



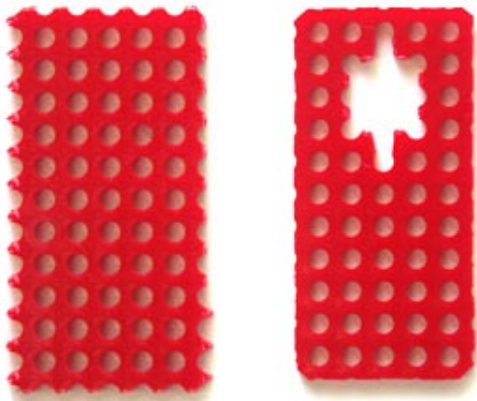
**ROBO DESIGNER**  
**Application book**

**ラジコン用サーボを使った4足ロボット**



## 足を作る (ユニバーサルプレートとサーボホーンの加工)

カットしたユニバーサルプレートをサーボの出力軸の根元が取り付けられるように加工します。  
サーボに付属のサーボホーンの穴を 5mm ピッチ / 3mm  $\phi$  穴に合うように加工します。

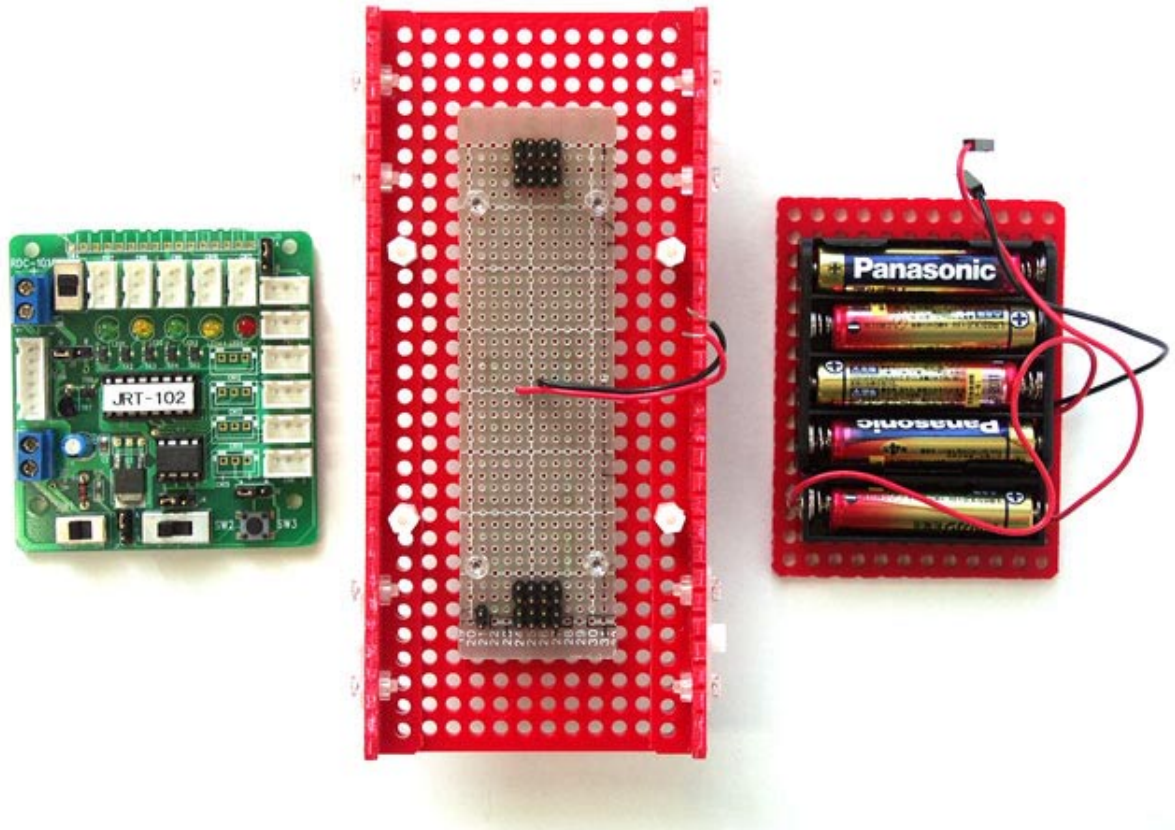




## 組み立て：胴体にコントローラーボードと電源を固定する

胴体のユニバーサルプレート、足の付け根のL字プレート、コントローラーボードは胴体上部、電池ボックスは胴体下部にスペーサで固定します。

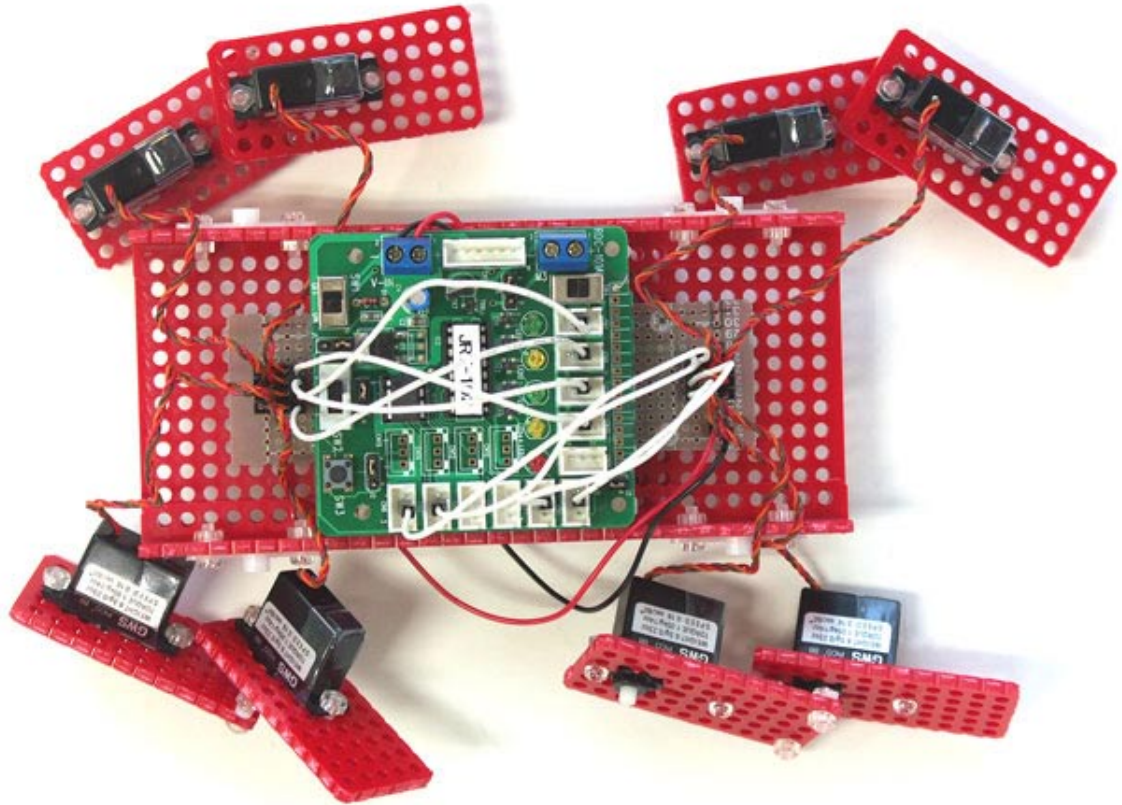
多数のサーボを同時に使用するので、たくさんの電流が必要になります。コントローラーボードのコネクタからの電流では足りません。アルカリ電池4本でも6Vですが、すぐに4.8V以下になってしまうので、単四型ニッケル水素充電電池5本で駆動するのが理想的です。





コントローラーボード、電源、サーボを配線します。サーボホーンのネジはまだ固定しません。

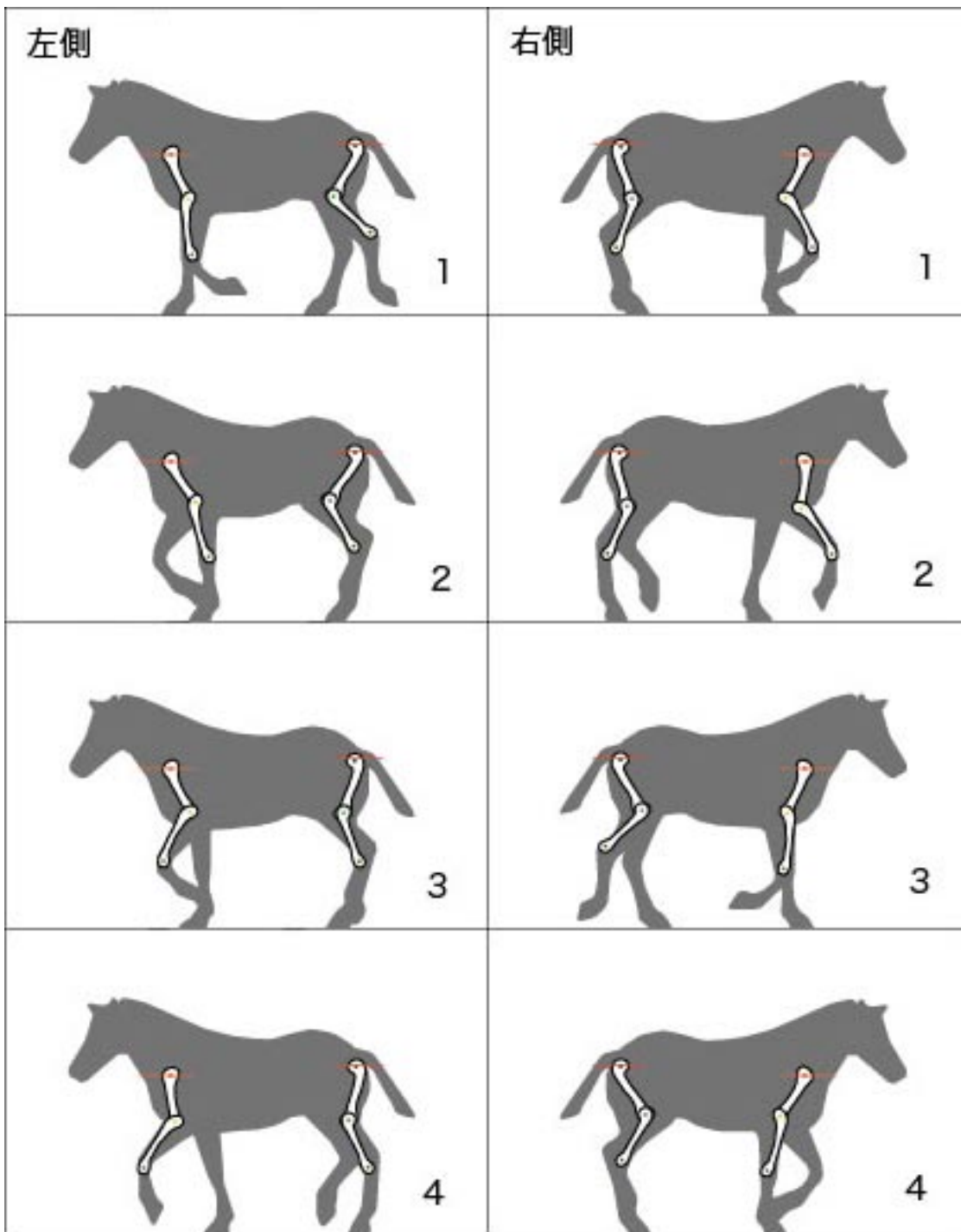
※ CN3、CN4 をアナログセンサーで使用できるように CN5 と CN6 を使用しています。  
ジャンパーの位置に注意してください。J1、J2、J3 は A、J4 は B です。



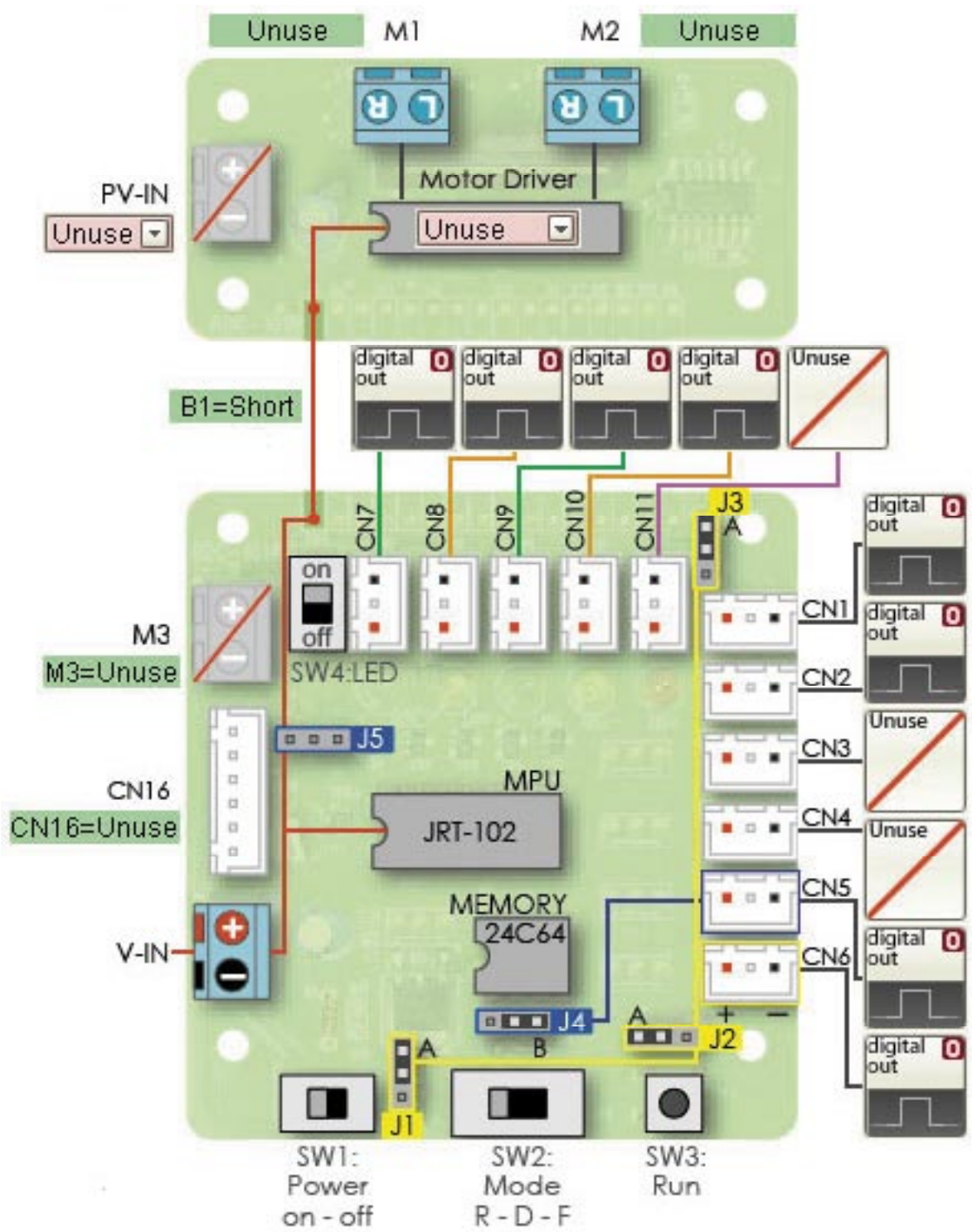


プログラム :8 個のサーボを順番に動かして歩く

プログラムは、動きの途中のポーズを取り出して順番にアニメーションのようにつなげていきます。実験ではサーボ角度のパラメーター設定を簡単にするため、4つのポーズに減らしてプログラムを考えます。



ハードウェア設定





## プログラムソース（変更部分）

※正常にコンパイル後、ダウンロードしてもうまく動かない場合は Stack-Location#2 に設定してください。

```

/*-----*/
/* Variable declaration */
/*-----*/

//pulse to pulse:10-20msec/pulse width:0.7-2.3msec center=1.5msec
#define T_BASE 30 //base for motion interval adjustment
#define PWM_BASE 9 //base for PWM adjustment

/*
4- 3-|F|-1 -2
  | |
8- 7-|B|-5 -6
*/
#define SERVO_1 CN1 //servo-CN matching
#define SERVO_2 CN2
#define SERVO_3 CN5
#define SERVO_4 CN6
#define SERVO_5 CN7
#define SERVO_6 CN8
#define SERVO_7 CN9
#define SERVO_8 CN10

byte walk_repeat;

/*-----*/
/* Prototype declaration */
/*-----*/

void servo(byte deg1, byte deg2, byte deg3, byte deg4, byte deg5, byte deg6, byte deg7, byte deg8, byte motion_time);

/*-----*/
/* main */
/*-----*/

void main(void) {

    init(); //Initialization

    while(1){
        servo(15,15,15,15,15,15,15,3);
        DELAY_SEC(50);
        for(walk_repeat=0;walk_repeat<=5;walk_repeat++){
            servo(15,13,15,15,15,15,15,13,1);
            servo(14,13,13,15,17,15,16,13,1);
            servo(15,15,15,17,15,17,15,15,1);
            servo(17,15,16,17,14,17,13,15,1);
            COPCTL = 0;
        }

        COPCTL = 0;
    }
}

```

```

/*-----*/
/* function */
/*-----*/

void servo(byte deg1, byte deg2, byte deg3, byte deg4, byte deg5, byte deg6, byte deg7, byte deg8, byte motion_time){
    word i;
    word m;
    byte n;
    word p;

    motion_time*=T_BASE;
    for(m=0;m<=motion_time;m++){
        for(n=1;n<=8;n++){
            if(n==1){p=PWM_BASE*deg1,SERVO_1=1;}
            if(n==2){p=PWM_BASE*deg2,SERVO_1=0,SERVO_2=1;}
            if(n==3){p=PWM_BASE*deg3,SERVO_2=0,SERVO_3=1;}
            if(n==4){p=PWM_BASE*deg4,SERVO_3=0,SERVO_4=1;}
            if(n==5){p=PWM_BASE*deg5,SERVO_4=0,SERVO_5=1;}
            if(n==6){p=PWM_BASE*deg6,SERVO_5=0,SERVO_6=1;}
            if(n==7){p=PWM_BASE*deg7,SERVO_6=0,SERVO_7=1;}
            if(n==8){p=PWM_BASE*deg8,SERVO_7=0,SERVO_8=1;}
            for(i=0;i<=p;i++){
                COPCTL = 0;
            }
            COPCTL = 0;
        }
        p=PWM_BASE*40,SERVO_8=0;
        for(i=0;i<=p;i++){
            COPCTL = 0;
        }
        COPCTL = 0;
    }
}

```



## 2 足歩行の実験

4 足用のプログラムのパラメーター（各関節を曲げる順番）を書き換えて 2 足ロボットを動かすこともできます。

写真の実験モデルは、付け根、膝、足首の曲げと傾きで片足 4 自由度、両足で 8 自由度です。

デザインは、4 足よりもバランスが難しくなります。各関節、回転軸の位置関係や重心を低くする工夫、足裏の大きさなどが重要です。このモデルはまだ安定性が低く、よく転んでしまいます。また、さらに単純な組み立て方や、様々なデザインができるでしょう。

市販のヒューマノイドロボットの体の各部分の比率も研究してみてください。人間とはかなり違うことがわかります。

ラジコン用サーボをベースにした市販のヒューマノイドロボットも基本的にはこの方法で歩いています。

