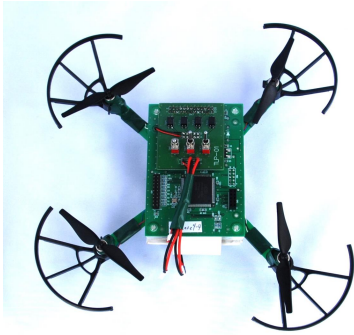




Drone3

D-Slave8

D-Slave9

Music Drone Project

製作開始 2023/02/22

アップロミュージックスタジオ事務所

2024/08/27 h.

目標

- ・ 機体の軽量化
- ・ 揚力を上げる
とりあえず空中へ上昇、
ホバリングを目指す。
- ・ ジャイロを搭載する。
- ・ 距離センサーを付ける。
- ・ GPSを付ける。

浮力のテスト 1 (D-Slave8)

機体の重量を軽くして
浮力をテストする。

方法

CPU基板と機体を分離して
テストを行う。

モーター x 4

バッテリーは

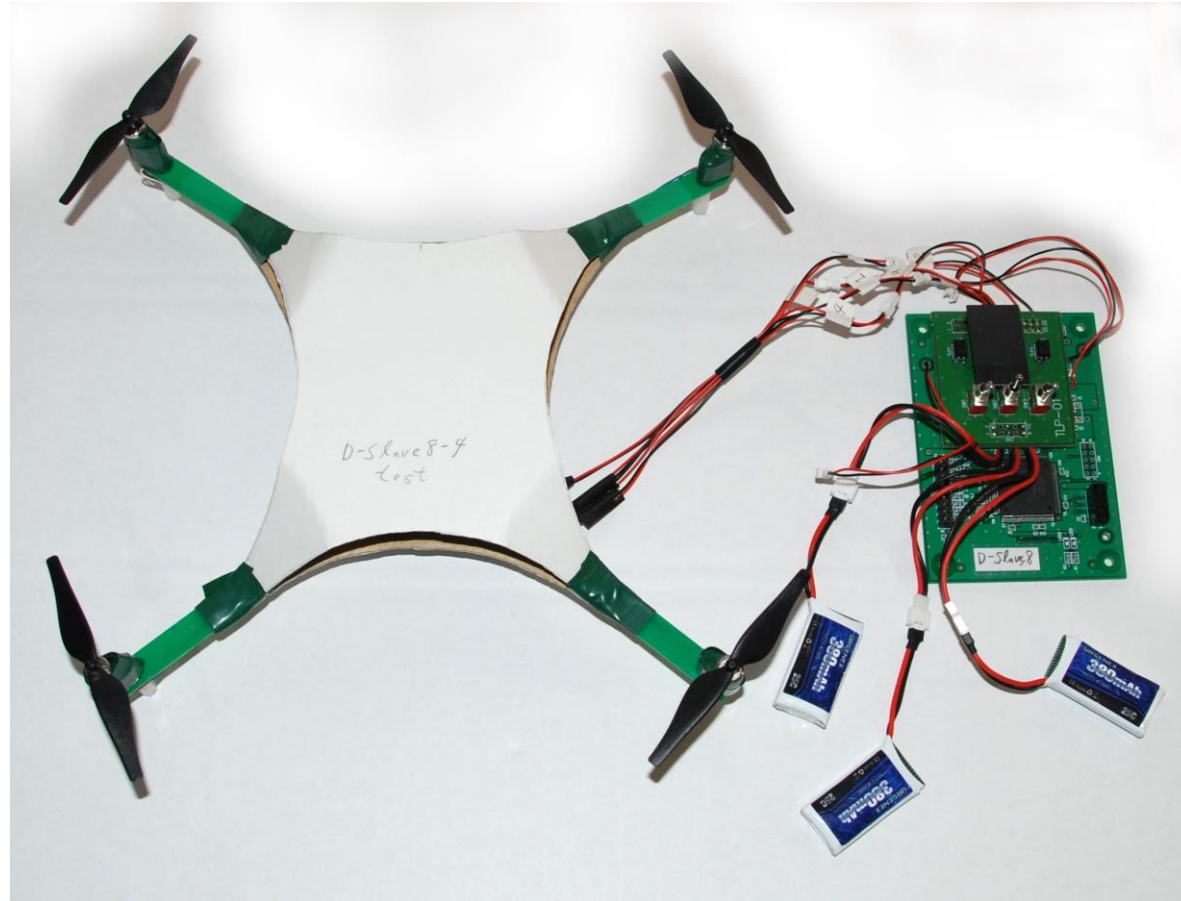
3.7V 380mAh

CPU 基板 x 1

ミニモーター用 x 2

結果

機体のみでの浮力はOK



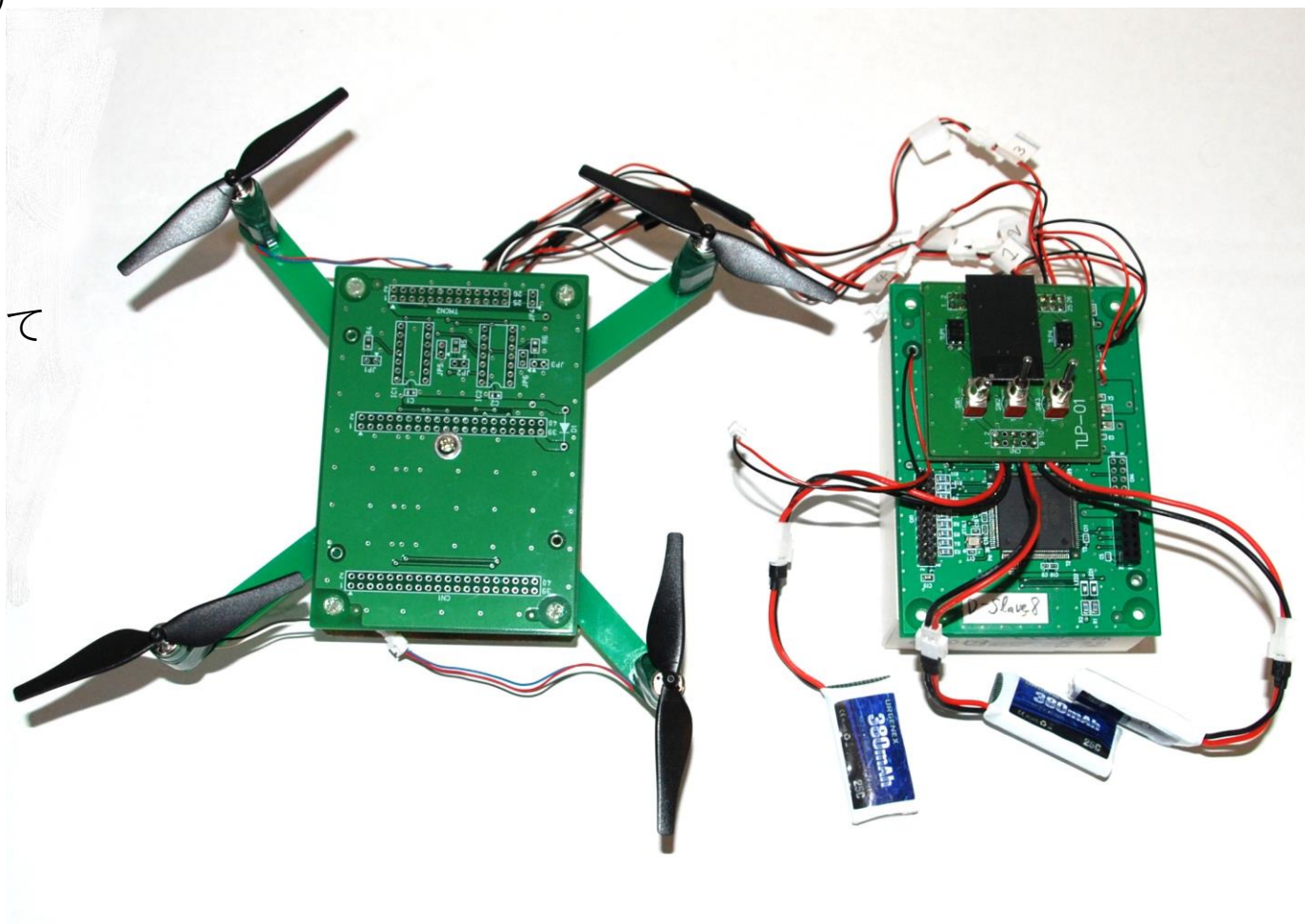
浮力のテスト 2 (D-Slave8)

機体の重量を軽くして
浮力をテストする。

方法
未使用基板で機体を作成して
テストを行う。

モーター x 4
バッテリーは
3.7V 380mAh
CPU 基板 x 1
ミニモーター用 x 2

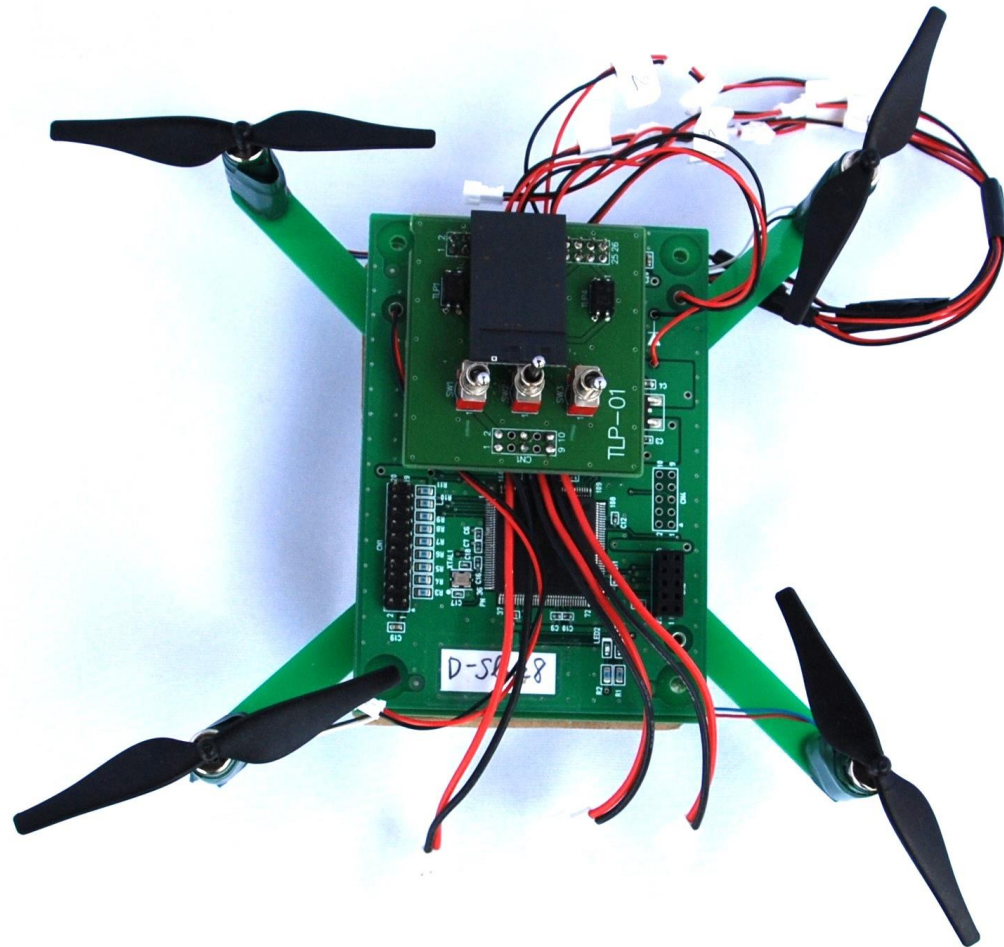
結果
機体のみでの浮力はOK



CPU基板と機体を組み合わせた
写真（基板のサイズは同じ）

結果
バッテリーの完全充電を行い最
初の間は、xx秒くらい浮くこと
ができる。

機体が重いと思われる。



浮力のテスト 3 (D-Slave8-8)

モーター、プロペラを2段にして浮力をテストする。

モーター x 8
バッテリーは
3.7V 380mAh
CPU 基板 x 1
ミニモーター用 x 2

結果

期待したほどの浮力は得られない。
バッテリーの容量が不足する。
バッテリーを追加すると重量が増えてしまうので結局浮力は上がらないためNGとする。



D-Slave8-4-4

重量軽減

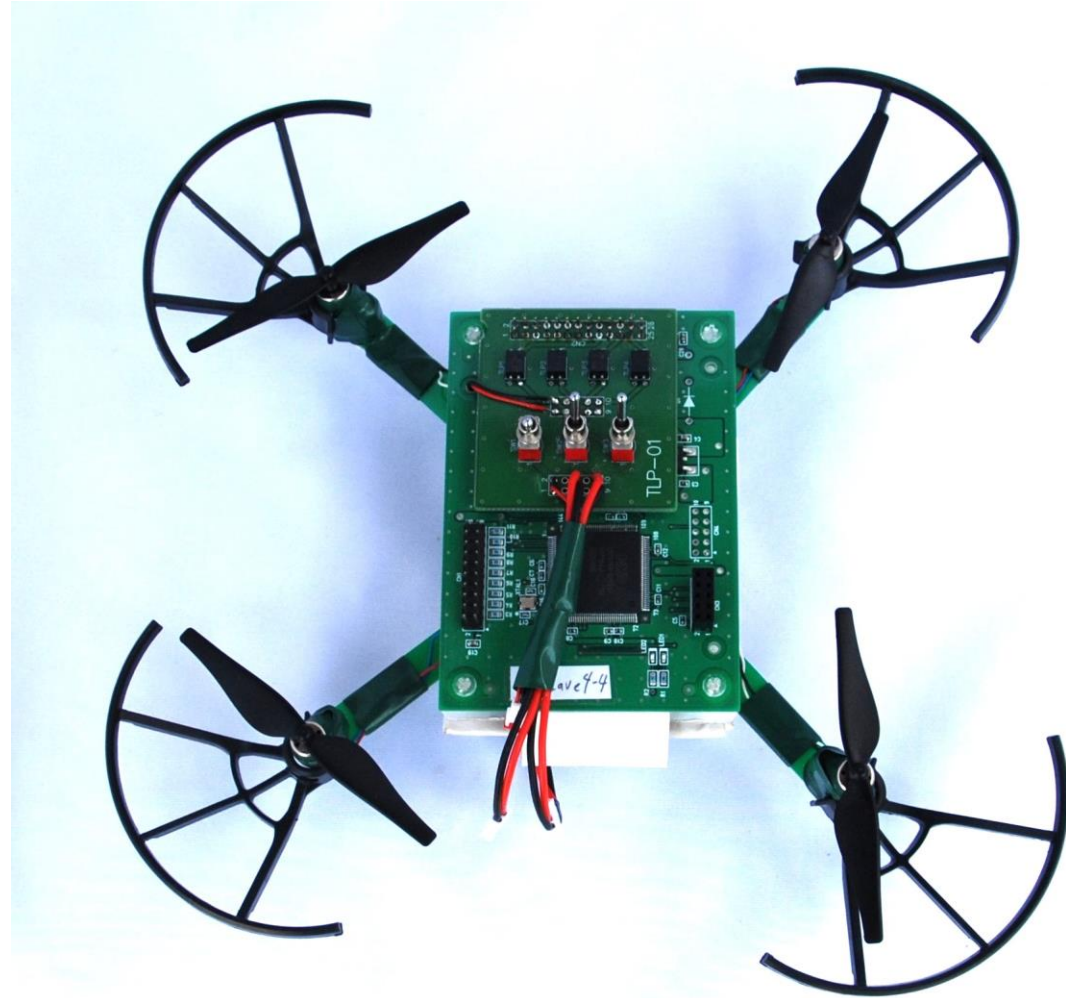
追加

プロペラガードを取り付け

結果

浮力 OK

ホバリング NG



D-Slave8-4-4

重量軽減

バッテリー

CPU 3.7V 220mAh x 1

モーター 3.7V 380mAh x 1

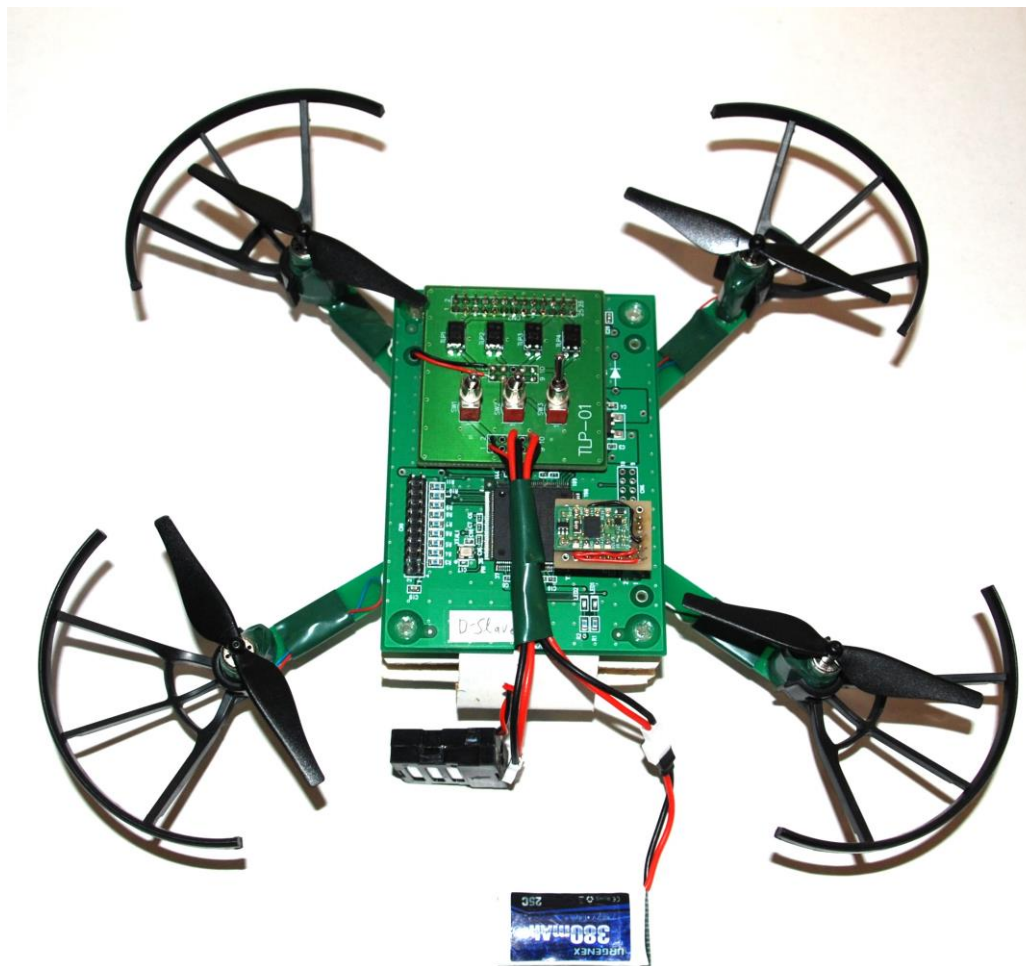
未使用基板を紙台紙に変更

追加

9軸センサーを取り付け

プロペラガードを取り付け

結果



D-Slave8-4-4

重量軽減

バッテリー

CPU 3.7V 220mAh x 1

モーター 3.7V 380mAh x 1

未使用基板を紙台紙に変更

追加

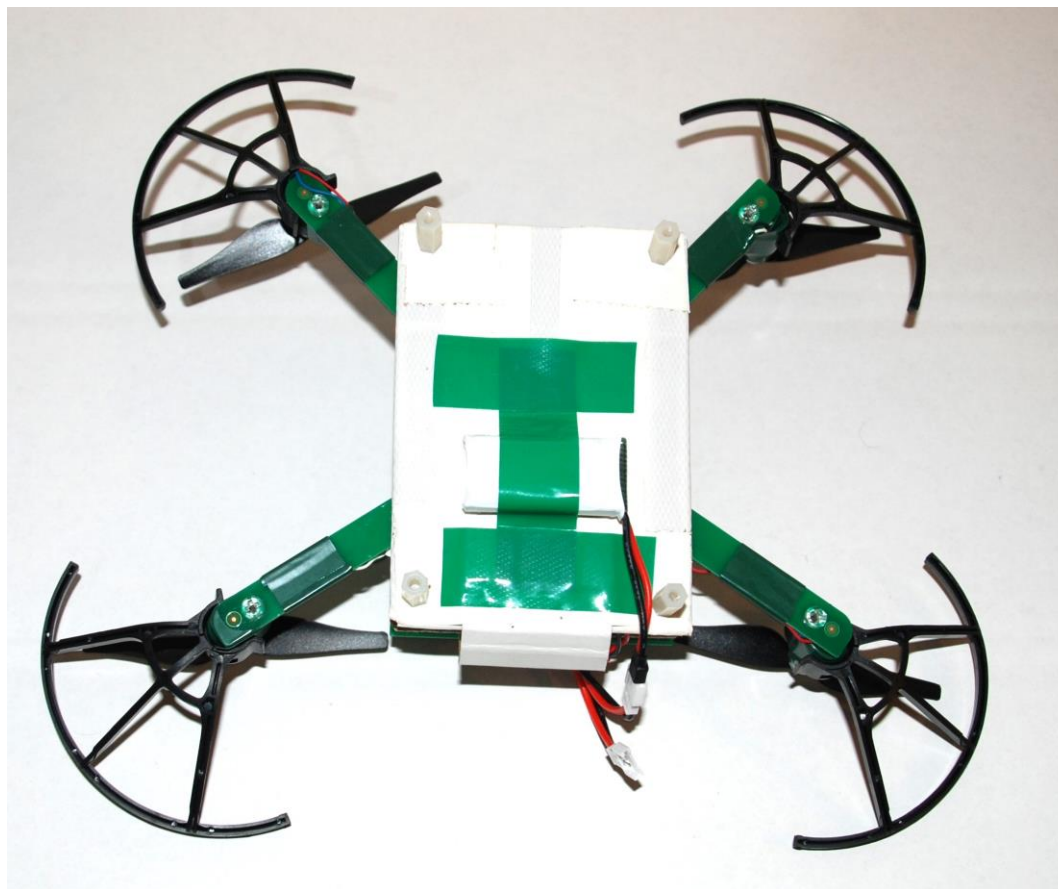
9軸センサーを取り付け

プロペラガードを取り付け

バッテリー固定

ビニールテープで裏面に

貼り付け



D-Slave8-4-4

重量軽減

バッテリー

CPU 3.7V 220mAh x 1

モーター 3.7V 380mAh x 1

未使用基板を紙台紙に変更

追加

9軸センサーを取り付け

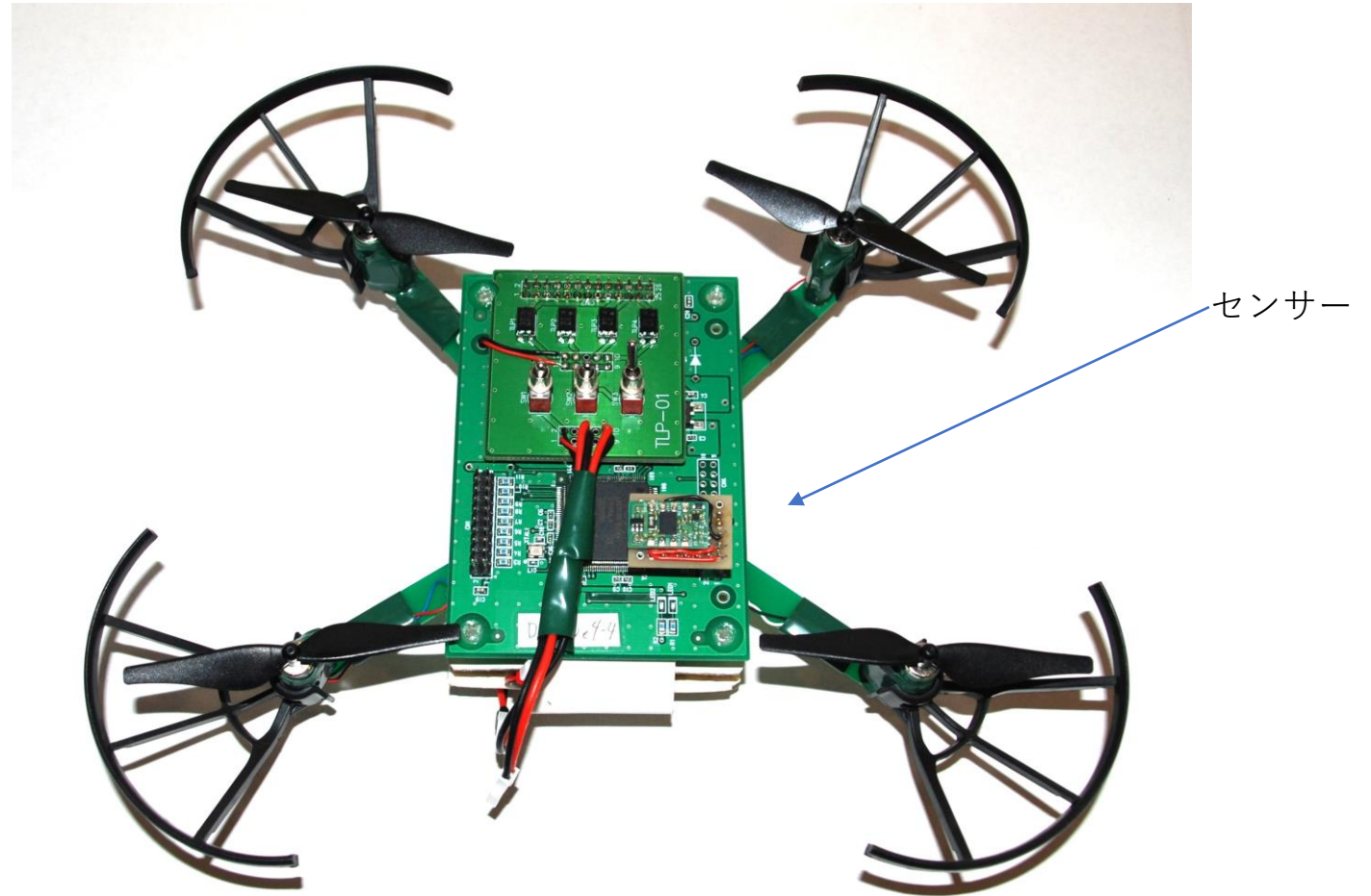
プロペラガードを取り付け

結果

浮力 OK

ホバリング NG

安定飛行するには、機体の変更を含めて全体の見直しが必要、ドローンコントロールのテストには飛行物体は確認が困難なため地上走行の車にてテストを行うことにする。



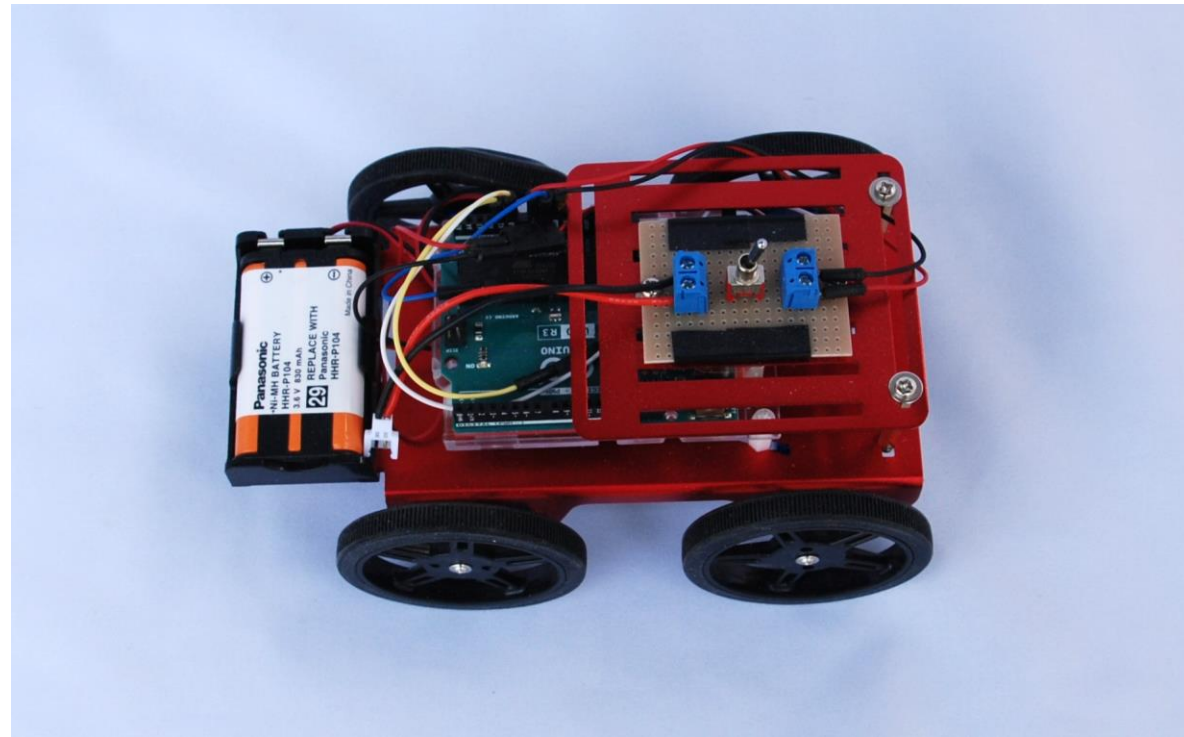
手始めに
動きは前進のみですがパワフル
に動きます。
コントロールは無しです。



```
#include <Wire.h>
#include <Servo.h>

int R_SPEED = 300; //-500 to 500
(0=STOP)
int L_SPEED = 300; //-500 to 500
(0=STOP)
Servo L_SERVO;
Servo R_SERVO;
void setup()
{
  L_SERVO.attach(14,1000,2000); //PORT(A0), MIN, MAX
  R_SERVO.attach(15,1000,2000); //PORT(A1), MIN, MAX
}
void loop()
{
  L_SERVO.writeMicroseconds(1500 + L_SPEED);
  R_SERVO.writeMicroseconds(1500 - R_SPEED);
  while(1);
}
```

Arduino uno を使ってサンプルプログラムを動かしてみました。（前進のみ）
Minirobot-4wd-sample

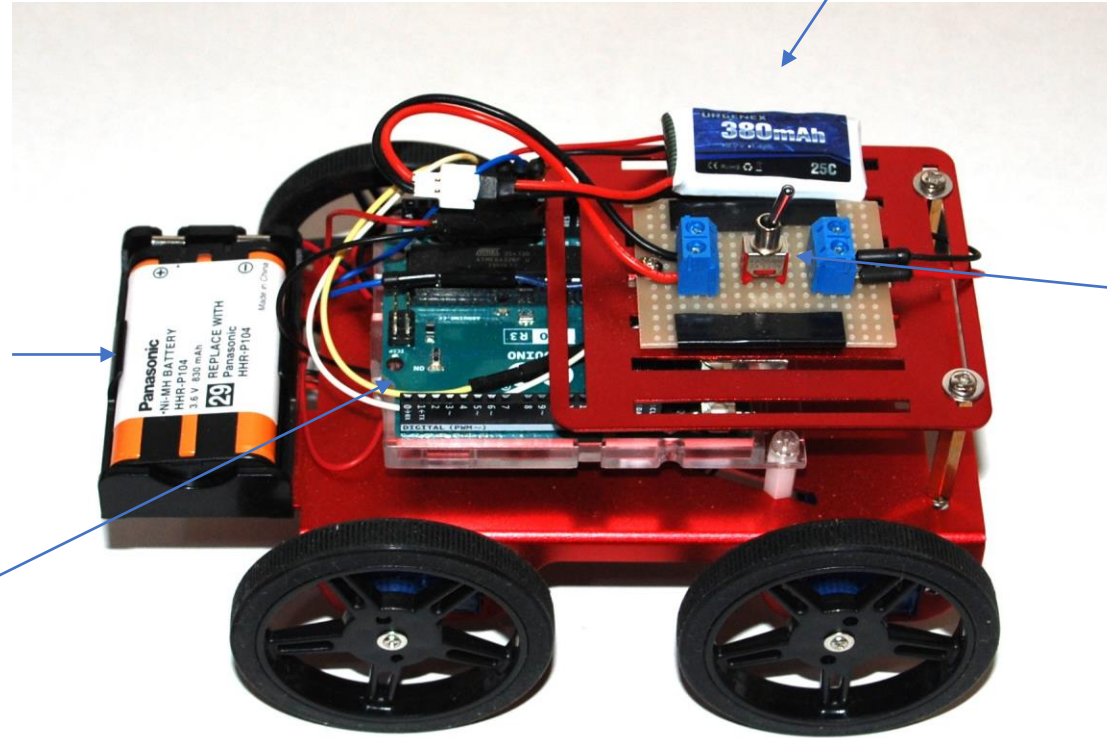


UNO用バッテリー
DC 3.7 V

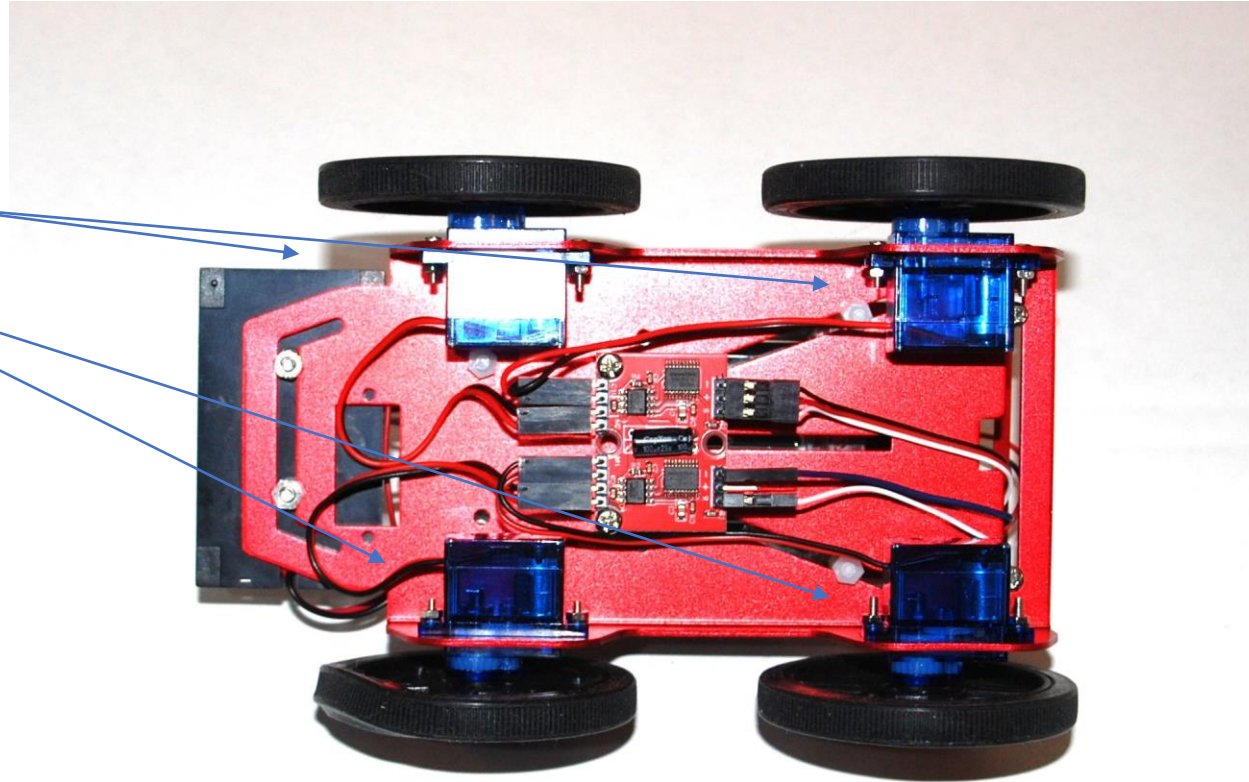
モーター用バッテリー
DC 3.6V

CPU電源スイッチ

Arduino UNO

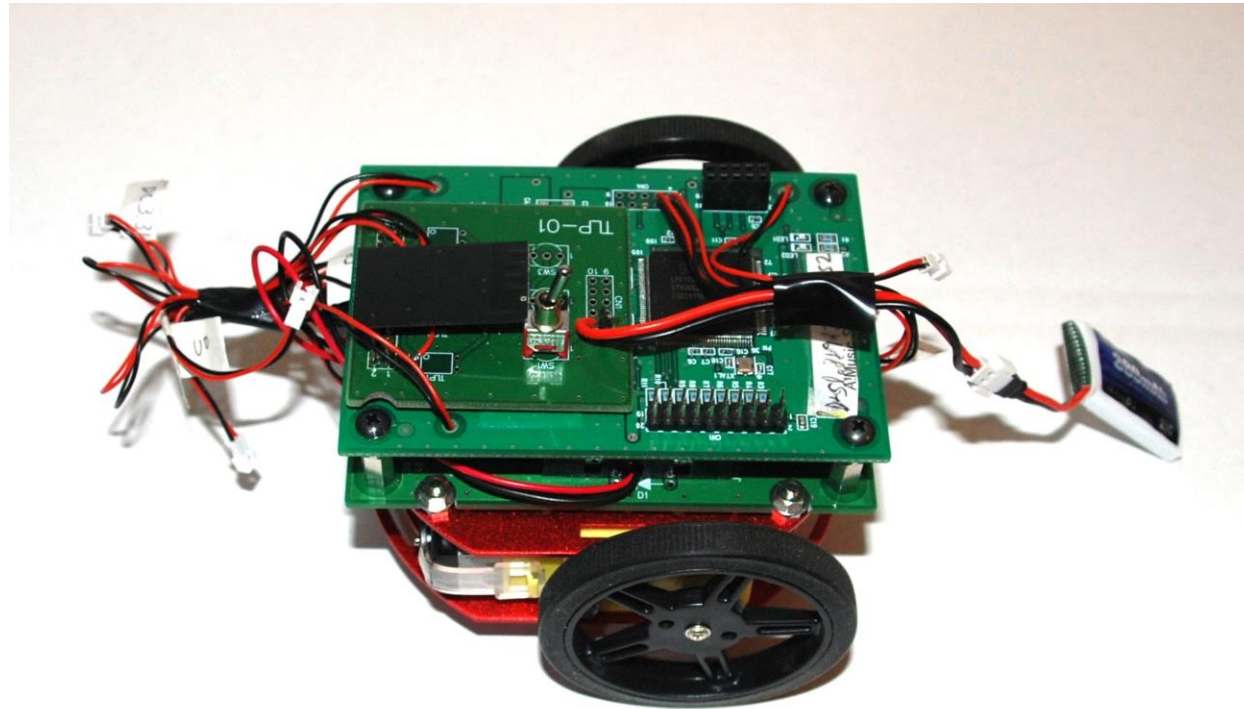


DCモーター x 4

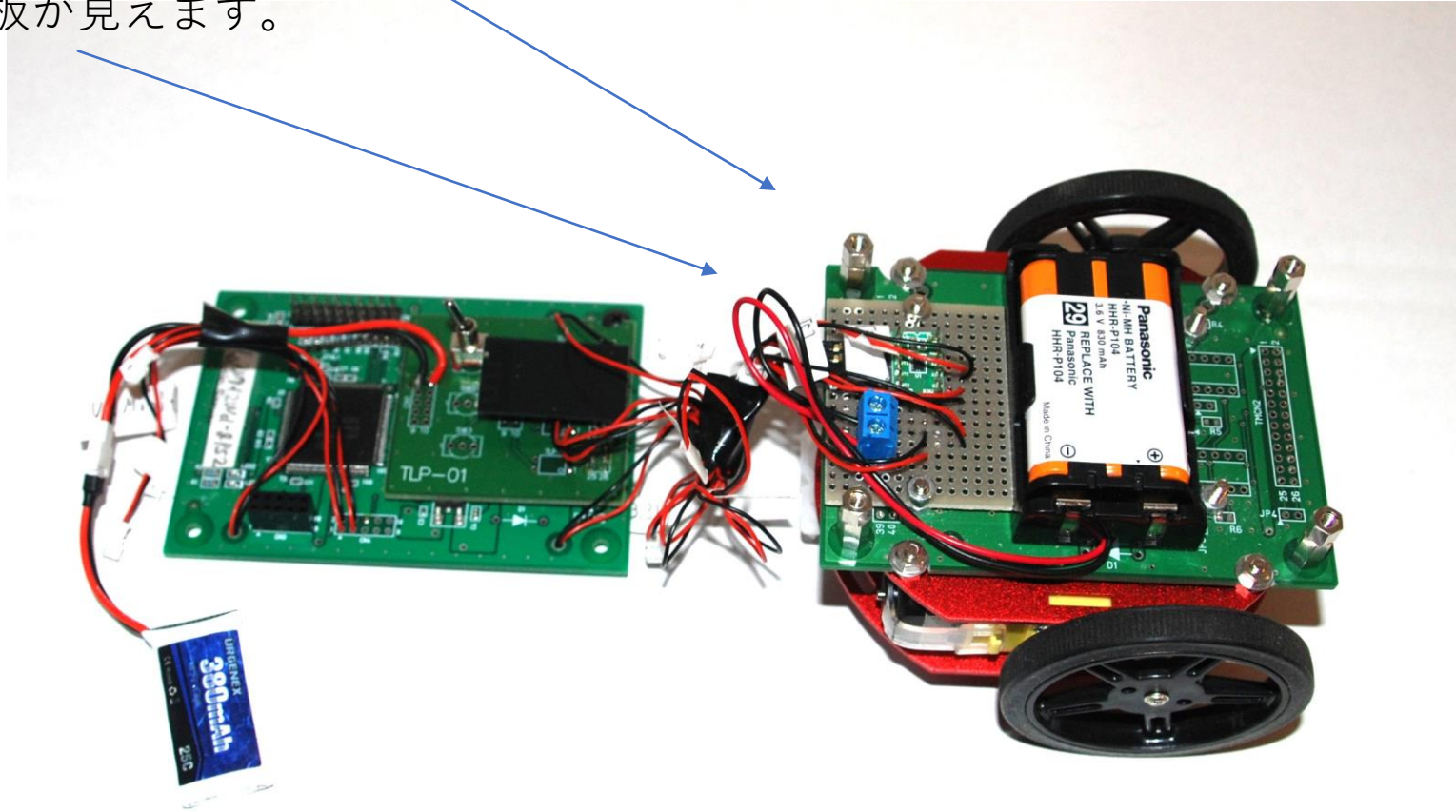


D-Slave9-2wd

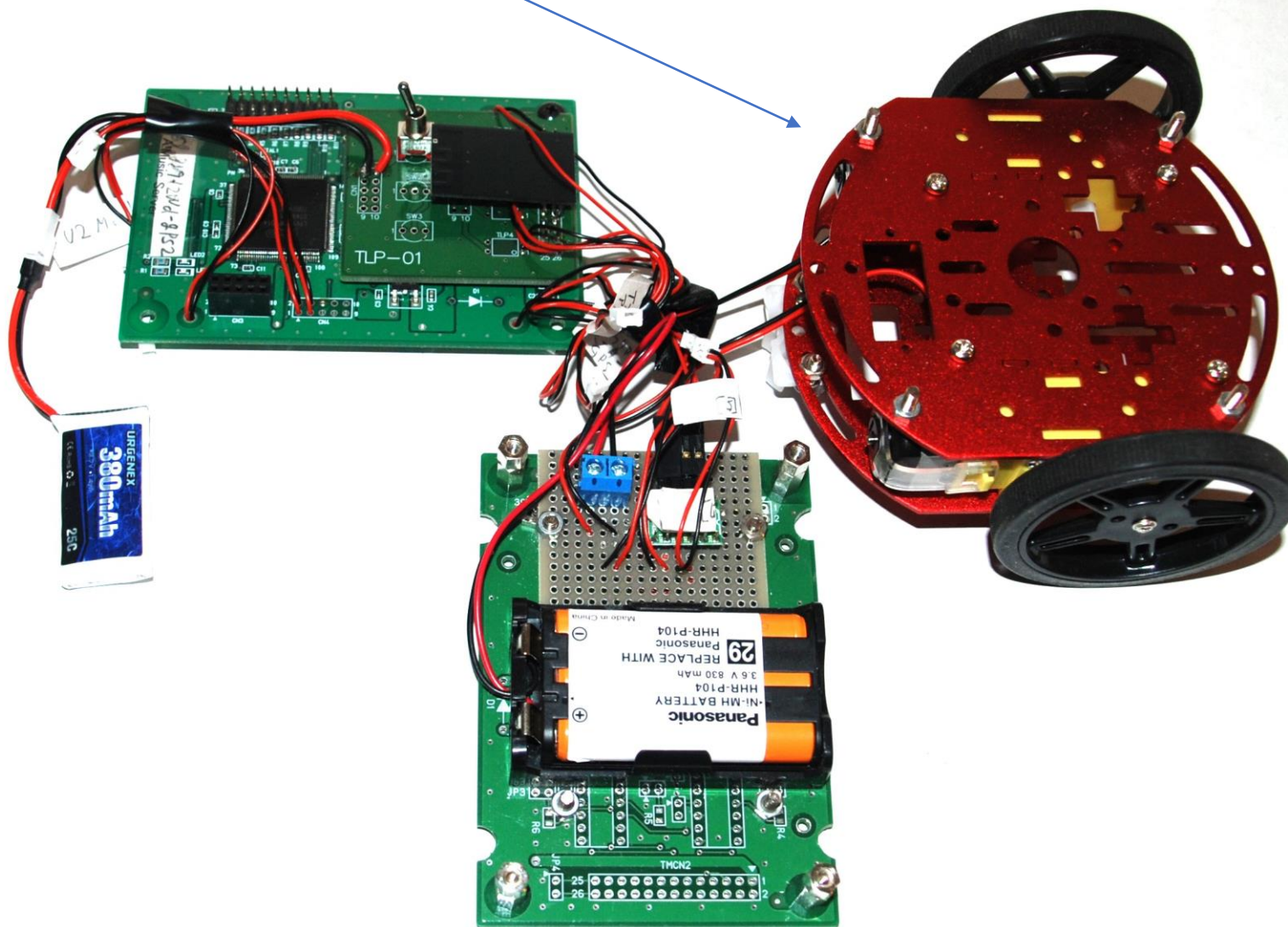
2WD Mini Smart Robot プラットホームキットを
LPC1788 CPU基板を使って、DCモーターを
コントロールします。



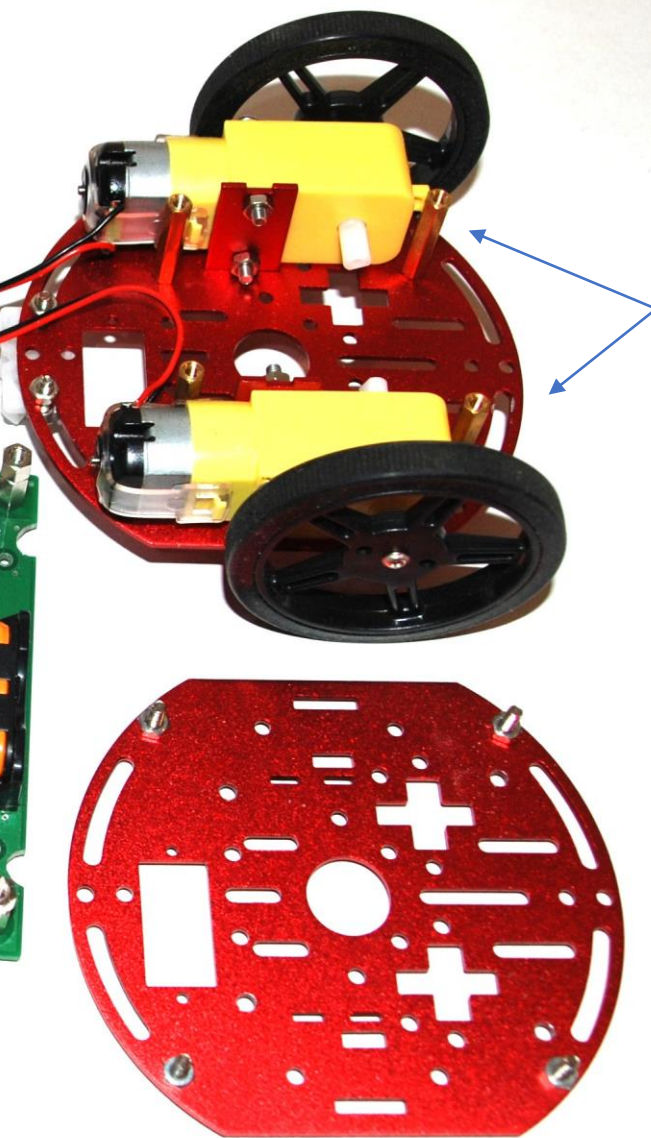
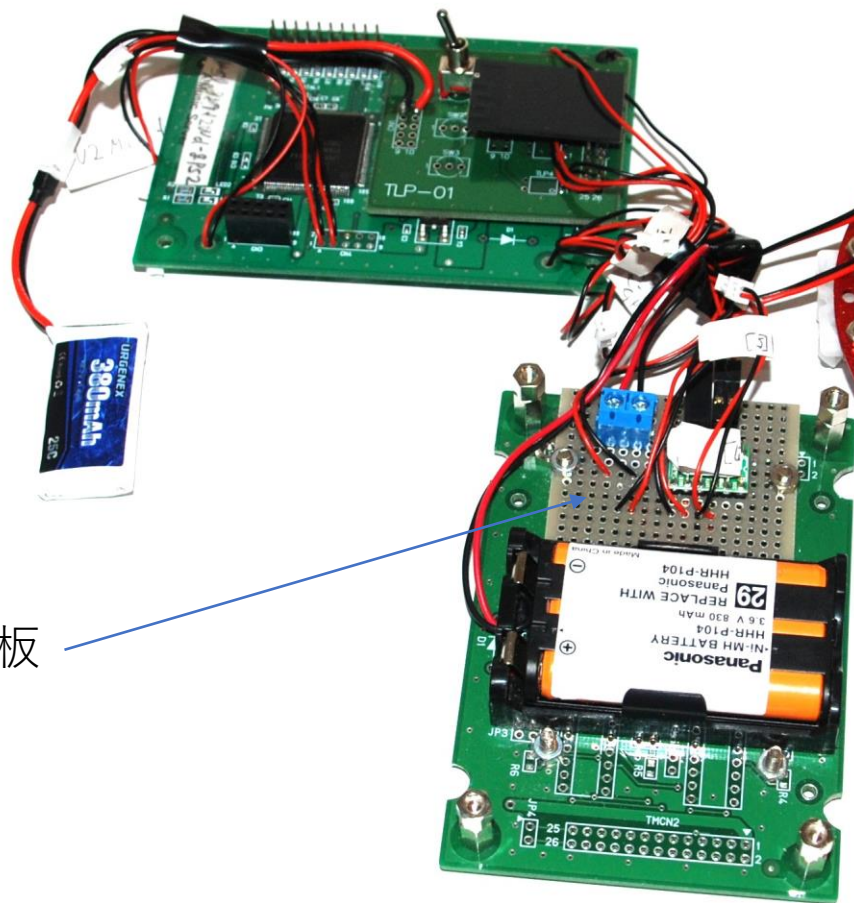
CPU基板を外し、バッテリー、
DRV8835基板が見えます。



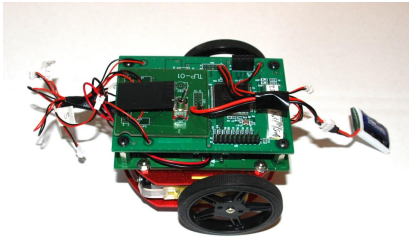
中間のフレーム板を外します。



DRV8835 基板



DCギヤモーター



D-Slave9-2wd program

```

if(check) {
    //gpio_drv8835_tst();           // DC moter test
    //gpio_drv8835_wfR(80);
    //for( i=0; i < 4000000; i++); //delay
    //gpio_drv8835_wbL(80);         // test
    //for( i=0; i < 4000000; i++); //delay

    //gpio_drv8835_wfL(80);
    //for( i=0; i < 4000000; i++); //delay
    //gpio_drv8835_wbR(80);
    //for( i=0; i < 4000000; i++); //delay

    for(j=0; j<2; j++) {
        gpio_drv8835_wf(200);      // test
        for( i=0; i < 9000000; i++); //delay
        gpio_drv8835_wb(100);      // test
        for( i=0; i < 3000000; i++); //delay
    }
    //for( i=0; i < 4000000; i++); //delay
    //gpio_drv8835_wtR(80);
    //for( i=0; i < 4000000; i++); //delay
    //gpio_drv8835_wtL(8);
    //for( i=0; i < 4000000; i++); //delay
    //gpio_drv8835_wtR(3);
}
/// drv8835 DC moter test ///////////

```

```

int gpio_drv8835_wb(vs)
int vs;
{
    int i,j;

    GPIO_P0_4_Low(); // AIN1-APHASE
    GPIO_P0_5_High(); // AIN2-AENBL
    GPIO_P0_6_Low(); // BIN1-PHASE
    GPIO_P0_7_High(); // BIN2-BENBL

    for(j=0; j < vs; j++) {

        GPIO_P0_4_Low();
        for( i=0; i < 90000; i++); //delay
        //GPIO_P0_4_Low();
        for( i=0; i < 10000; i++); //delay
        //GPIO_P0_4_High(); // A-Enable

        GPIO_P0_6_Low(); // B-Enable
        for( i=0; i < 90000; i++); //delay
        //GPIO_P0_6_Low(); // B-Enable
        for( i=0; i < 10000; i++); //delay
        //GPIO_P0_6_High(); // B-Enable

    }

    GPIO_P0_4_Low(); // DRV8835 A-IN1-Enable Low
    GPIO_P0_5_Low(); // DRV8835 a-IN1-Enable Low
    GPIO_P0_6_Low(); // DRV8835 B-in1-Phase Low
    GPIO_P0_7_Low(); // DRV8835 B-in2-Phase Low

    return(0);

// Forward PASE/Enable model
int gpio_drv8835_wf(vs)
int vs;
{
    int i,j;

    GPIO_P0_4_Low(); // AIN1-APHASE
    GPIO_P0_5_High(); // AIN2-AENBL
    GPIO_P0_6_Low(); // BIN1-PHASE
    GPIO_P0_7_High(); // BIN2-BENBL

    for(j=0; j < vs; j++) {

        GPIO_P0_4_High();
        for( i=0; i < 90000; i++); //delay
        //GPIO_P0_4_Low();
        for( i=0; i < 10000; i++); //delay
        //GPIO_P0_4_Low(); // A-Enable

        GPIO_P0_6_High(); // B-Enable
        for( i=0; i < 90000; i++); //delay
        //GPIO_P0_6_Low(); // B-Enable
        for( i=0; i < 10000; i++); //delay
        //GPIO_P0_6_High(); // B-Enable

    }

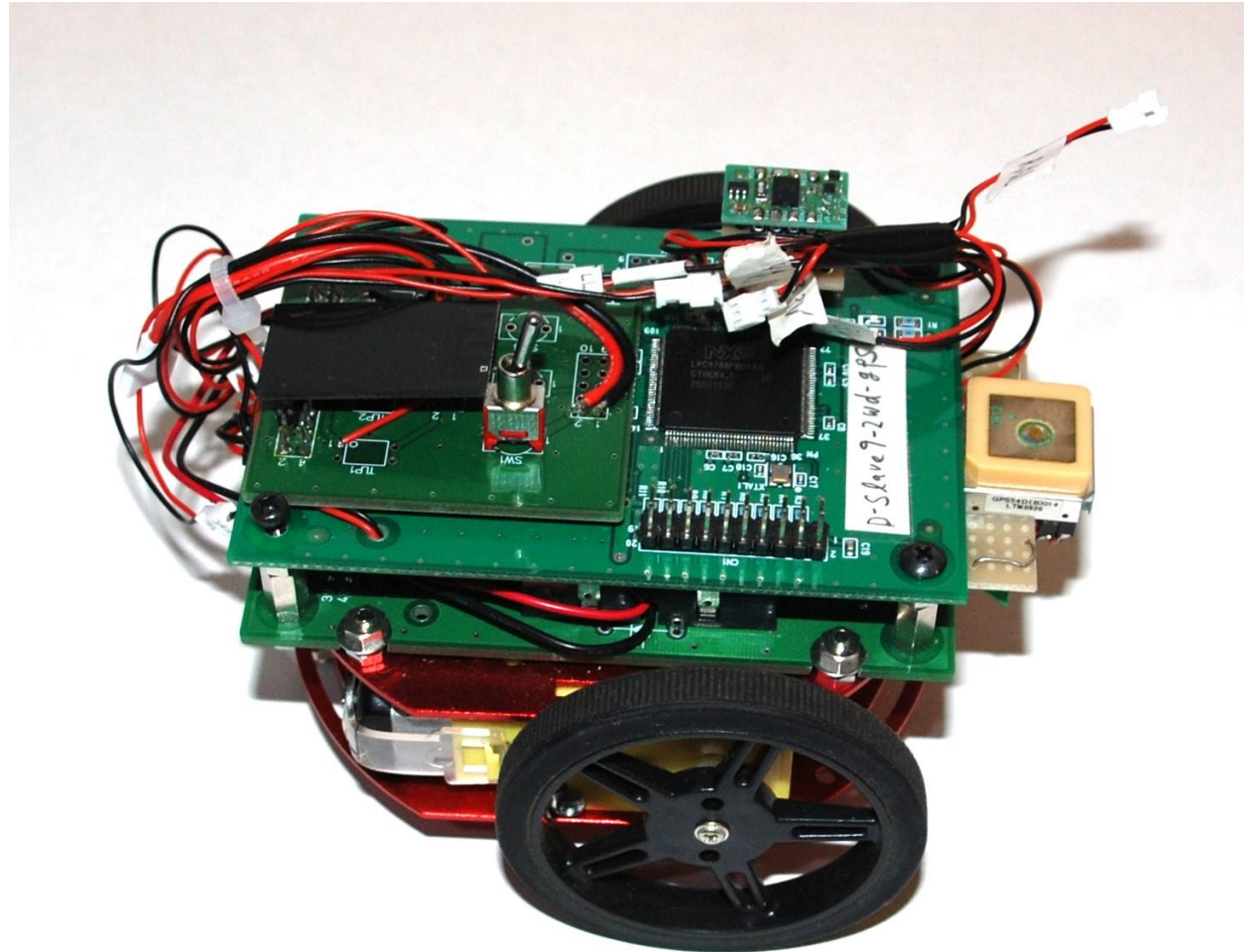
    GPIO_P0_4_Low(); // DRV8835 A-IN1-Enable Low
    GPIO_P0_5_Low(); // DRV8835 a-IN1-Enable Low
    GPIO_P0_6_Low(); // DRV8835 B-in1-Phase Low
    GPIO_P0_7_Low(); // DRV8835 B-in2-Phase Low

    return(0);
}

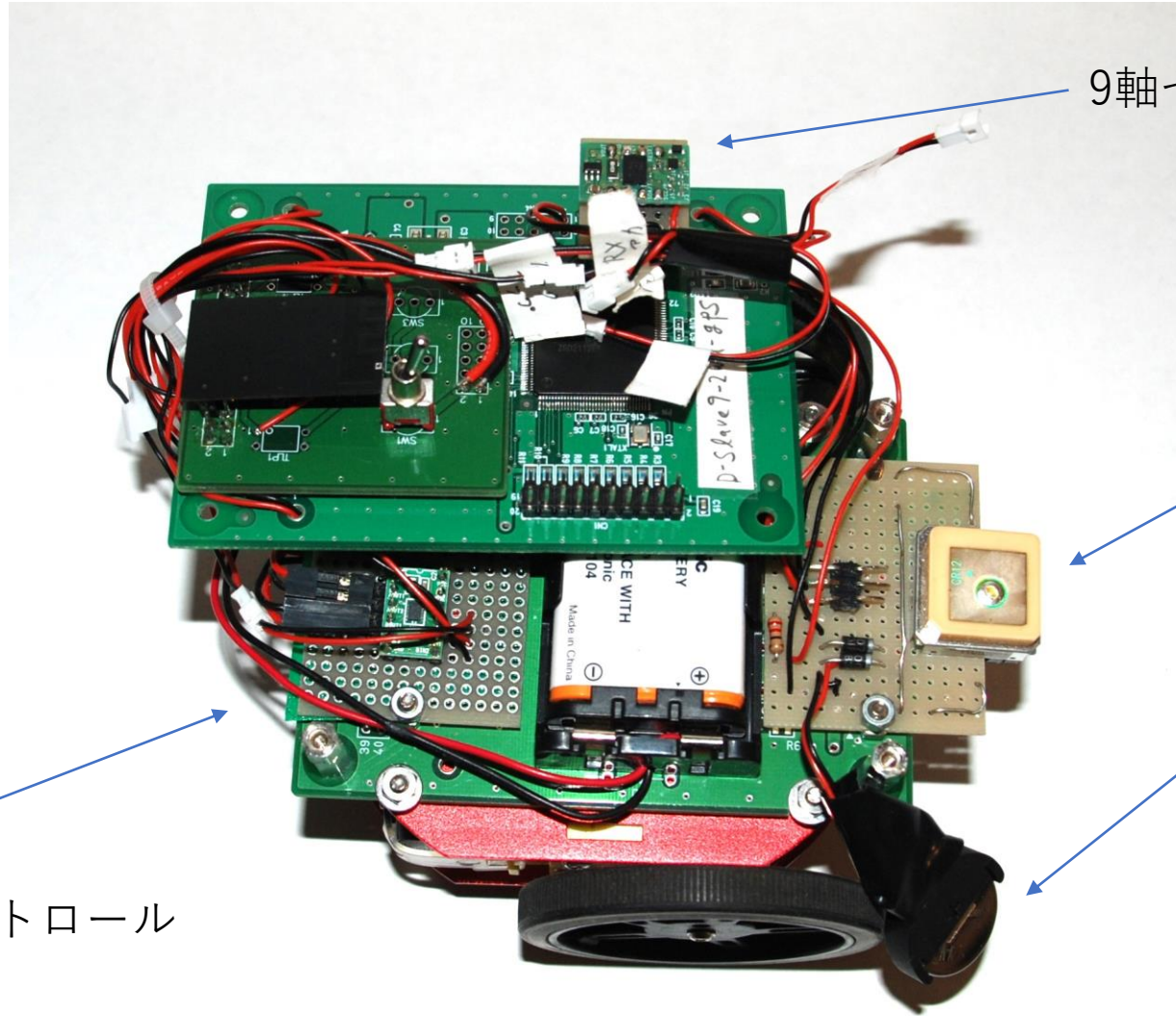
```

D-Slave9-2wd-gps

2WD Mini Smart Robot プラットホームキットを
LPC1788 CPU基板を使って、DCモーターを
コントロールします。
(GPSモジュールを追加)



DRV8835基板
DCモーターコントロール



9軸センサー（ジャイロ他）

GPSモジュール

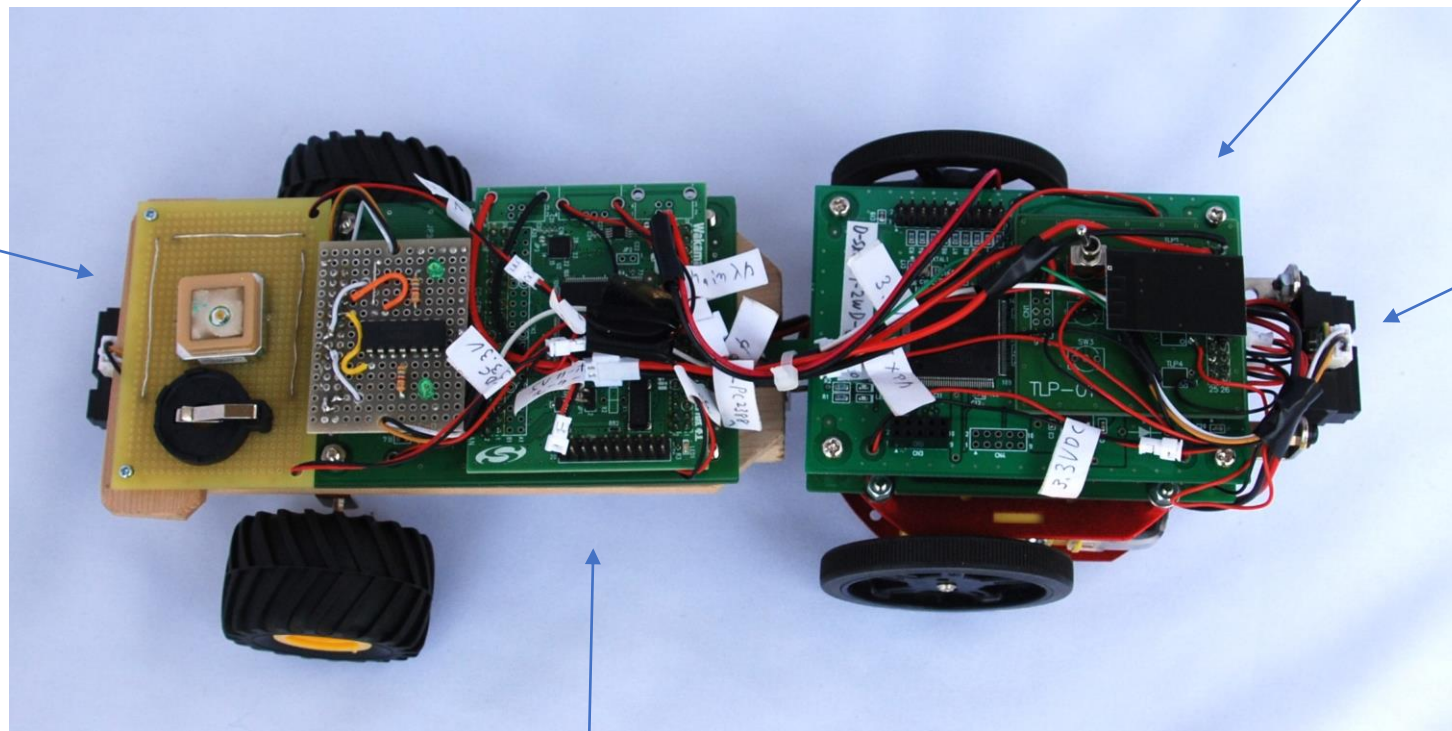
GPS用補助バッテリー

LPC2388 アナログ入力インターフェースを使った、
距離センサーモジュールのテスト

GPSモジュール

LPC1788基板

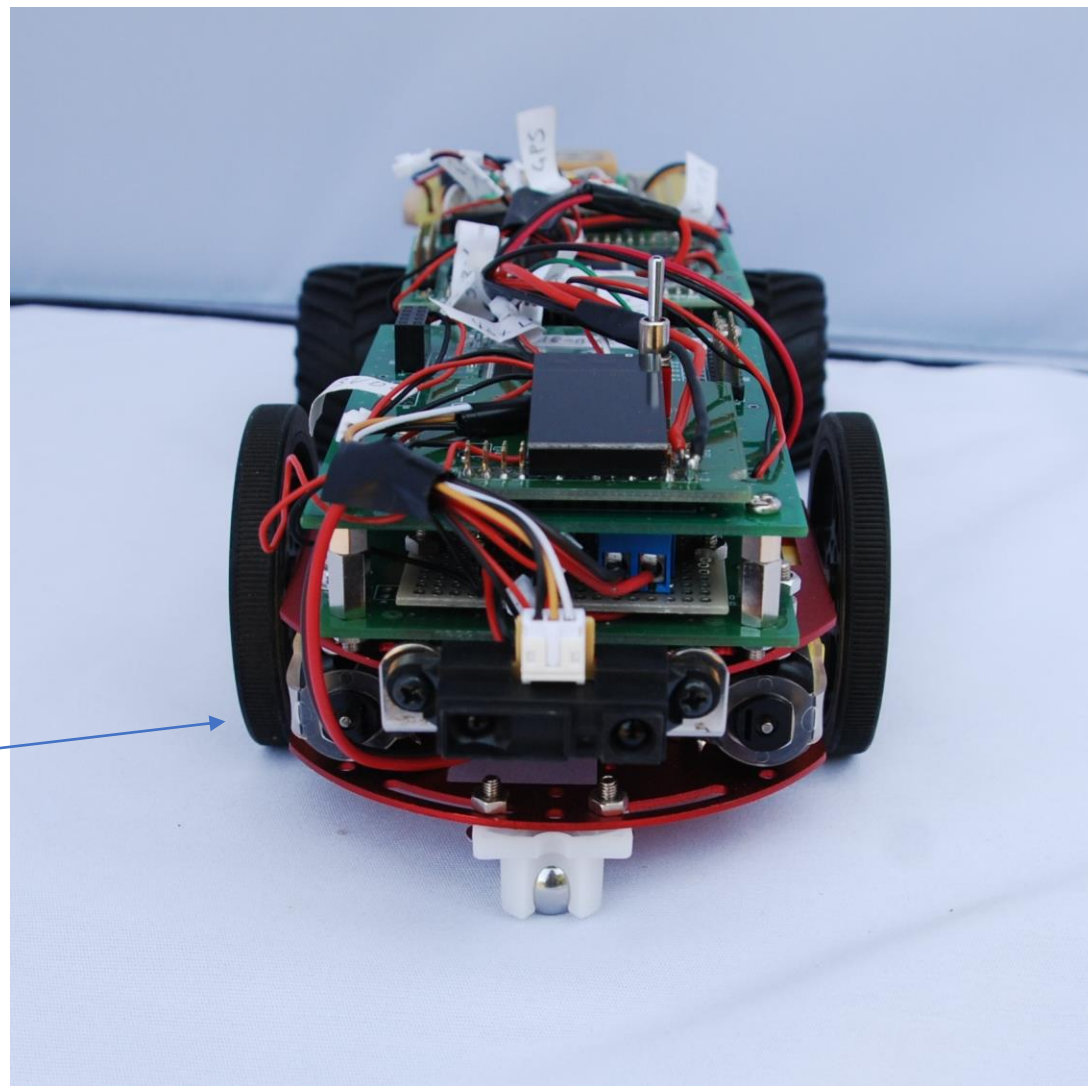
距離センサー
約30 cm 認識



LPC2388基板

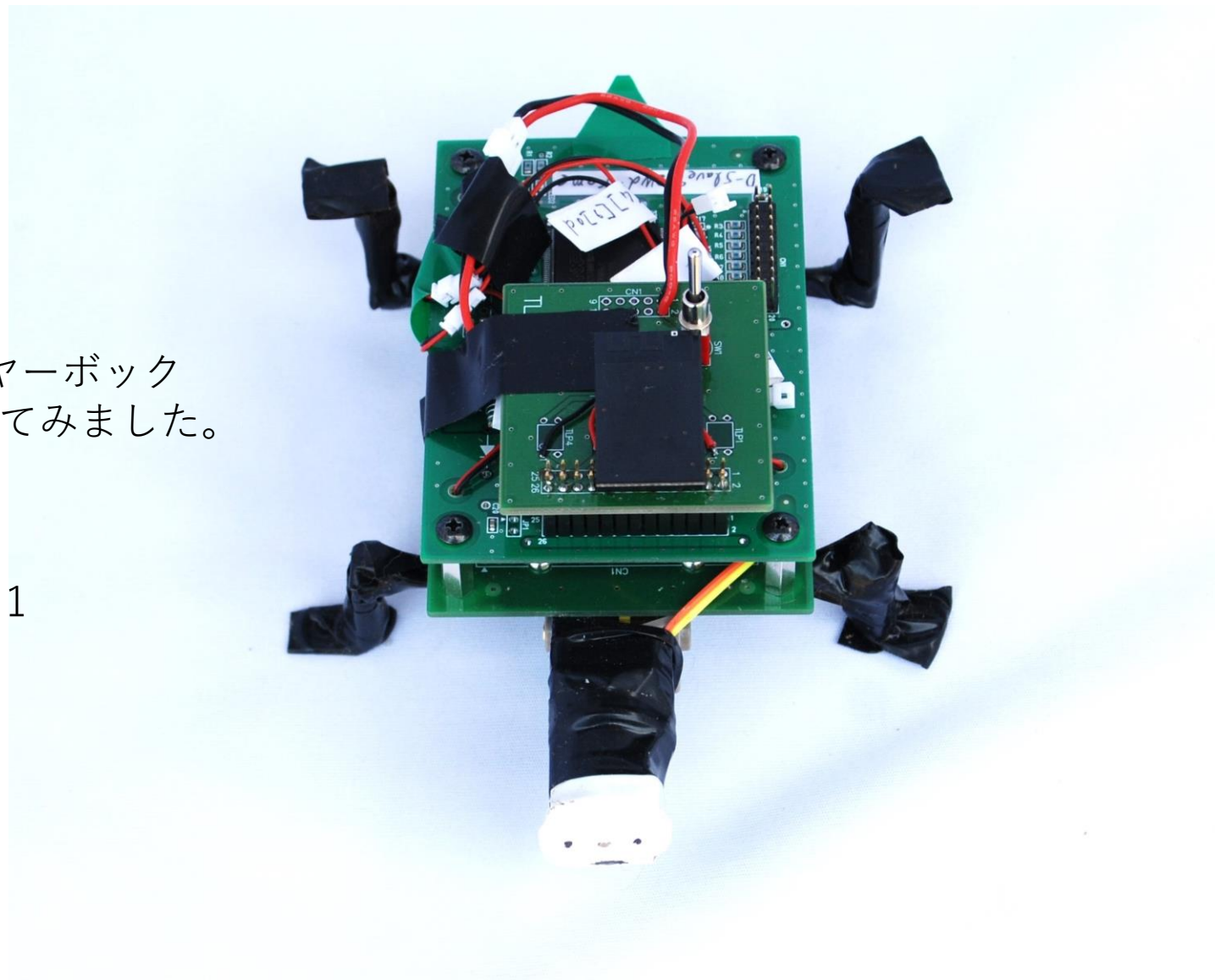
LPC1788基板のGPIOを使って距離センサーを動作の確認が出来ました。(GPS + 距離センサー)

距離センサー



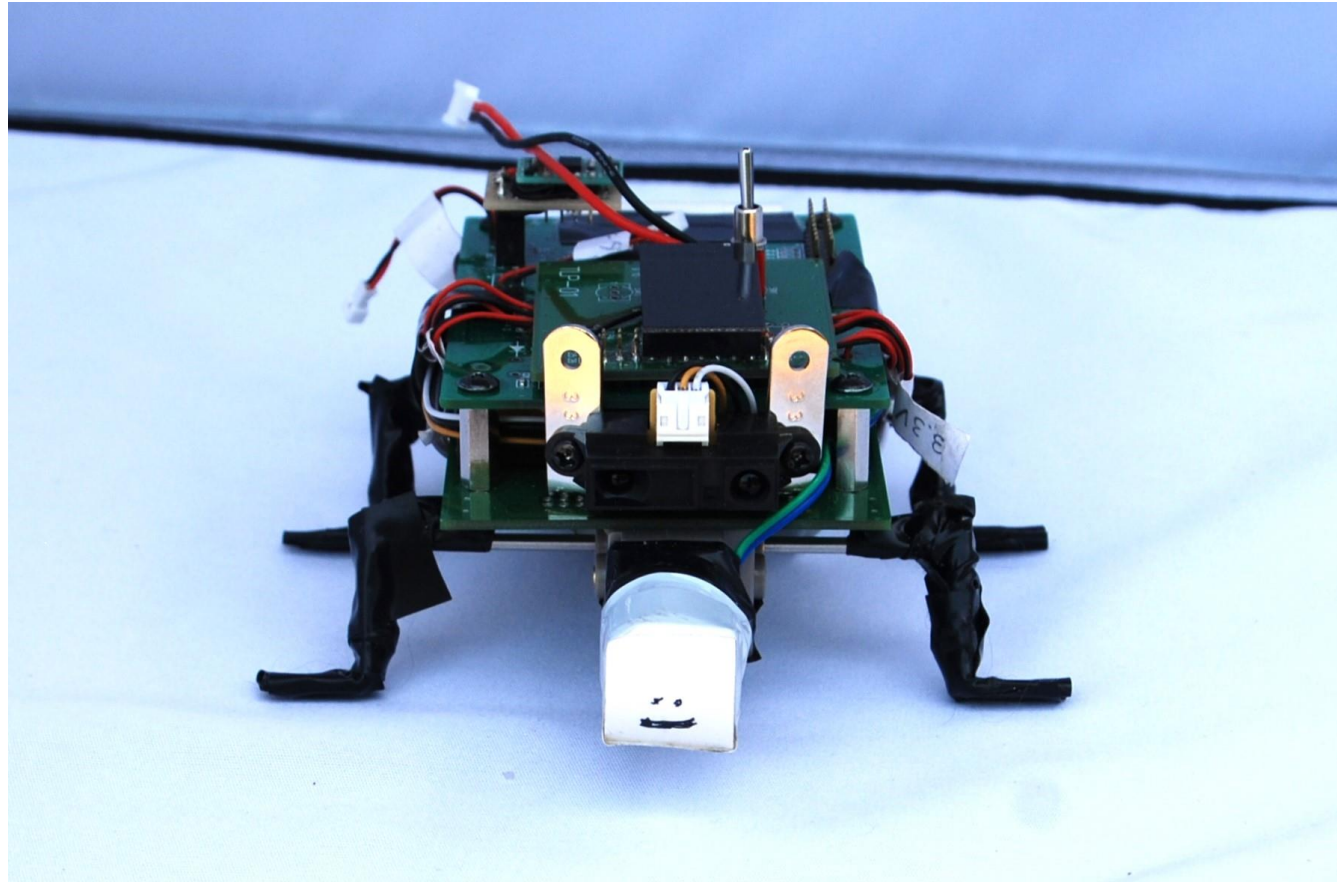
タミヤの3速クランクギヤーボックス
を使って歩く亀を作ってみました。

ギヤーボックス x 2
LPC1788 CPU 基板 x 1
TWELITE x 1



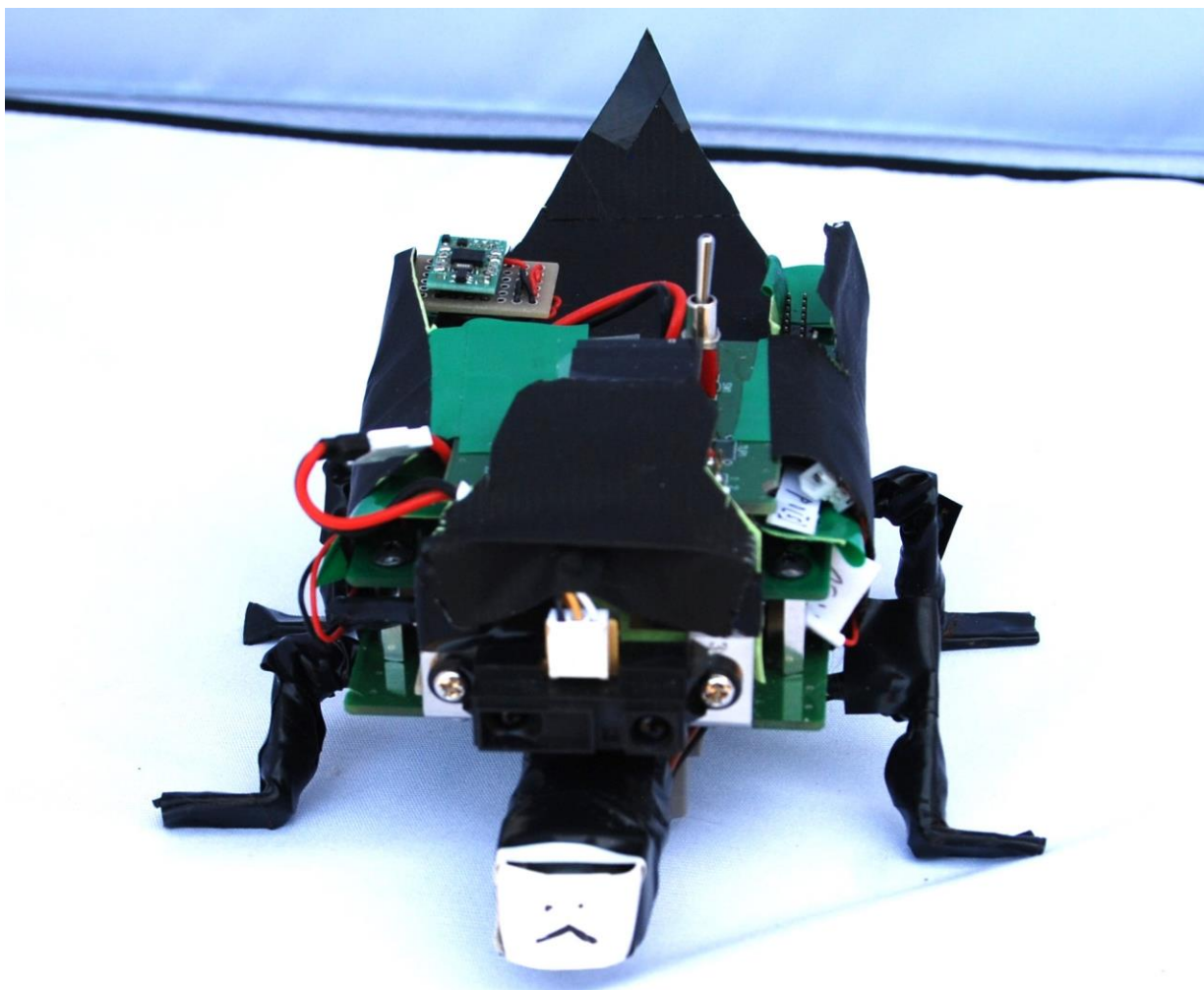
タミヤの3速クランクギヤーボックス
を使って歩く亀を作ってみました

ギヤーボックス x 2
LPC1788 CPU 基板 x 1
TWELITE x 1
BNO055 ジャイロ
距離センサーを追加

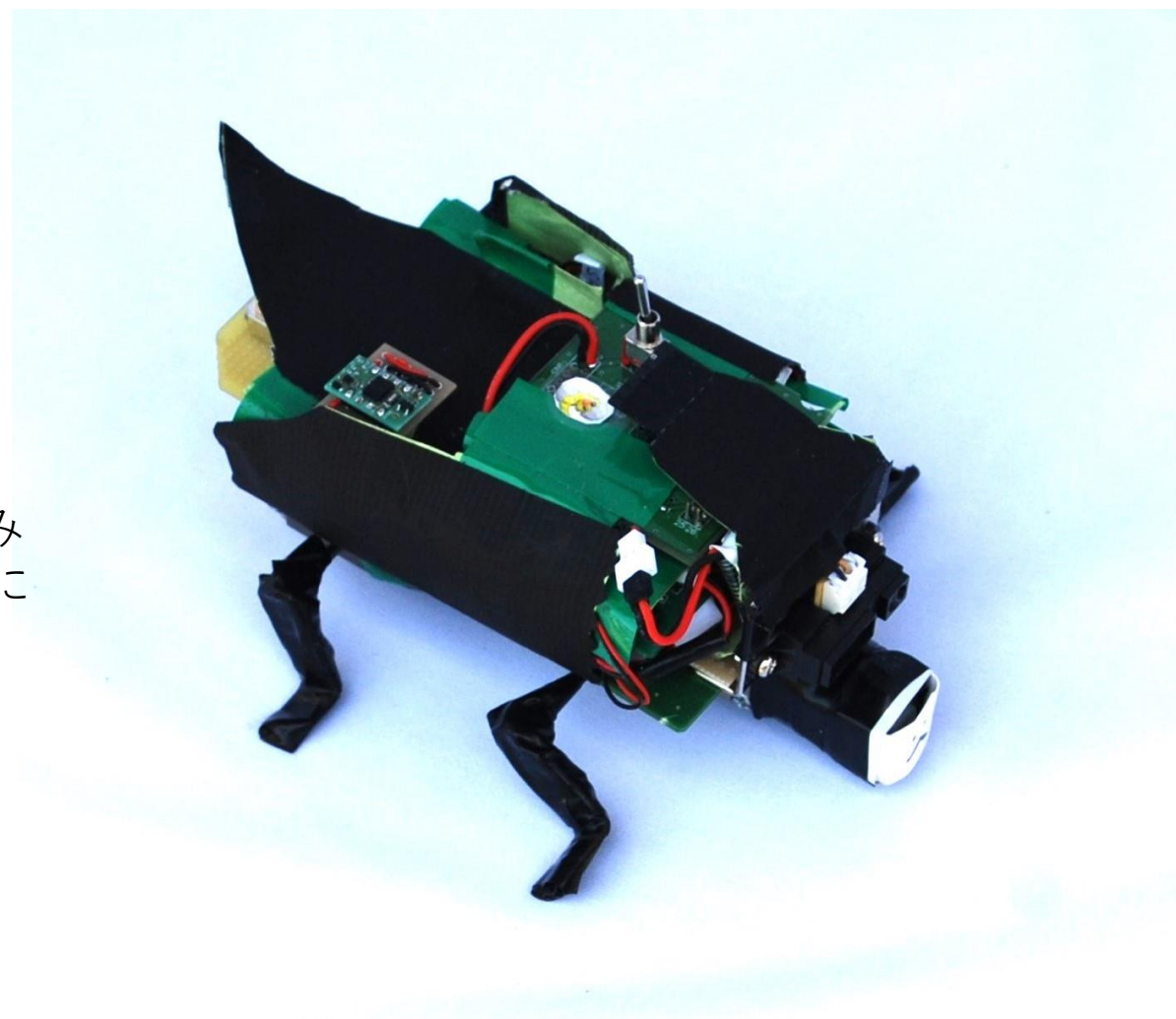


タミヤの3速クランクギヤーボックス
を使って歩く亀を作ってみました

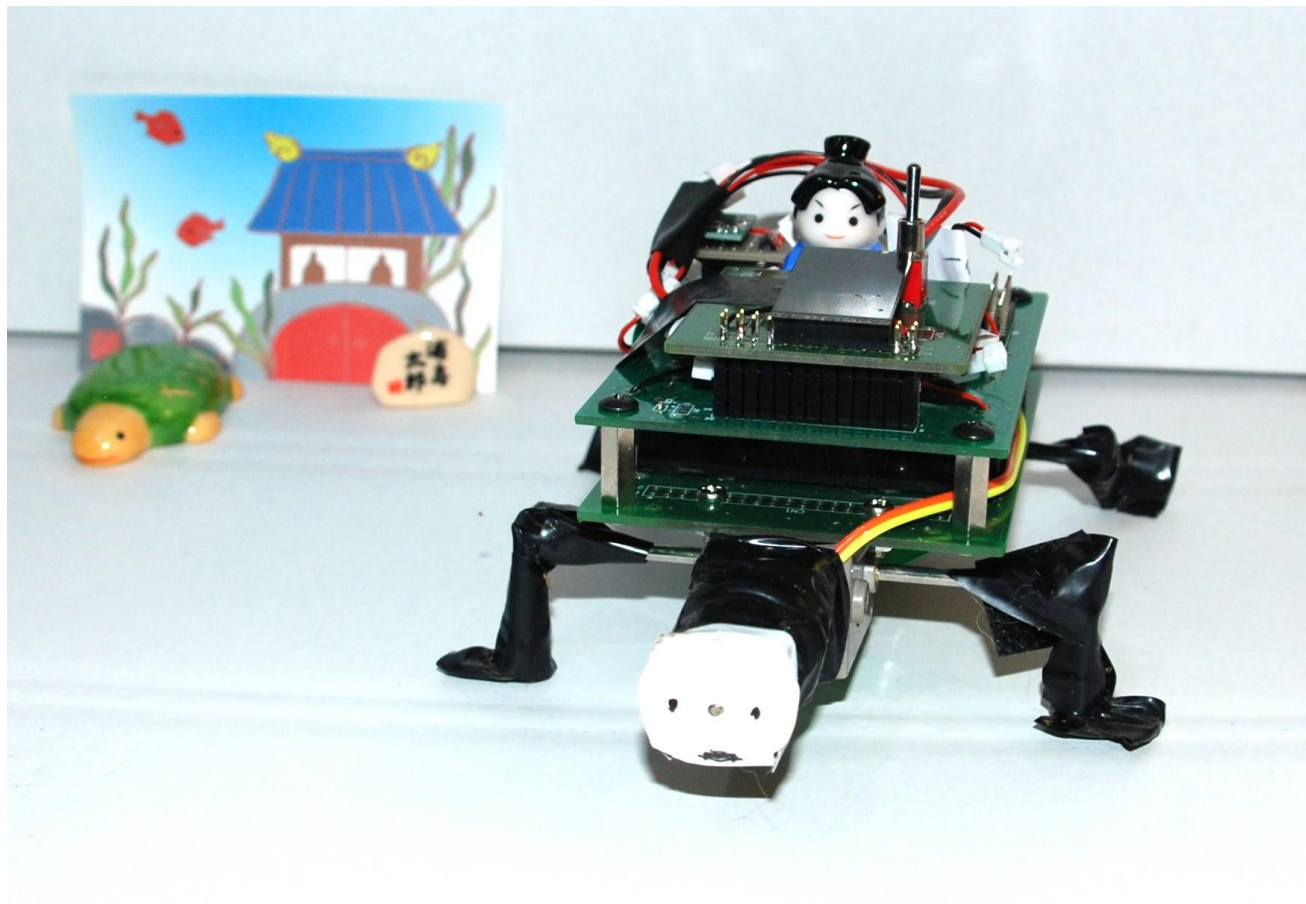
ギヤーボックス x 2
LPC1788 CPU 基板 x 1
TWELITE x 1
BNO055 ジャイロ
距離センサーを追加
GPS を追加
後ろ脚は左右独立で動きます。
(方向変更が可能です)



ほぼ完成
動きは亀に似ています。
障害物を検出して方向変更が可能です。
BNO055の航行で目的の方向とGPSの組み
合わせで目的の場所（指示された場所）に
移動可能です。



亀に似ているので浦島太郎を乗せてみました。



「うらしまたろう」の曲を作りましたので聞いてみてください。
Appyo.jp の音楽ページにあります。



まとめ

飛行に可能な重量が下がらない。

原因

モーターが重い、L型金具が重い、
プロペラ固定用アルミ線が重い

FA-130RA は回転数が少ないので揚力が上がらない。

飛びドローンは制御が困難なので、地上走行の車、
さらに、動作の遅い歩行ドローンに変更しました。