プログラムの作法

Arduinoのプログラミング

今回学ぶもの

今回のセミナーで学ぶものは、プログラムの処理の基本である

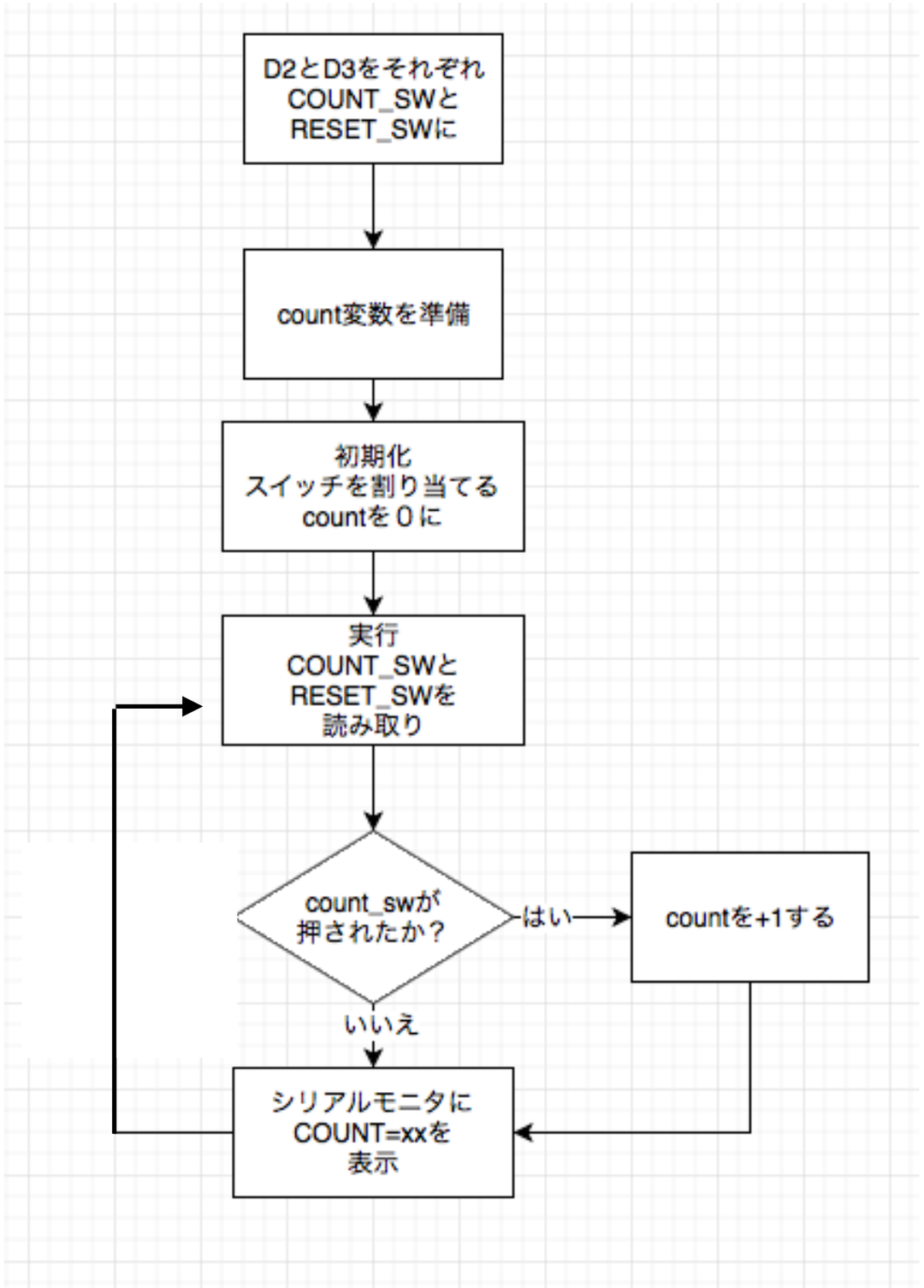
・条件分岐　if文の例を数多く学ぶ
・繰り返し処理　for文の具体例を学ぶ
・繰り返し処理　while文の具体例を学ぶ

１．カウントをするプログラムを作ってみよう

Ｄ２にプッシュスイッチ（カウント用）Ｄ３にタッチスイッチ（リセット用）を ＧＲＯＶＥに繋いでfig-1を開く

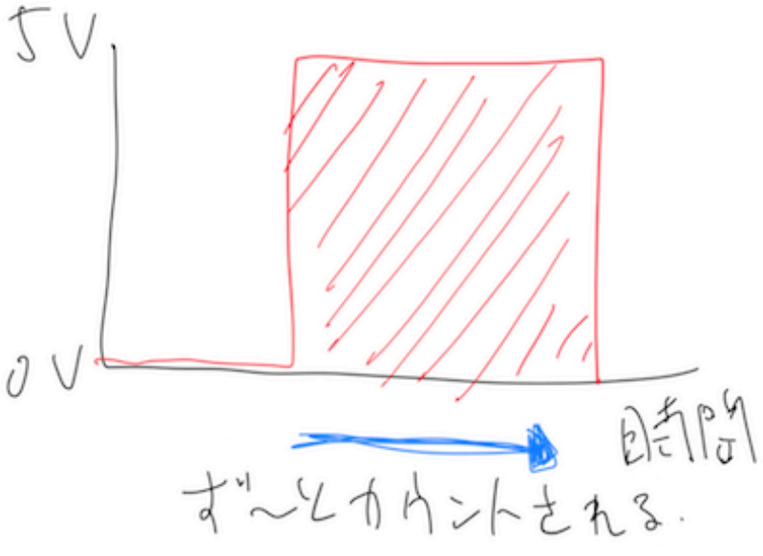
fig-1

```
const int COUNT_SW=2;
const int RESET_SW=3;

int count;

void setup() {
  Serial.begin(9600);
  pinMode(COUNT_SW,INPUT_PULLUP);
  pinMode(RESET_SW,INPUT_PULLUP);

  count=0;
}

void loop() {
   int count_sw=digitalRead(COUNT_SW);
   int reset_sw=digitalRead(RESET_SW);
   if (count_sw==HIGH) {
       count=count+1;
   }
   Serial.print("COUNT=");
   Serial.println(count);
}
```



うまく動かない理由は？

Ｄ２を少し押しただけでもマイコンは高速に動作しているので、とても長い間押した状態と同じになる。



うまく動かすためには？

ボタンを押す動作の中で「一瞬しか現れない」現象をカウントの条件にする。
（例）スイッチを押した状態から指を離した瞬間

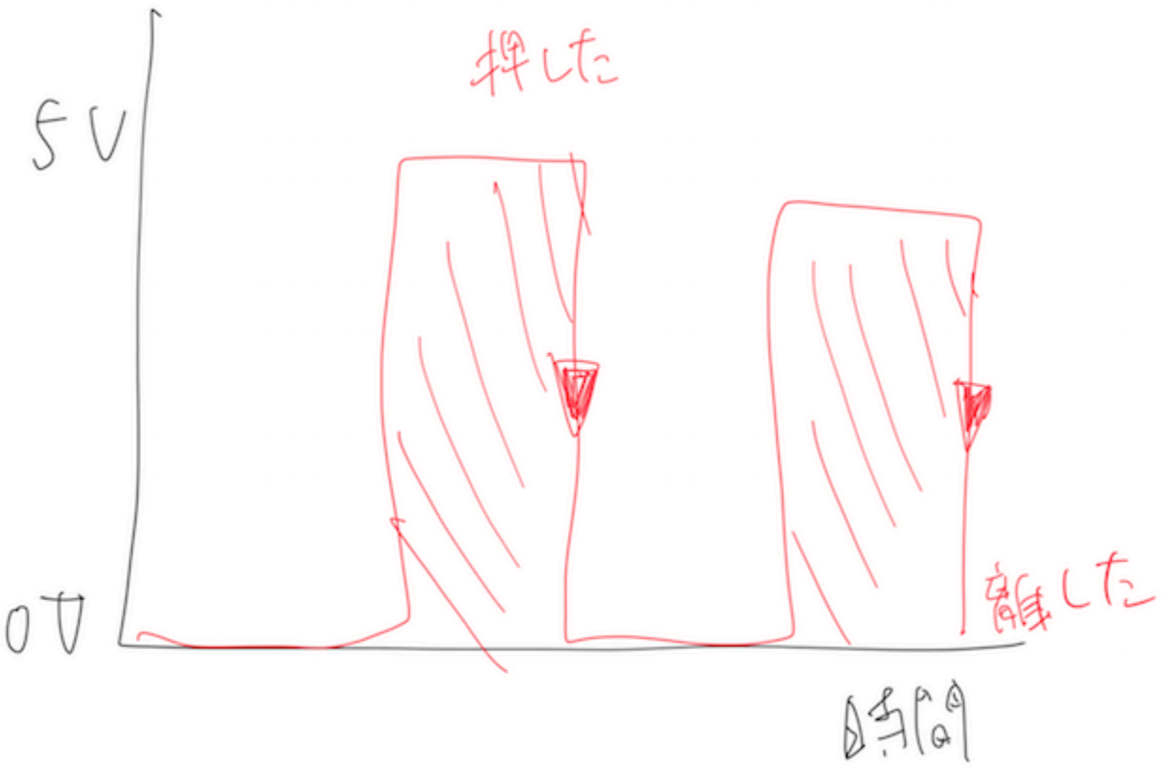


fig-1a

```
const int COUNT_SW=2;
const int RESET_SW=3;

int count;
int sw_status;

void setup() {
  Serial.begin(9600);
  pinMode(COUNT_SW,INPUT_PULLUP);
  pinMode(RESET_SW,INPUT_PULLUP);

  count=0;
}

void loop() {
    int count_sw=digitalRead(COUNT_SW);
    int reset_sw=digitalRead(RESET_SW);


    if (count_sw==LOW && sw_status==HIGH) {
        count=count+1;
    }
    Serial.print("COUNT=");
    Serial.print(count);
    Serial.print(",SW_STATUS=");
    Serial.println(sw_status);
    sw_status=count_sw;
}
```

sw_statusの宣言の場所はどうして
setup()やloop()のところじゃないのかな？

かんがえてみよう！

条件分岐　if文で複数の条件を扱う時

「A && B」　は Aの条件が成立し、なおかつBの条件も成立する時　「アンド」という。
「A || B」　　は Aの条件が成立したか、もしくはBの条件が成立した時　「オア」という。

先ほどのプログラムの場合、

if (count_sw==LOW && sw_status==HIGH){ なので

count_swが離された状態でなおかつsw_statusが前回押された状態ならばになる。
（スイッチから指を離した瞬間にカウントする）

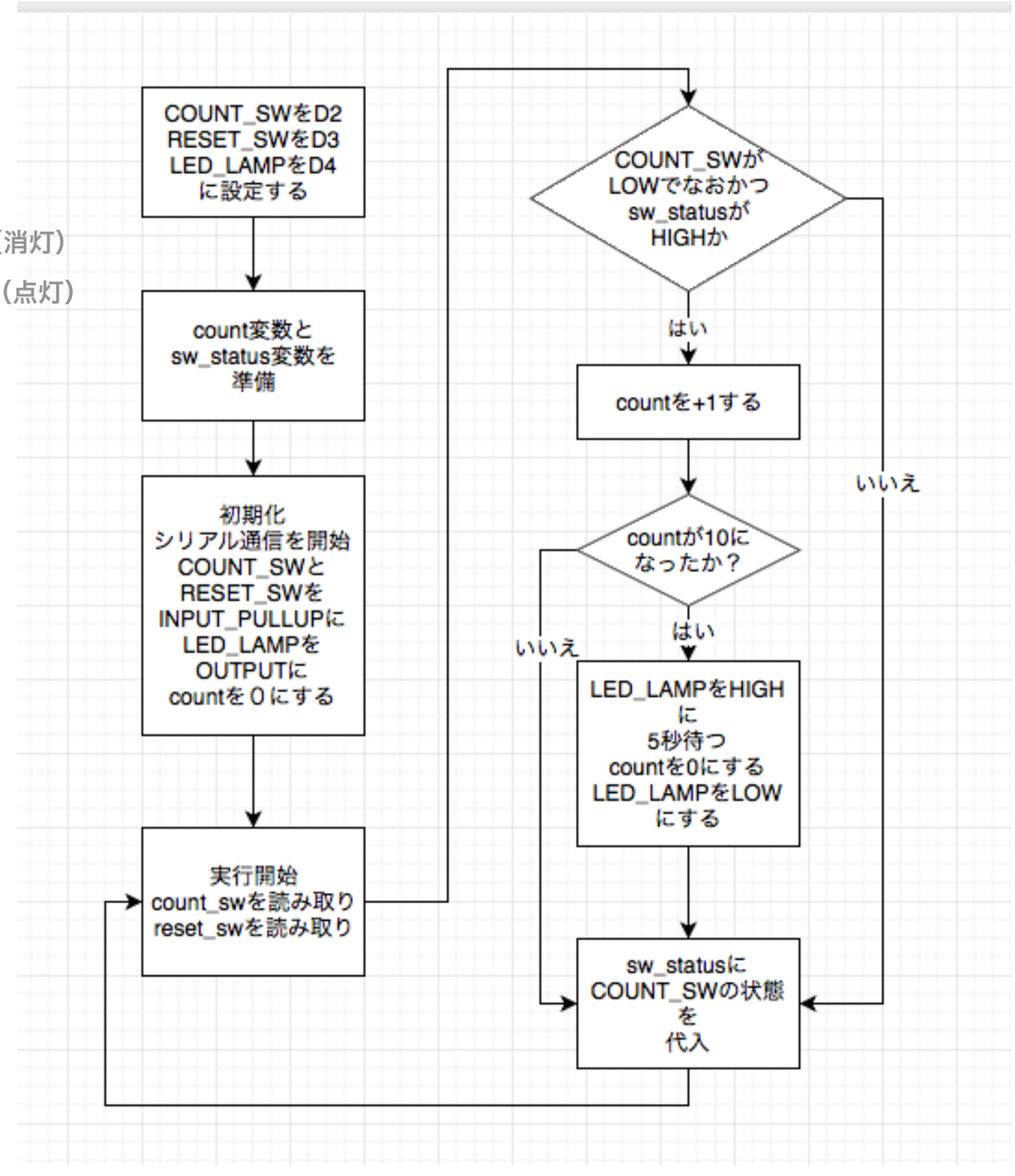
理解できた人から順に、RESET_SWを押したらcountを０にする処理を作ってみよう。
それが終わったら、スイッチを押した瞬間にカウントするプログラムに変更してみよう。

条件分岐のまとめ

スイッチを１０回押したら、Ｄ４に繋いだＬＥＤが5秒点灯するプログラムを作ってみよう。

作る時のヒント

```
const int LED_LAMP=4; //LEDランプはＤ４に
pinMode(LED_LAMP,OUTPUT); //D4を出力に
digitalWrite(LED_LAMP,LOW); //D4の出力をLOWに（消灯）
digitalWrite(LED_LAMP,HIGH); //D4の出力をHIGHに（点灯）
delay(5000);//５秒(5000ミリ秒）待つ
```



2．繰り返し処理をするプログラムを書いてみよう

A0にサウンドセンサをＧＲＯＶＥに繋いでfig-3を開く

fig-3

```
const int SOUND_SENSOR=A0;//A0から入力


void setup() {
   Serial.begin(9600);
}

void loop() {
   int value=analogRead(SOUND_SENSOR);
   Serial.println(value);
   delay(1000);
}
```

シリアルモニタで見てみると・・・

周りの雑音を拾うので、読み取り値が安定しない。
手などを叩いて確認してみても安定しない。

変動の大きいセンサ値は **「平均をとる」** と有効な値を取りやすくなる。

平均を求めるってどうするの？

例えば、シリアルモニタで読み取った数値が以下の場合・・・

20+0+5+0+0+2+2+8+8+0+2+6+8+1＝62

62を読み取った個数14で割ると4.4
という数値になる。

ばらつく数値も平均を取ると扱いやすくなる。

読み取った個数が少ない状態で平均を取ると・・・早いけどばらつきが多い。
読み取った個数が多い状態で平均を取ると・・・・遅くなるけどばらつきが少ない。
（ただし、あまり個数が多いと必要な変化も埋もれてしまうので注意）

for文を使って平均化してみよう

fig-3a §

```
const int SOUND_SENSOR=A0;//A0から入力

void setup() {
   Serial.begin(9600);
}

void loop()
  int value=0;
   for(int i=0;i<100;i++) {
      value=value+analogRead(SOUND_SENSOR);
   }
   Serial.println(value/100);
}
```

ここで読み取り値を１００回読み取って全て足して合計を求める。

合計を１００で割れば平均値が求められる。

fig-3

```
const int SOUND_SENSOR=A0;//A0から入力


void setup() {
   Serial.begin(9600);
}

void loop() {
   int value=analogRead(SOUND_SENSOR);
   Serial.println(value);
   delay(1000);
}
```

実際にシリアルモニタで数値を見てみよう。

そのあとにi<100をi<3とかi<200にして結果を見てみよう！

(自動スクロールのチェックをオフにすると表示を止められます)

【応用】if文を使って最大値と最小値を取得してみよう

fig-3b

```
const int SOUND_SENSOR=A0;//A0から入力
unsigned int min_value,max_value;//最小と最大値を格納する変数を準備

void setup() {
   Serial.begin(9600);
   min_value=65535;//最小値には扱える一番大きい数値を
   max_value=0;//最大値には扱える一番小さい数値で初期化
}

void loop() {
  unsigned int value=0;
   for(int i=0;i<100;i++) {
      value=value+analogRead(SOUND_SENSOR);
   }
   value=value/100;
   if (value<min_value) min_value=value;//min_valueより小さい値がきたら更新
   if (value>max_value) max_value=value;//max_valueより大きい値がきたら更新

   Serial.print("value=");
   Serial.print(value);
   Serial.print(",min_value=");
   Serial.print(min_value);
   Serial.print(",max_value=");
   Serial.println(max_value);
}
```

このプログラムを改造して、max_valueの半分の値を超えたら、Serial.printlnを使って”DETECT!!!”と表示するように改造してみよう。

while文はどんな時に使う？

for文は「あらかじめ繰り返す回数がわかっている処理の時」に使う。
while文は「ある条件を満たすまで繰り返したい処理の時」に使う。

fig-4

```
const int HUMAN_SENSOR = 2;
unsigned long detect_time;//0~86400秒（24Hをカウントできるように）

void setup() {
   Serial.begin(9600);
   detect_time=0;
   Serial.println("keibi wo kaishi simasu.");
}

void loop() {
  int human_detect=LOW;
  while (! human_detect){
    delay(1000);
    detect_time=detect_time+1;
    human_detect=digitalRead(HUMAN_SENSOR);
  }
  Serial.println("shinnyuusya ari!!");
  Serial.print("DETECT TIME=");
  Serial.println(detect_time);

}
```

human_detectがHIGHになるまで
１秒ごとにdetect_timeをカウントアップ

24時間、警備を開始してから侵入者が入った時の秒数を表示するプログラム
D2にタッチセンサを繋いで実行してみよう！

while文の条件の「！」って？？

fig-4

```
const int HUMAN_SENSOR = 2;
unsigned long detect_time;//0~86400秒（24Hをカウントできるように）

void setup() {
   Serial.begin(9600);
   detect_time=0;
   Serial.println("keibi wo kaishi simasu.");
}

void loop() {
  int human_detect=LOW;
  while (! human_detect){
    delay(1000);
    detect_time=detect_time+1;
    human_detect=digitalRead(HUMAN_SENSOR);
  }
  Serial.println("shinnyuusya ari!!");
  Serial.print("DETECT TIME=");
  Serial.println(detect_time);

}
```

while(! human_detect)の「!」は、
「NOT」を表します。

マイコンは条件が成立すると「１」成立しない場合は「０」で判断しています。

whileは条件が成立している間だけ実行になるので、human_detectがLOW=0の場合に条件を成立させたいときは「NOT」の「！」を付けるのです。

「! human_dectect」ではなく
「human_detect==LOW」でも同じ結果になります。

今回のセミナーは以上になります。