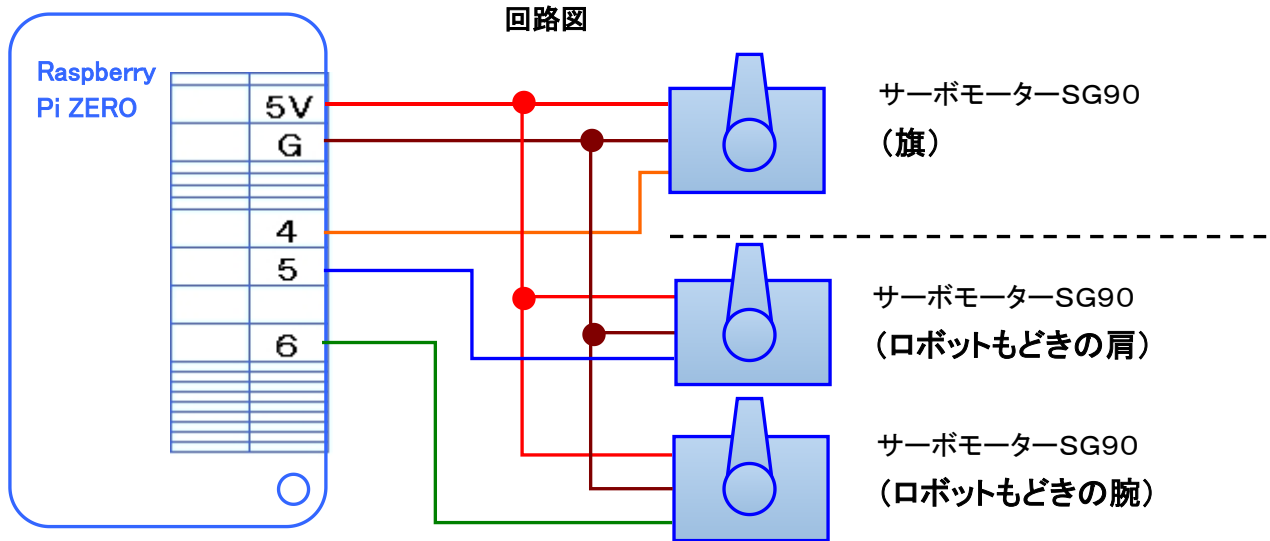


PWM とモーターでロボットもどき

(PWM :Pulse Width Modulation パルス幅制御)



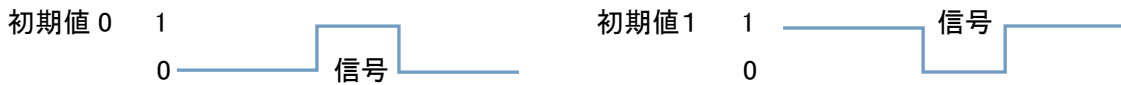
```
#include<softPwm.h>
```

ソフトウェアPWMを使用するためのライブラリを読み込む。

・ Wiring Pi の関数(その6)

softPwmCreate(ピン番号, 初期値, 周期) ソフトウェアPMWのセットアップ

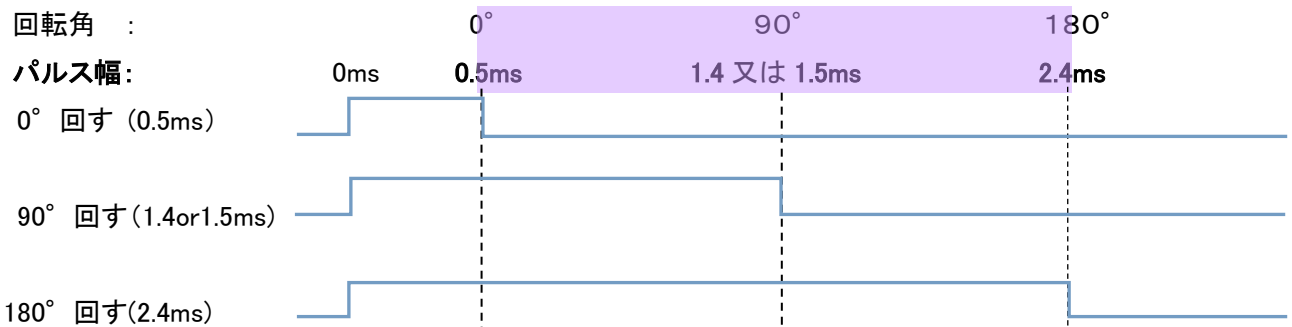
初期値: 信号を送っていない時0か1か。 周期: 0.1ms(100μs)単位 いずれもサーボモーターの仕様による。



softPwmWrite (ピン番号,パルス幅) ピン番号のピンからパルス幅の PMW 信号を出力する。

パルス幅: 0.1ms(100μs)単位。どの位置にサーボモータを回すかの情報を与える。

例:サーボモーター SG90 の仕様は、周期20ms、パルス幅0.5~2.4ms 回転角は 0° ~180° なので、



注)パルス幅は、あくまで「サーボモータに取るべき位置を知らせる為の情報(パルスの幅が角度を示す)」であって、このパルスが出ている間サーボモーターが回る、というような機械的な信号ではありません。

サーボモータで旗の上げ下ろしをするプログラム motor.c

```
#include<wiringPi.h> /* Wiring Pi のライブラリを読み込む */
#include<softPwm.h> /* ソフトウェア PWM のライブラリを読み込む */
int main()
{
    wiringPiSetup(); /* WiringPi の初期化 */
    softPwmCreate(4,0,200); /* 4 番ピンを PMW 出力に 初期値は 0 周期 20ms */
    softPwmWrite(4,15); /* 4 番ピンから 1.5ms パルス出力(位置を 90° に) */
    delay(500); /* 500ms 動作を待つ */
    softPwmWrite(4,5); /* 4 番ピンから 0.5ms パルス出力(位置を 0° に) */
    delay(500); /* 500ms 動作を待つ */
    return 0;
}
```

SG90は#3で紹介した「ハック ラズベリーパイ Raspberry Pi 電子工作入門キット」に1個入っているサーボモーターですが、安価なため、いろいろなキットについています。

ここでは、Amazon の「SG90 サーボ用 2 軸カメラマウント(マウント+SG90 2個)」¥1,049-

http://www.amazon.co.jp/gp/product/B010BWNZO4?ie=UTF8&tag=hiro345-22&link_code=wjl

を使って、ロボットもどきを作り、動かしてみましよう。(コマンドは次ページを参照)



```
##include<stdio.h>
#include<stdlib.h>
#include<wiringPi.h>          /* Wiring Pi のライブラリを読み込む */
#include<softPwm.h>          /* ソフトウェア PWM のライブラリを読み込む */
int main()
{
    char cmd[5];
    wiringPiSetup();          /* WiringPi の初期化 */
    softPwmCreate(4,0,200);   /* 4 番ピンを PMW 出力に 初期値は 0 周期 20ms */
    softPwmCreate(5,0,200);   /* 5 番ピンを PMW 出力に 初期値は 0 周期 20ms */
    softPwmCreate(6,0,200);   /* 6 番ピンを PMW 出力に 初期値は 0 周期 20ms */
    printf("command = "); scanf("%s",cmd);          /* コマンド入力 */
    while (cmd[0] != 'e')    /* 'e' は eND 終了コマンド */
    {
        if(cmd[0] == 'l' || cmd[1] == 'l') /* 'l'は IO 「肩を下に」 */
            softPwmWrite(5,6); /* 5 番ピンは肩 0.6ms は、ほぼ 0 度 */
        if(cmd[0] == 'm' || cmd[1] == 'm') /* 'm'は mIDDLE 「肩を水平」 */
            softPwmWrite(5,14); /* 5 番ピンは肩 1.4ms は、ほぼ 90 度 */
        if(cmd[0] == 't' || cmd[1] == 't') /* 't'は tOP 「肩を上」 */
            softPwmWrite(5,24); /* 5 番ピンは肩 2.4ms は、ほぼ 180 度 */
        if(cmd[0] == 'u' || cmd[1] == 'u') /* 'u'は uP 「腕を上」 */
            softPwmWrite(6,5); /* 6 番ピンは腕 0.5ms は、ほぼ 0 度 */
        if(cmd[0] == 'd' || cmd[1] == 'd') /* 'd'は dOWN 「腕を下」 */
            softPwmWrite(6,15); /* 6 番ピンは腕 1.5ms は、ほぼ 90 度 */
        delay(500); /* 500ms 動作を待つ */
        if(cmd[0] == 'a') /* 'a'は aUTO デモを自動実行 */
        {
            softPwmWrite(5,6); delay(500); /* 「肩を下に」初期状態 */
            softPwmWrite(6,15); delay(500); /* 「腕を下に」初期状態 */
            softPwmWrite(5,14); delay(500); /* 「肩を水平」腕は前へ */
            softPwmWrite(5,6); delay(500); /* 「肩を下に」腕が下に */
            softPwmWrite(5,14); delay(500); /* 「肩を水平」腕は前に */
            softPwmWrite(6,5); delay(500); /* 「腕を上」腕は横に */
            softPwmWrite(5,24); delay(500); /* 「肩を上」腕は回転 */
            softPwmWrite(6,15); delay(500); /* 「腕を下」腕は上に */
            softPwmWrite(5,6); delay(500); /* 「肩を下に」初期状態 */
        }
        printf("command = "); scanf("%s",cmd); /* コマンド入力 */
    }
    softPwmWrite(5,6); delay(500); /* 「肩を下に」初期状態 */
    softPwmWrite(6,15); delay(500); /* 「腕を下に」初期状態 */

    return 0;
}
```

参考記事:「RaspberryPi と WiringPi でサーボを動かす」

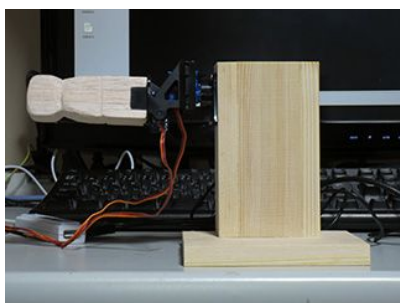
<http://qiita.com/locatw/items/f15fd9df40153bbb4d27> (SG90の仕様のみ参考にしました。)

上記記事に書かれているのはハードウェア PWM ですが、今回使ったのはソフトウェア PWM です)

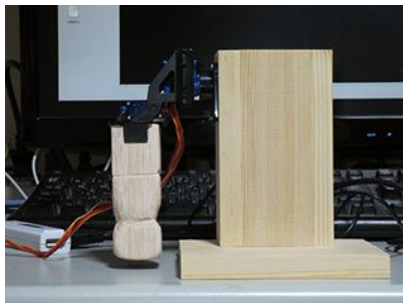
参考資料:トランジスタ技術 2016 年 3 月号別冊付録「全ラズパイ対応! 楽々I/O関数セット WiringPi 虎の巻」

ロボットもどきのコマンド(command=)

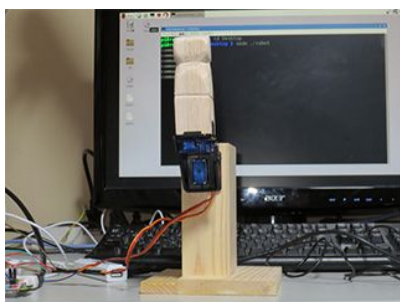
腕を上げる 'u' uP



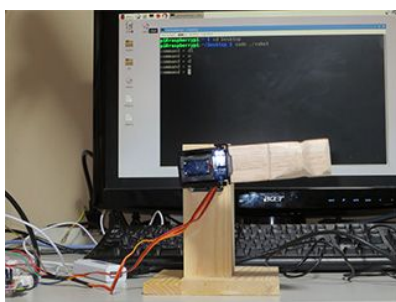
腕を下げる 'd' dOWN



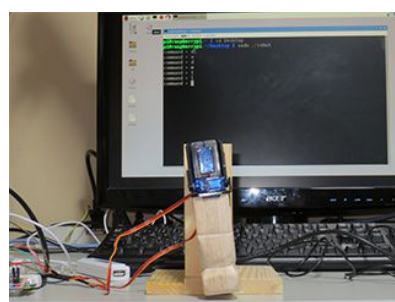
肩を上に戻す 't' tOP



肩を水平に 'm' mIDDLE

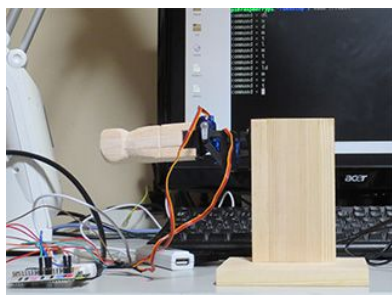


肩を下に戻す 'l' lO

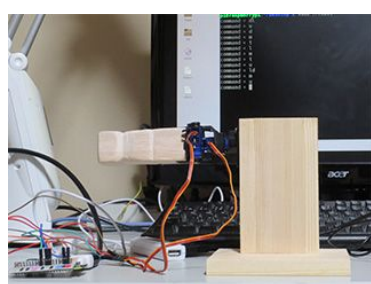


組み合わせ

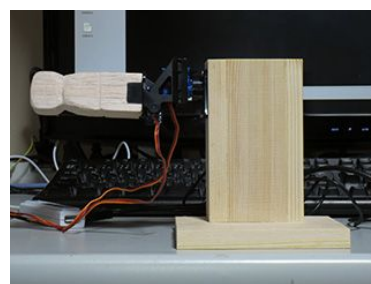
① tu



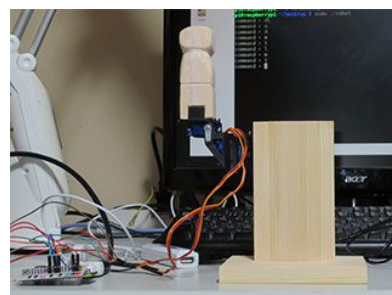
② mu



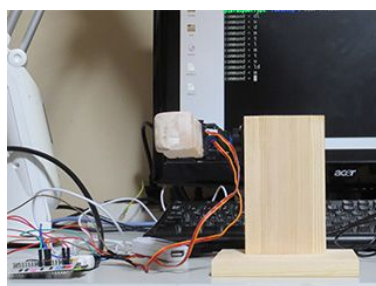
③ lu



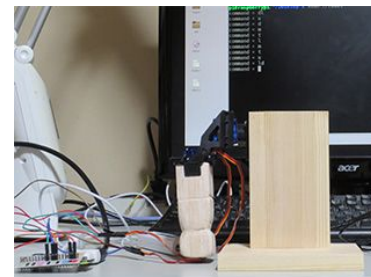
④ td



⑤ md



⑥ ld



デモ(自動実行) 'a' aUTO

上記組み合わせで言うと⑥→⑤→⑥→⑤→②→①→④→⑥の順に実行