

実験6 -(1)

RA2スイッチを押すと両輪が回転し、RA4を押すと停止するプログラム
両輪の回転数をなるべくそろえる。

```
#include<pic.h>

void main(){

    ADCON1 = 0b100; //A/Dポート構成コントロール設定
    TRISA = 0b111111; //PORTAを入力側に切り替え

    while (1) { //ループ開始
        if(RA2 == 0){ //RA2が押されたら…
            while (RA2 == 0) {} //RA2が離されるのを待つて

            TRISC = 0b10000000; //RC1を出力に

            CCP2CON = 0b1100; //CCP2をPWMモードに設定
            CCP1CON = 0b1100; //CCP1をPWMモードに設定

            PR2 = 0xff; //タイマ2のカウンタと比較する値
            T2CON = 0b110; //タイマ2を分周比1:16でオン

            CCPR2L =160; //左輪のデューティサイクル設定
            CCPR1L =120; //右輪のデューティサイクル設定
        }

        if(RA4 == 0){ //RA4が押されたら
            while (RA4 == 0) {} //RA4が離されるのを待つて
            TRISC = 0b00000000; //停止処理
            CCP2CON = 0b0000;
            CCP1CON = 0b0000;
        }
    }
}
```

結果

ほぼ期待通りに動作した。

フローチャート

page()に示す

考察

両輪の動きを合わせるためのおおよその比が求まった。

実験6-(2)

RA2スイッチを押すと2秒間で徐々に加速、3秒等速、2秒減速するプログラムの制作

```
void accel(int time){
    int a;
    for(a = 0; a < time; a++){
        CCPR2L++;
        if(a%4>0){CCPR1L++;} //デューティサイクル調整のため4回に一回スルー
        flat(1);
    }
}
void flat(int time){ //TOIFがたまるまで同じ速度で移動 13.1ms
    int b;
    for(b = 0; b<time; b++){
        while (1) {
            if(TOIF == 1){
                break;
            }
        }
        TOIF = 0;
    }
}
void brake(int time){
    int c;
    for(c = 0; c < time; c++){
        CCPR1L--;
        if(c%4>0){CCPR2L—;}
        flat(1);
    }
}
```

```

void move(int acceTime, int evalTime){
    TRISC = 0b10000000; //6-1 と同等の初期設定
    CCP2CON = 0b1100;
    CCP1CON = 0b1100;
    T2CON = 0b110;
    PR2 = 0xff;
    CCPR2L = 0;
    CCPR1L = 0;
    accel(acceTime);
    flat(evalTime);
    brake(acceTime);
    CCPR1L = 0;
    CCPR2L = 0;
}

void main(){
    TRISA = 0b11011111;
    OPTION = 0b11010111; //タイマ0に割り当て分周比1:256 1 サイクル13.1ms
    ADCON1 = 0b100; //6-1 と同等の初期設定
    TRISA = 0b1111111;
    while (1) {
        if(RA2 == 0){
            while (RA2 == 0) {}
            move(152,228); //2 s = 13.1ms × 152
        }
    }
}

```

結果

初動と停止前の挙動が少し期待とは外れ
カーブするようになってしまった

フローチャート

page()に示す

考察

実験 1 で調整したデューティサイクル比を参考にインクリメントを

調整したものの、結果は上記のようになった。

原因としては

- ・ギアの噛み合わせ
- ・タイヤの摩擦力の問題
- ・モーターの性質

など、今回では機械的要因が大きいと考えられる。

実験7

光の強い方へ進むプログラムを作成

```
#include<pic.h>
//ホールド用コンデンサの充電待ち関数
//25.6 $\mu$ s > 19.72 $\mu$ sより充電完了する
void charge(){
    OPTION=0b11011000;           //PTION_REGレジスタ 比1:1
    TOIF=0;
    TMR0=0b10000000;             //タイマ0を25.6 $\mu$ sの状態にする
    while(TOIF != 1){}
}

void main(){
    int right,left;
    double rate = 160/120;
    TRISA = 0b11011111; //RA0,RA1,RA3を inputに設定
    TRISC = 0b10000000; //RC1,RC2を outputに設定

    //PWMmode
    CCP2CON = 0b1100;
    CCP1CON = 0b1100;

    //timer initialize
    T2CON = 0b110;
    PR2 = 0xff;
    CCPR2L = 0;
    CCPR1L = 0;
```

```

while (1) {
    //A/D結果をADRESHの7-0bit:ADRESLの7-6bitへ 下位2bit無視
    ADCON1 = 0b00000100;

    //right
    ADCON0 = 0b11000001; //RA0 強ければ右輪を強く
    charge();
    ADGO = 1; //A/D変換開始
    while (ADGO != 0) {}; //変換中
    if(ADRESH > 0b00111100){ //60以上の時 上限を60に設定
        right = 60;
    }else{
        right = ADRESH;
    }
    CCPR1L = right;

    //left
    ADCON0 = 0b11001001; //RA1 強ければ左輪を強く
    charge();
    ADGO = 1; //A/D変換開始
    while (ADGO != 0) {}; //変換中
    if(ADRESH > 0b00111100){ //上限80
        left = 60 * rate;
    }else{
        left = ADRESH * rate;
    }
    CCPR2L = left;
}
}

```

結果

光の強い方に進んだ。

フローチャート

page()に示す

考察

デューティサイクルの上限値があったため、
コード上で制御する必要があった。

コンデンサが十分に充電されていないと
A/D変換がうまくされず挙動がおかしくなったので
確認すべきであった。

コンデンサの充電に必要な時間以上の
サイクル停止時間を設けることにより充電を行っている。