート概要：演習 2.3.4 で実装したブレッドボード配線図、プログラムおよび演習結果（表）および `analogWrite` の引数と 2 つの LED の明るさを見比べた結果を報告せよ。また、PWM 制御によりなぜ LED の明るさが変化するのか、その理由を考察せよ。

レポート 2.3.5 デジタル PWM 出力による LED の明るさ調整 2

レポート概要：課題 2.3.3 で実装した、ブレッドボード配線図、プログラムおよび実験結果を報告せよ。また、プログラム作成において工夫した点を記せ。

レポート 2.3.6 発展課題 2.3.1

レポート概要：発展課題 2.3.1 で実装したブレッドボード配線図、プログラムおよび実験結果を報告せよ。また、圧電ブザーが鳴る原理について調査せよ。

レポート 2.3.7 発展課題 2.3.2

レポート概要：発展課題 2.3.2 で実装したブレッドボード配線図、プログラムおよび実験結果を報告せよ。また、作成したプログラムで工夫した点を記せ。

レポート 2.3.8 発展課題 2.3.3

レポート概要：発展課題 2.3.3 で実装したブレッドボード配線図，プログラムおよび実験結果を報告せよ。2つの PWM の変化方法により，LED の変化状態がどのように違ったのかを報告し，なぜ違いが生じているのかを考察せよ。さらに，プログラム作成において工夫した点を記せ。

・AD 変換実験使用部品一覧

表 1 使用部品一覧

部品	個数	備考
Arduino Uno	1	
USBケーブル Aオス-Bオス 1.5m A-B	1	
ブレッドボード	1	
ブレッドボード・ジャンパーワイヤ	1	
3mm 赤色LED	1	
3mm 黄色LED	1	
3mm 黄緑色LED	1	LED（緑）相当
半固定抵抗 10k Ω	1	
1/4W 330 Ω	2	
1/4W 10k Ω	2	
タクトスイッチ	1	
照度センサ	1	
圧電スピーカ	1	

参考図書

- [1] Massimo Banzi, 船田 巧：Arduino をはじめよう 一第 2 版一，株式会社オライリー・ジャパン（2014）.
- [2] 河連 庸子，山崎 文徳，神原 健：Arduino スーパーナビゲーション，株式会社リックテレコム（2012）.
- [3] 神崎 康宏：Arduino で計る，測る，量る，CQ 出版株式会社（2013）.