講習会

Arduino

4回目

アナログ入力と出力

アナログとは？

- 連続した量（例えば時間）を他の連続した量（例えば角度）で表示すること。
- 時計や温度計などがその例である。
- エレクトロニクスの場合、情報を電圧・電流などの物理量で表すのがアナログ、数字で表すのがデジタルである。

長所

情報（信号）をアナログ的に処理することの長所として以下のようなものがある。

- 瞬間的、直感的な情報処理が可能。スピードメーターの視認やオペアンプによる演算がデジタルと比べると高速である。
- デジタルにある量子化誤差が存在しない。
- 連続時間処理であるため、クロックジェネレータ等は不要。

短所

対して短所は次の点である。

- 内部の熱雑音等の物理的な影響を受ける。
- 外部からの擾乱（雑音など）の影響を受けやすい。
- 保存・複製・転送による劣化が生じる。
- 一旦、誤差が生じると復元出来ない。デジタルはエラー訂正等で程度にもよるが修復が可能である。
- 時間軸が基本的に連続処理（アナログ）であり時間軸補正は困難である。

アナログとして使えるピン

- 入力
- A0~A5（アナログ入力ピン）

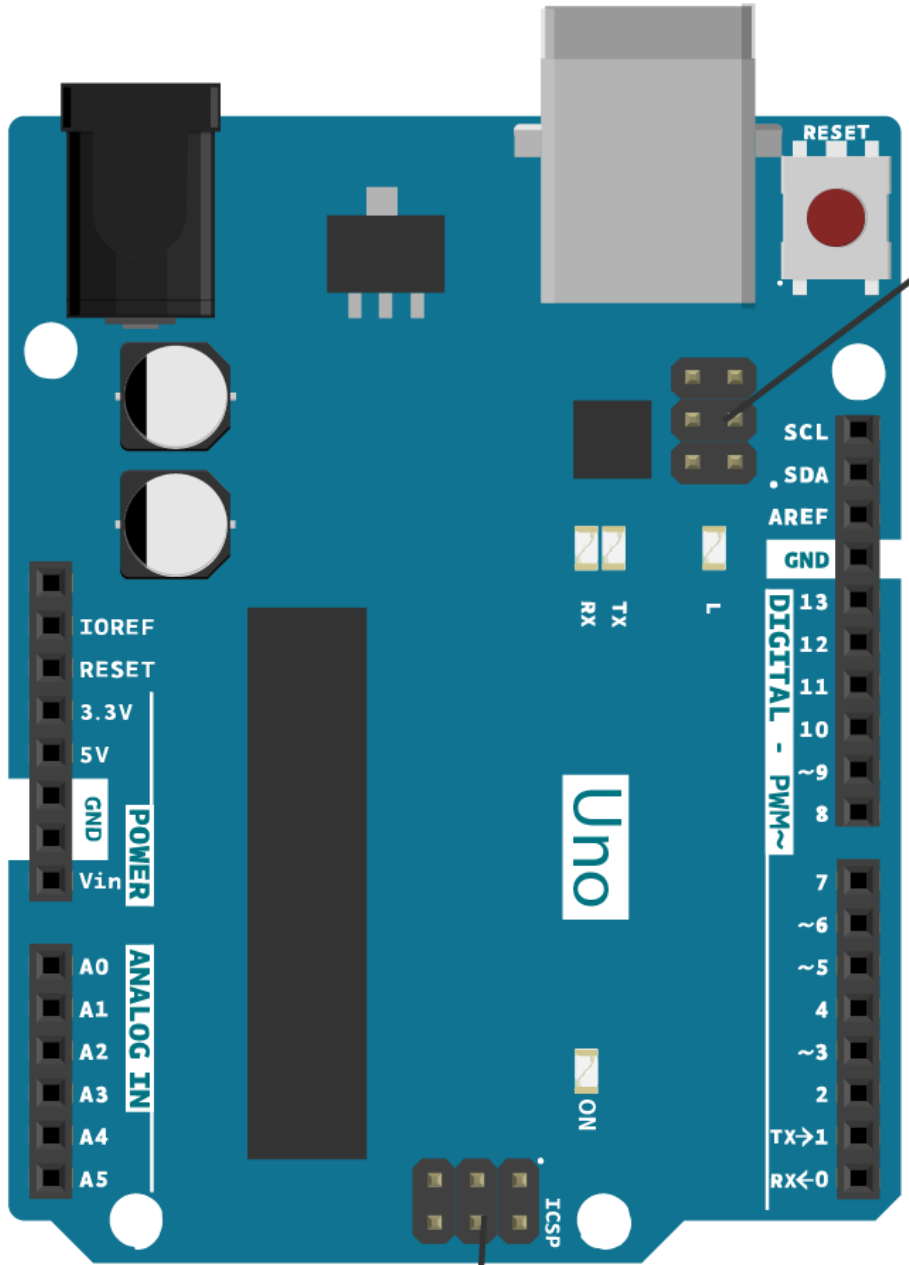
- 出力
- D3,D5,D6,D9,D10,D11（PWM~マークの付いているピン）

IORef: 5V

Vin: 7-12V DC max.

Serial: Serial is attached to pins 0 and 1, and to the USB-Serial micrcontroller on board.

The Uno has a second microcontroller on board to handle USB-to-serial communications.
This is the ICSP header for that microcontroller.



IORef
Reset
+3.3V
+5V
Gnd
Gnd
Vin

ADC0 GPIO14
ADC1 GPIO15
ADC2 GPIO16
ADC3 GPIO17
SDA ADC4 GPIO18
SCL ADC5 GPIO19

Comm. ADC GPIO

ICSP:

Reset	SCK	MISO
Gnd	MOSI	+5V

GPIO	ADC	Comm.	PWM	Interrupts	
GPIO18	ADC4	SDA			
GPIO19	ADC5	SCL			
	AREF				
Gnd					
GPIO13		SCK			LED
GPIO12		MISO			
GPIO11		MOSI	PWM11		
GPIO10		CS	PWM10		
GPIO9			PWM9		
GPIO8					
GPIO7					
GPIO6			PWM6		
GPIO5			PWM5		
GPIO4					
GPIO3			PWM3	INT1	
GPIO2				INT0	
GPIO1		TX			
GPIO0		RX			

- pinModeで設定をする必要はない

アナログ処理の関数

- analogReference()
- analogRead()
- analogWrite()

analogReference()

- アナログ入力に利用する参照電圧(入力電圧の最大値)を設定する。

- analogReference(mode);

mode　参照電圧の形式。具体的な値は以下の通り。

- DEFAULT

デフォルトの参照電圧。5VのArduinoボードでは5V、　3.3VのArduinoボードでは3.3V。

- INTERNAL

内蔵の参照電圧。ATmega168とATmega328では1.1V、ATmega8では2.56V。ArduinoMegaでは利用不可。

- INTERNAL1V1

内蔵の1.1Vの参照電圧(ArduinoMegaだけ)

- INTERNAL2V56

内蔵の2.56Vの参照電圧(ArduinoMegaだけ)

- EXTERNAL

AREFピンに与えられる電圧。ただし、0Vから5Vの範囲。

注意

- 参照電圧を変更後の数回は、analogRead()戻り値が不正確になることがある。

警告

- 0Vよりも低い、もしくは、5Vよりも高い参照電圧をAREFピンに入力しないこと。AREFピンの外部参照電圧を利用するときは、analogRead()を呼ぶ前に参照電圧をEXTERNALに設定すること。そうしないと、内部で発生させている参照電圧とAREFピンとがショートし、Arduinoボードのマイクロコントローラを破壊することがある。
- 5kΩの抵抗を接続して外部の参照電圧を与えると、外部の参照電圧と内部の参照電圧とを切り替えることができる。AREFピンには32kΩの内部抵抗があるため、接続した抵抗によって参照電圧が変わることに注意すること。2つの抵抗が分圧器となるため、たとえば、2.5Vを供給した時には、
2.5(V)×32÷(32+5)≒2.2VがAREFピンにかかることになる。

analogRead()

- 指定したアナログピンから値を読み取る。Arduinoボードは6チャンネル(MiniとNanoは8チャンネル、Megaは16チャンネル)、10ビットのアナログ-デジタル変換機を搭載している。これにより、0Vから5Vの入力電圧を、0から1023の整数値に変換する。読み取り精度は、5/1024Vもしくは、0.0049V（4.9mV）となる。入力電圧の範囲と精度はanalogReference()を使うことで変更できる。
- アナログ入力を読み取るのに100マイクロ秒（0.0001秒)かかる。このため、1秒間に最大10000回値を読み取ることができる。

- analogRead(pin);

- pin

読み取りに利用するアナログ入力ピンの番号。ほとんどのボードでは0から５、MiniとNanoでは、0から７、Megaでは、0から15。

0から1023までの整数を返す

注意

- アナログ入力ピンに何も接続されていないとき、analogRead()が返却する値は、種々の要因(他のアナログ入力の値、手とボードとの距離等）によって変動する。

```
int analogPin = 3;     // 可変抵抗(の中央の端子)をアナログピンの3番につなぐ
                       // 両端の端子は0Vと5Vにつなぐ
int val = 0;           // 読み取った値を格納する変数

void setup(){
  Serial.begin(9600);        //  シリアルポートを設定
}

void loop(){
  val = analogRead(analogPin);    // アナログ入力ピンを読み取る
  Serial.println(val);            // デバグ用に値を出力する
}
```

- アナログピンの番号は0から5までの数字以外に、A0からA5という変数を使うこともできます。A0は、(Arduino Unoの場合)14と定義されていますが、analogRead()の最初に、ピン番号が14以上のときは14を引くという処理が入っているので、特に問題なく動作します。

analogWrite()

- アナログ値(PWM波)をピンに書き込む。LEDの明るさやモーターのスピードを変えることができる。analogWrite()を呼び出した後、指定したデューティー比の安定した方形波(矩形波)がピンに出力される。同じピンに対してanalogWrite()もしくは、digitalRead()、digitalWrite()を発行するまで、その方形波は出力され続ける。ほとんどのピンでPWMの周波数は約490Hzである。Unoでは、5番ピンと6番ピンのPWMの周波数は約980Hzである。Leonardoでは、3番ピンと11番ピンが約980Hzである。
- ATmega168やATmega328といったほとんどのArduinoボードでは、この関数は、3、5、6、9、10、11番ピンで利用できる。Arduino Megaでは、2から13番ピンと44から46番、ATmega8を搭載した古いArduinoボードでは、9、10、11番ピンで利用できる。
- analogWrite()を呼び出す前にpinMode()を利用してピンの設定を行う必要はない。
- analogWrite()関数は、アナログピンやanalogRead()関数とは何の関連もない。

- analogWrite(pin, val);

- pin

書き込みに利用するピン

- value

0(常にオフ)から255(常にオン)の間のデューティー比

注意

- 5番ピンと6番ピンに出力されるPWM出力は、指定したデューティー比よりも高い。これは、PWM出力を生成するために利用する内部タイマを共有しているmillis()やdelay()関数との相互作用によるものである。デューティー比が低いとき(例えば0から10)に特に注意が必要で、デューティー比を0にしても、5番ピンと6番ピンでは、完全に出力を0にすることはできない。

- int ledPin = 9;　　// 9番ピンにLEDを接続
- int analogPin = 3;　// 3番ピンに可変抵抗器を接続
- int val = 0;　　// 読み取った値を格納する変数
-
- void setup(){
- pinMode(ledPin, OUTPUT);　// ピンを出力(OUTPUT)に設定
- }
-
- void loop(){
- val = analogRead(analogPin);　// 入力ピンを読む
- analogWrite(ledPin, val / 4);　// analogRead()の戻り値は0から1024、analogWrite()に与える値は0から255なので、4で割って変換する。
- }

- PWMをサポートしないピンに対して、analogWrite()を実行した場合は、valが128未満のときにはLOW、128以上のときにはHIGHが出力されます。

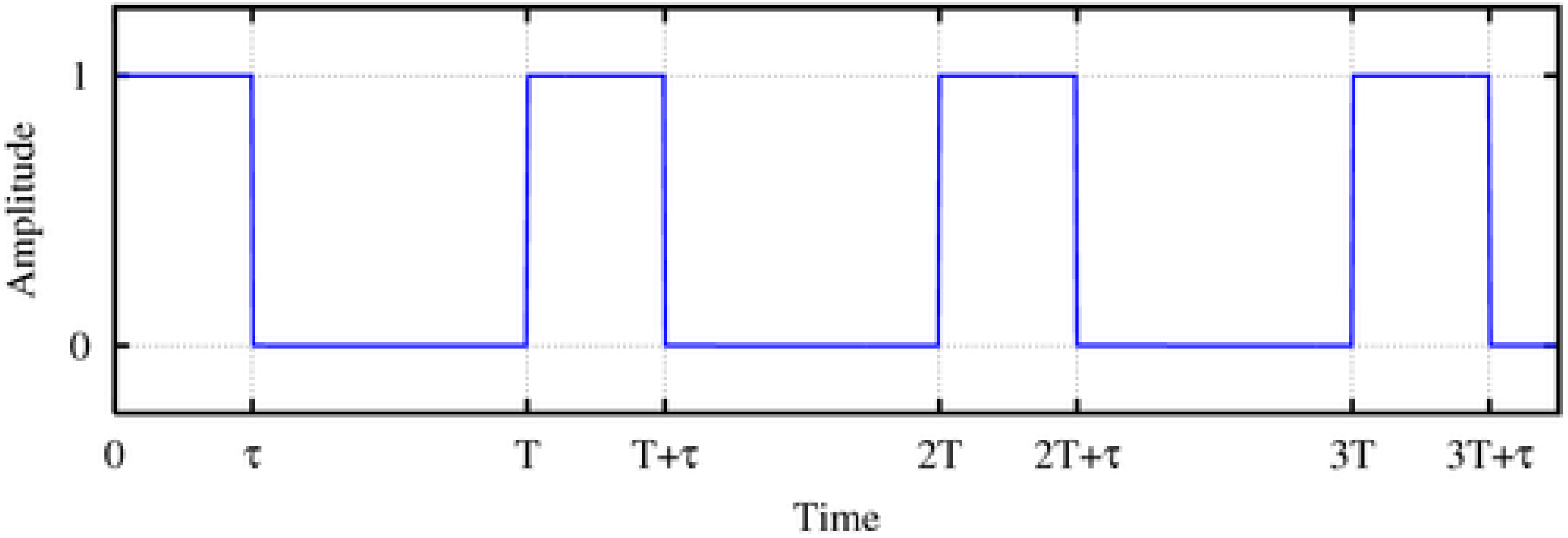
PWMとは？

- パルス幅変調（PWM）とは変調方法の一つであり、パルス波のデューティ比を変化させて変調すること。

デューティ比とは？

- 周期的な現象において、"ある期間" に占める "その期間で現象が継続される期間" の割合である。制御、電気通信や電子工学で使う。
- $D = \frac{t}{T}$
- tは、信号（関数）がゼロでない期間。
- Tは、信号（関数）の期間。周期。

- 例えば、理想的なパルス列（方形波のパルス）では、パルス幅をパルス期間（周期）で割ったものがデューティ比である。パルス幅が1μsでパルス周期が4μsの場合、デューティ比は0.25である。矩形波ではデューティ比は0.5または50%である。


Amplitude
1
0
0
τ
T
T+τ
2T
2T+τ
3T
3T+τ
Time

アナログ値を扱うために

- ディジタルと違って単純ではない。そのためにアナログ値を変数に一旦格納すると扱いやすい。

- int a;
- a = analogRead(A0);

- あとは現時点でのセンサーの値を取得して元の値との差を求めたりする。変数と演算で行った内容を使えば可能。

```
int a,b,c;
void setup() {
  Serial.begin(9600);
  a = analogRead(A0);
}

void loop() {
  b = analogRead(A0);
  c = b - a;
  Serial.println(c);
}
```

終わりに

- アナログ値を変数に格納さえできれば演算でいくらでも制御できるようになる。
- マイコンカーでもアナログ値を利用した制御はされている。
  (ポテンショメータ等)
- 次回は制御分岐を取り扱う。

課題

- 次のプログラムの動作を考えよう

```
int a,b,c;
void setup() {
  Serial.begin(9600);
  a = analogRead(A0);
}

void loop() {
  b = analogRead(A0);
  c = b - a;
  Serial.println(c);
}
```