

要救助者探査ロボットの製作

岡 芳正 片沼 俊輔
清 裕翔 角田 陽太

1. 研究概要

要救助者探査ロボットとは、地震や水害などの災害で被災した人間を危険地域から発見することを目的として設計されたロボットである。

探査ロボットの種類として建物内部を捜索するロボットや瓦礫の上を走行するもの、空中から捜索し発見するものなどがある。今回制作したのは崩れた瓦礫の中を踏み分けて、被災者を探すための探査ロボットである。

2. 研究の具体的内容

(1) 通信機器について

今回の研究で使用する通信機器は TWELITE DIP である。この TWELITE DIP は TWELITE を 2.54 ピッチの基板に搭載した 28 ピン DIP 型で、この TWELITE DIP は手作業での配線が容易である。

汎用基板やソケット、ブレッドボードに直接搭載することができて、試作や小規模ロット生産に適しているため今回採用した(図 1)。

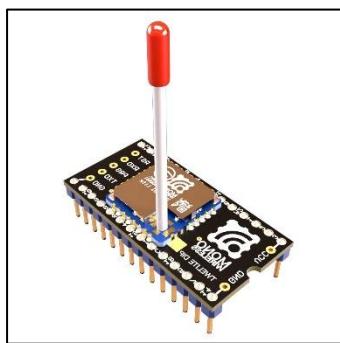


図 1 TWELITE DIP

(2) カメラについて

探査ロボットに乗せるカメラは、TapoC210 である。TapoC210 は TP-Link 社から販売さ

れているネットワーク Wi-Fi カメラであり、このカメラは解像度 2K で首が、水平 360° 垂直 114° 動き、また双方で会話が可能なため、被災地では被災者との会話ができる。被災者に場所を教えてもらい、被災者を安心させることができるようにしたため、今回採用した。(図 2)。



図 2 Tapo C210

(3) 車輪について

車輪は三角クローラを使用している。広い接地面積が持て、接地圧が小さいため、三角クローラを用いると地盤が悪くても、安定した移動をすることができるようになるため、三角クローラを採用した(図 3)。

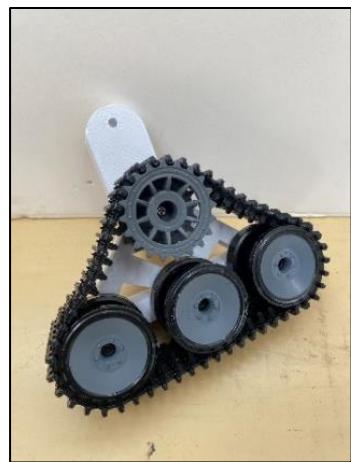


図 3 三角クローラ

(4) バッテリについて

バッテリは 12V の LiFePO4 バッテリを使用した。12V モータで車体を動かすには、12V の電源が必要になるためである（図 4）。



図 4 LiFePO4 バッテリ

(5) Arduinoについて

Arduino は Arduino Mega 2560 を使用している。Arduino Mega 2560 は AT Mega マイクロコントローラをベースにした開発ボードで、初心者向けの Arduino Uno とは違い、Mega はより複雑なプロジェクトに使用可能な設計である。3D やロボットベースのプロジェクトで使用することに適しており、接続可能な入出力ピンが Arduino Uno より多いため今回採用した（図 5）。

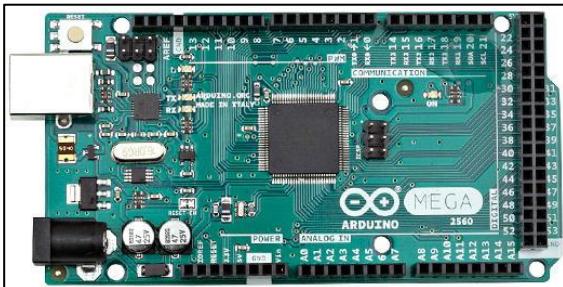


図 5 Arduino Mega 2560

(6) 固定具について

車輪や車体部品の固定には 3D プリンタで印刷したものや、レーザー加工機で加工したプラスチック板を使用し、少ない予算のなかでより軽く強度を持たせた（図 6）。

固定具は天板固定用と車軸側板固定用、車軸固定用の 3 つを作成した。

それぞれ仮止めから全ナットの取り付けの際に、固定する作業をしやすくしている。



図 6 固定具

(7) 車体の途中経過

固定具を使用して、上部板と下部板を接続した。車体のほとんどを木とアクリル板で構成している（図 7）。



図 7 車体

(8) ロボット完成予定図

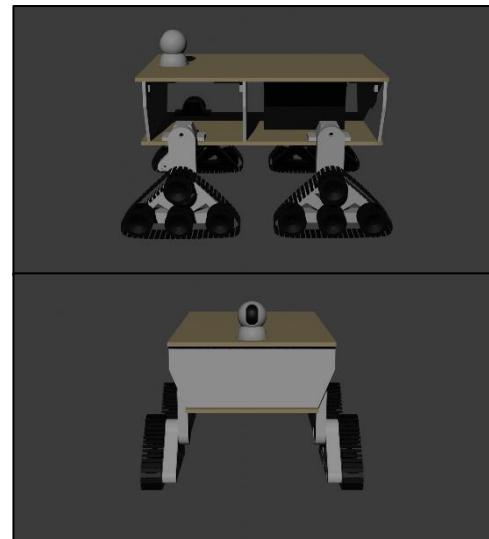


図 8 ロボットの 3D モデル

(9) 降圧回路の作成

今回、バッテリの 12V の電圧を TWELITE DIP の適正電圧 2.3~3.6V あたりに降圧するにあたり、まず 12V から 5V に降圧するため三端子レギュレータを使用した（図 9）。

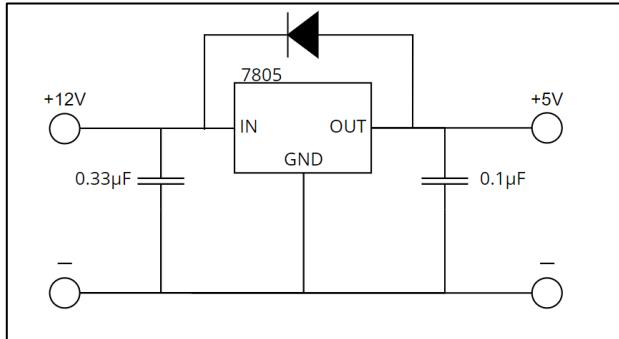


図 9 12V-5V 降圧回路図

三端子レギュレータは発熱するが今回の回路ではほとんど発熱しない。しかし、念には念を入れる為、ヒートシンクを取り付けて使用した。また、TWELITE の適正電圧帯に降圧するために、抵抗器を使用した降圧回路を作成した（図 10）。

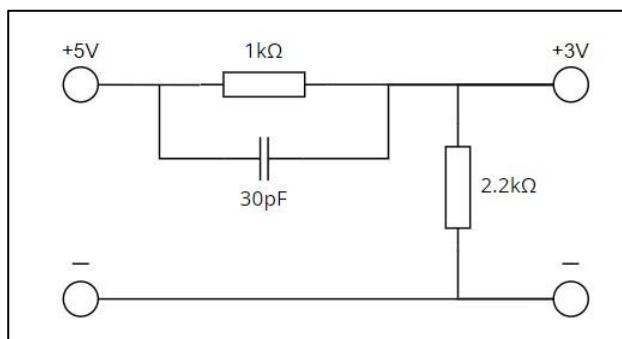


図 10 5V-3.5V 降圧回路図

図 9 の降圧回路と図 10 の降圧回路をつなげ、12V から 3.5V に降圧できる仕組みにした。12V から 3V への三端子レギュレータがあれば一つの回路で済んだが部品がなかったため、5V に降圧した後で 3.5V へ降圧した。三端子レギュレータを使用するのは、抵抗による降圧では、安定した電圧が取れないことや、抵抗が電流に耐えられず発火する恐れがあるためである。

(10) モータドライバ

三角クローラの 12V モータを回転させる為のモータドライバには、L298N を用いたモータドライバを使用する。

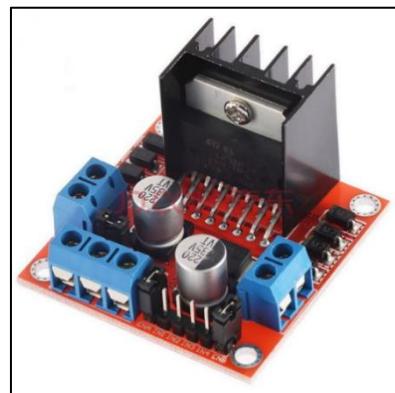


図 11 モータドライバ

入力電圧最大 12V のモータドライバで 2つのモータを制御でき、今回は接続を変え 4つのモータを制御する。Arduino から直接接続し制御することが可能である。

(11) コントローラの作成

ロボットを操作する時に必要なコントローラは、3D プリンタで制作した。頑丈にプリントし、通常の操作では壊れないようにした。

コントローラの内部に Arduino Mega を格納する。また、Arduino 電源用の 9V 角電池を Arduino の下部に取り付けられるようにした。

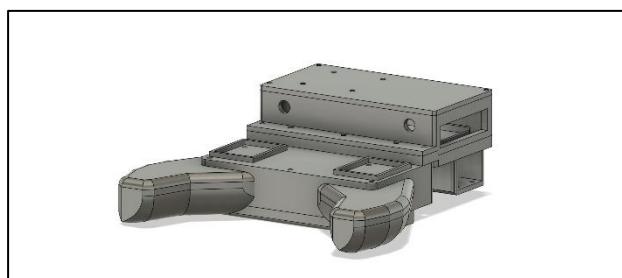


図 12 コントローラ

コントローラは Arduino Mega を内蔵するため、大まかに分解できるような構造になっている。接続や固定はネジで行い、持ち手、Arduino 格納ケース下、Arduino 格納ケース

上、Arduino 格納ケース蓋、ジョイスティックを付けた本体で構成している。図 13 はそれぞれのパーツを分けた図である。

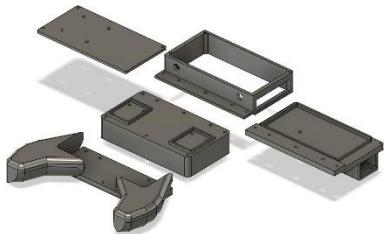


図 13 コントローラのパーツ分け

(12) ロボットのプログラムについて

以下がプログラム全文で、使用言語は Arduino 言語である。

<コントローラ側>

```
#define JS_LX A0
#define JS_RY A2
#define JS_RX A3
#define LMT_IN1 7
#define LMT_IN2 6
#define RMT_IN1 5
#define RMT_IN2 4

int JSR_Y = 0, JSL_Y = 0;

void setup() {
    //ジョイスティック
    pinMode(JS_LY, INPUT);
    pinMode(JS_RY, INPUT);
}

void loop() {
    JSL_Y = analogRead(JS_LY);
    JSR_Y = analogRead(JS_RY);

    //JSL
    if (JSL_Y <= 100) {
        digitalWrite(LMT_IN1, HIGH);
        digitalWrite(LMT_IN2, LOW);
    } else {
        digitalWrite(LMT_IN1, LOW);
        digitalWrite(LMT_IN2, LOW);
    }

    if (JSR_Y >= 923) {
        digitalWrite(LMT_IN1, LOW);
        digitalWrite(LMT_IN2, HIGH);
    } else {
        digitalWrite(LMT_IN1, LOW);
        digitalWrite(LMT_IN2, LOW);
    }

    //JSR
    if (JSR_Y <= 100) {
        digitalWrite(RMT_IN1, HIHG);
        digitalWrite(RMT_IN2, LOW);
    } else {
        digitalWrite(RMT_IN1, LOW);
        digitalWrite(RMT_IN2, LOW);
    }

    if (JSR_Y >= 923) {
        digitalWrite(RMT_IN1, LOW);
        digitalWrite(RMT_IN2, HIHG);
    } else {
        digitalWrite(LMT_IN1, LOW);
        digitalWrite(LMT_IN2, LOW);
    }
    delay(1);
}
```

```

<ロボット側>
#define JST1_F_PIN 11
#define JST1_B_PIN 10
#define JST2_B_PIN 9
#define JST2_F_PIN 2
#define MOTOR_R_IN1_PIN 7
#define MOTOR_R_IN2_PIN 6
#define MOTOR_L_IN1_PIN 1
#define MOTOR_L_IN2_PIN 4

void setup() {
    //ジョイスティック
    pinMode(JST1_F_PIN, INPUT);
    pinMode(JST1_B_PIN, INPUT);
    pinMode(JST2_F_PIN, INPUT);
    pinMode(JST2_B_PIN, INPUT);
    //モーター
    pinMode(MOTOR_R_IN1_PIN, OUTPUT);
    pinMode(MOTOR_R_IN2_PIN, OUTPUT);
    pinMode(MOTOR_L_IN1_PIN, OUTPUT);
    pinMode(MOTOR_L_IN2_PIN, OUTPUT);
}

void loop() {
    //forward
    if(digitalRead(JST1_F_PIN)==0) {
        digitalWrite(MOTOR_R_IN1_PIN, HIGH);
    } else{
        digitalWrite(MOTOR_R_IN1_PIN, LOW);
    }

    if(digitalRead(JST2_F_PIN)==0) {
        digitalWrite(MOTOR_L_IN1_PIN, HIGH);
    } else{
        digitalWrite(MOTOR_L_IN1_PIN, LOW);
    }
}

//Back
if(digitalRead(JST1_B_PIN)==0) {
    digitalWrite(MOTOR_R_IN2_PIN, HIGH);
} else{
    digitalWrite(MOTOR_R_IN2_PIN, LOW);
}

if(digitalRead(JST2_B_PIN)==0) {
    digitalWrite(MOTOR_L_IN2_PIN, HIGH);
} else{
    digitalWrite(MOTOR_L_IN2_PIN, LOW);
}
}

```

コントローラ側は、Arduino がジョイスティックの傾きを検知し、動作範囲に傾いたら、TWELITE DIP に信号を送り、ロボット側に送信するプログラムである。ロボット側は TWELITE DIP から受け取った信号を認識し、モータドライバに信号を送る。モータは左列、右列で動作し、ともに 2 つのモータを動作させる。

(13) ロボットの活動内容

今回製作したロボットは、探査を目的として製作しており、地震などの自然災害で出現した危険地帯への突入を行うためのものである。地震が発生し崩れそうなエリアなどへ最初から人を動員して捜索するのではなく。まずロボットを導入することで二次災害を防ぐことができる。

また、火山ガスなどが発生している可能性があるエリアで人が入る前に投入し、今回の製作では搭載していないが、ガス濃度センサを搭載した本機が危険エリアを浮かび上がらせる。

車体を大きく製作したので、狭い場所での作業は難しいが、その分多くの装置を搭載でき、汎用性を高めている。

3. 研究のまとめ

今回の研究では設計図を描かず進めたためにメンバー同士の意思疎通がとりづらくなってしまった。また、3Dプリンタで部品を印刷するのに時間がかかるてしまい、作業が滞ってしまった。メンバー同士で進捗の状況が分からぬなど情報の伝達が不十分であった。これらを改善するには、リーダーがメンバーに対してわかりやすいように指示し、予定を細かく決めることが重要だと考えた。

ロボットが全体的に大きく、災害現場の制限された場所では、行動しにくいと感じた。最初のコンセプトである、「瓦礫を乗り越えて進む」という考えがあまり現実的ではなかった。」というのも反省点である。

また、当初の金属製・防水であるということも出来なかった。

最後に、今回の課題研究で制作したロボットは、実際の災害現場では役割を果たすのは難しいと感じた。しかし、ロボットを小型化し、アームなどの機能を追加すれば、実際の現場でも活躍できると感じた。

参考文献

降圧回路

<https://voltechno.com/blog/creverse-5v3v/>

TWELITE DIP

<https://mono-wireless.com/jp/products/twelite-dip/index.html>

回路図制作

<https://online.visual-paradigm.com/ja/>