

資料：slide11とsketch11の修正について

川上 博

0. 11回目の資料や説明で幾つかの初歩的記載ミスや検討不足の事柄がありました。これらの事柄に関して色々教えてくださった辻先生、要点を的確にご指摘下さった加村氏に感謝致します。

すぐに修正できる事柄は修正しましたが、道幅の広さと走行条件の与え方など未だ検討不足です。従って、このメモは修正箇所の指摘と問題点の指摘に過ぎず、問題点の解答にはなっていないことをお許しください。なお、現在アップロードされている資料は修正後のものです。

1. サーボモータのパラメータ

servoL.attach(pin, min_pw, max_pw)などの第2引数と第3引数の値

```
int centerR=1510, centerL=1541, pw=700;

void setup() {
  delay(1000);
  servoL.attach(servoL_Pin, centerL-pw, centerL+pw);
  servoR.attach(servoR_Pin, centerR-pw, centerR+pw);
  Serial.begin(9600);
}
```

centerR, centerLからどれだけ+-するかを与える変数 pwの値を800から700に変更しました。辻先生に教えて頂いたGWS社の資料：

<http://www.gws.com.tw/english/product/servo/servo.htm>

によると

サーボに与える信号のパルス幅は、0.8ms(800us)から2.2ms(2200us)の範囲
0.7ms - 0.8ms, 2.2ms - 2.3msの1msは安全のために確保された範囲

ということです。なお、私が使っていた値は

<http://arduino.cc/en/Reference/ServoAttach>

にある説明を読んで私が適当に選んだ値でした。そのうちに変えれば良いと思って結局変えてなかった値です。まったく申し訳ありません。辻先生の値に合わせさせて頂きました。

2. 関数：fwd(), bwd(), rot_CW(), rot_CCW(), stp()の引数tauの削除

```
int spd=20;

void fwd(){
  servoL.write(90-spd);
  servoR.write(90+spd);
}
```

のように修正しました。遅れが必要な時はloop()の中で書くことにしました。

なお、サーボに与えるパルス信号の周期は20msということなので、これまでのスケッチでdelay(10);などとして与えていましたが、あまり意味のないことでした。この変更でturtle graphicsで使用した同名の関数とは内容が違うことになります。注意してください。

3. 黒線を検出するセンサー：フォト・レフレクターのパラメータ

3つのフォト・レフレクターの寸法は右の図1のように設計されています。図の上が前方、下が後方と考え、左のフォト・レフレクター(L)、中央のそれ(C)、右のフォト・レフレクター(R)の値はそれぞれ、val_L, val_C, val_Rに0から1023の整数値で入力されます。

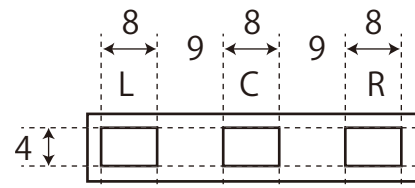


図1 フォト・レフレクターの寸法。

簡単のため、これを適当な閾値(th)で2値化して使うことにします。たとえばth=200として、値がth以下の場合が白地、値がth以上を黒地とみなすことにします。

4. 道幅と走行条件について

大雑把に考えて、黒い曲線を追尾する場合、前進する、右折する、左折するの最低3つの走行が必要です。これらの関数は、3つのフォト・レフレクターの値val_L, val_C, val_Rが適当な条件式を満たす場合に実行され、JJ2を曲線に沿って動かせることとなります。そこで

1. 道幅（黒い曲線の幅）,
2. フォト・レフレクターの寸法

から、関数「前進する、右折する、左折する」の条件式を求めることがここでの問題です。

実は、前回の講座の開講日(2014/11/08)前日まで私は黒色のビニールテープ（幅19mm）で作った道を使っていました。講義のとき9mmの道幅を使うことになり、何か勝手に違うことに戸惑いました。結果からすると、辻先生の設計されたJJ2は、当初から曲線追尾の道幅は9mm程度を考えるとということが仕様にあったのではないかと思います。一方、私はインクジェット・プリンターで出力した道はフォト・レフレクターが感知してくれないこともあり、あり合わせのテープを使って道を作っていました。

以下、道幅が9mm程度の場合、20mm程度の場合、道幅が可変な場合、白地と黒地の境界線の場合に、直進を中心とした運動を想定した条件を問題にすることにしましょう。すなわち、道と3つのフォト・レフレクターの並びの直線が直交している状態を想定します。これは最も簡単な場合です。それでも扱いが十分でないかも知れません。問題提起であるとお考えください。

4.1 道幅が9mm程度の場合：Example1101Abis

道幅が9mmまでの道として、道幅を固定した場合は、3つのフォト・レフレクターは同時に黒色を検出することはないので、それぞれが黒色を検出した場合だけを考えればよい。

```
int th=200, val_L, val_C, val_R;

void loop() {
  val_L=analogRead(A0);
  val_C=analogRead(A1);
  val_R=analogRead(A2);
  // prtData(val_L, val_C, val_R);
  if(val_L>th && val_C>th && val_R>th){
    stp();
  }else if(val_C>th){
```

```

    fwd();
}else if(val_R>th){
    rot_CW();
}else if(val_L>th){
    rot_CCW();
}else{
    fwd();
}
// delay(40);
}

```

4.2 道幅が20mm程度の場合：Example1101Ater

この時は黒い道幅が広がるので、直進は $val_C > th$ （道の中心線とセンサーの中心線が一致した位置より、 $10\text{mm} + 4\text{mm} = 14\text{mm}$ 左右に動かすと条件が破れる）よりも両脇のセンサー値を使った $val_L < th \ \&\& \ val_R < th$ （ $21\text{mm} - 10\text{mm} = 11\text{mm}$ 動かすと条件が破れる）の方がわずかではあるが早く変化を検出できる。これは道幅が広がるほど差が大きくなる。

```

void loop() {
    val_L=analogRead(A0);
    val_C=analogRead(A1);
    val_R=analogRead(A2);
    // prtData(val_L, val_C, val_R);
    if(val_L>th && val_C>th && val_R>th){
        stp();
    }else if(val_L<th && val_R<th){
        fwd();
    }else if(val_L<th && val_R>th){
        rot_CW();
    }else if(val_L>th && val_R<th){
        rot_CCW();
    }else{
        // fwd();
    }
    // delay(40);
}

```

4.3 道幅が20数ミリ程度まで（幅が場所によって異なる幅凸凹も含めて）：Example1101A

色々考えると面白そうなのですが、ここでは3つのセンサーの満たす条件を丁寧に書いてそれぞれの動きをしてもらうことにします。思考停止で申し訳ありません。

```

void loop() {
    val_L=analogRead(A0);
    val_C=analogRead(A1);
    val_R=analogRead(A2);
    // prtData(val_L, val_C, val_R);
    if(val_L>th && val_C>th && val_R>th){
        stp();
    }else if(val_L<th && val_C>th && val_R<th){
        fwd();
    }else if(val_L<th && val_C<th && val_R>th){
        rot_CW();
    }else if(val_L>th && val_C<th && val_R<th){

```

```

    rot_CCW();
  }else{
    fwd();
  }
  // delay(40);
}

```

4.4 白地と黒地の境界線を追尾する場合

ちゃんと考えていませんが、3個あるセンサーの隣あわせの中間線のどちらを境界線に合わせるかで条件が変わります。例えば、左のセンサーと中央のセンサーの間の変化をみて制御するとすれば次のようになるでしょう。

```

void loop() {
  val_L=analogRead(A0);
  val_C=analogRead(A1);
  val_R=analogRead(A2);
  // prtData(val_L, val_C, val_R);
  if(val_L<th && val_C>th && val_R>th){
    fwd();
  }else if(val_L<th && val_C<th && val_R>th){
    rot_CW();
  }else if(val_L>th && val_C>th && val_R>th){
    rot_CCW();
  }
  // delay(40);
}

```

4.5 デフォルト値

以上のスケッチでは、次の事柄が暗黙の了承として組み込まれたものがあります。

1. JJ2を持ち上げると停止状態になる。これは、持ち上げるとセンサー値は黒となるので、このことを利用しました。
2. 白地に置くと前進する。これはしばしば問題を起こします。例えば、4.2のスケッチを9mmの道に適用すると、左折動作または右折動作の間に直進動作が入って脱線することがあるかも知れません。4.1あるいは4.3を使うなど、適切に対処をお願いします。
3. delay(40);などをloop()に含ませたこと。これは、どの程度にすれば脱線するかをテストするために入れました。講義で使った道ではdelay(100);過ぎから脱線を始めます。適当に消去してください。

5. スライド：slide11の修正

気づいた点は修正しました。修正後のスライドをアップロードしてあります。