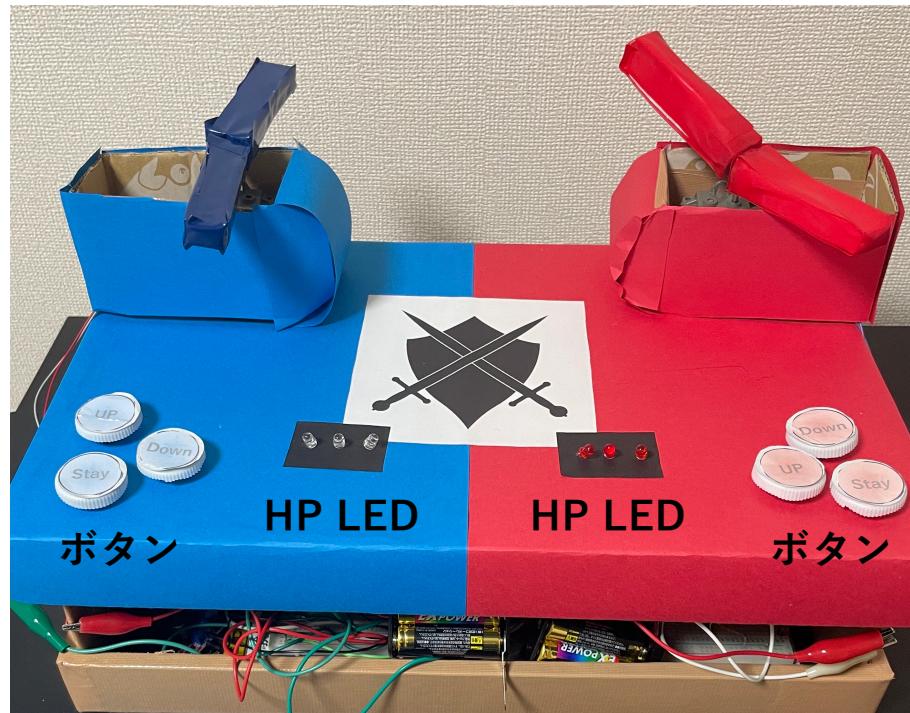


A · 2班

## ゲーム設定

2P ヘリコプター



1P ヘリコプター

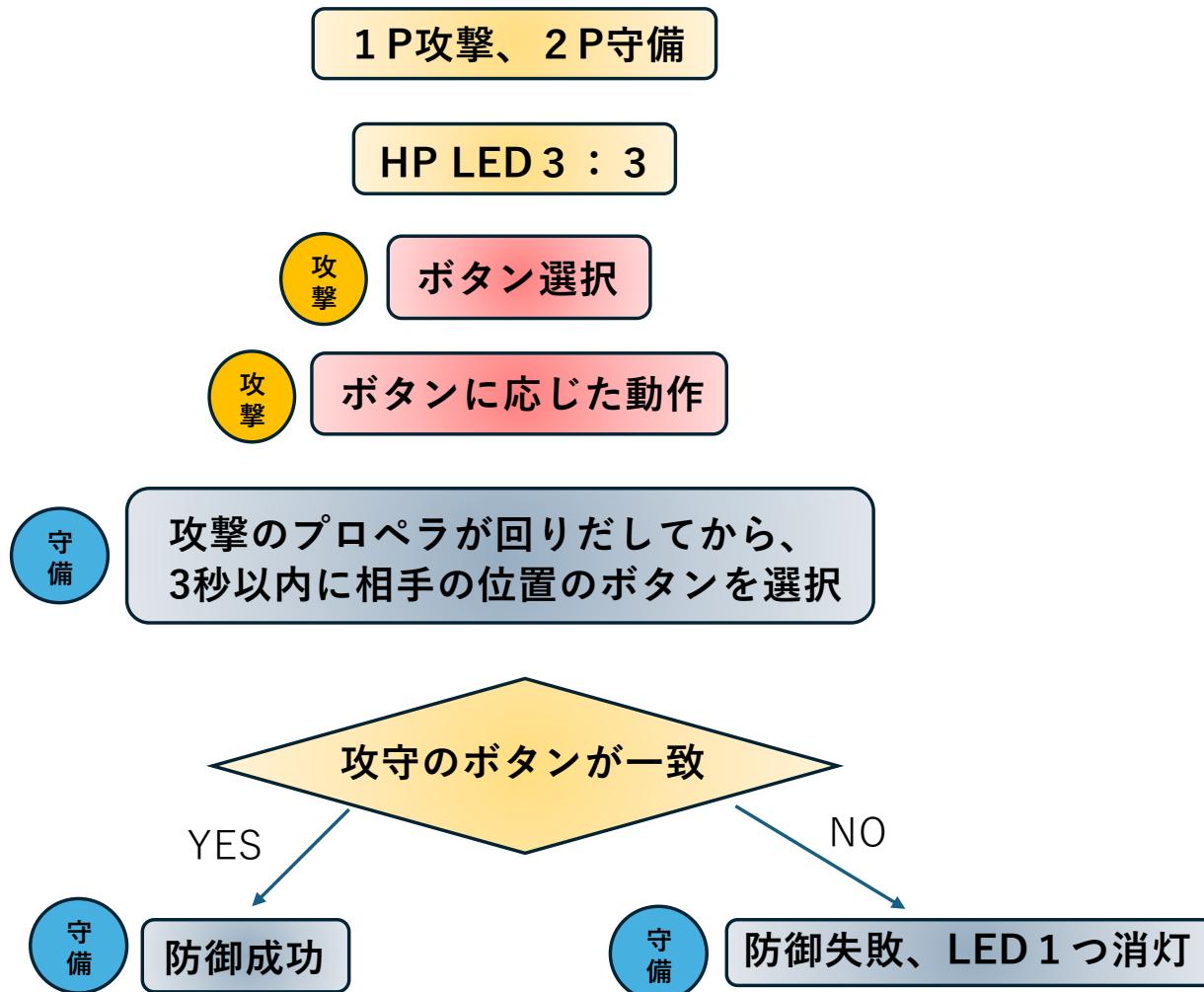
### ヘリコプター

- ・上下方向に攻撃
- ・相手の位置が分かれば迎撃できる
- ・プロペラ  
時計回り：上昇  
反時計回り：下降

### ボタン

- UP：上昇
- Stay：留まる
- Down：下降

## ゲームの流れ



## ヘリコプター

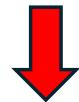
- ・上下方向に攻撃
- ・相手の位置が分かれれば迎撃できる
- ・プロペラ  
時計回り：上昇  
反時計回り：下降

## ボタン

- UP：上昇
- Stay：留まる
- Down：下降

## ゲームの流れ

攻守交代



相手のHP LEDを 0 にした方が勝ち

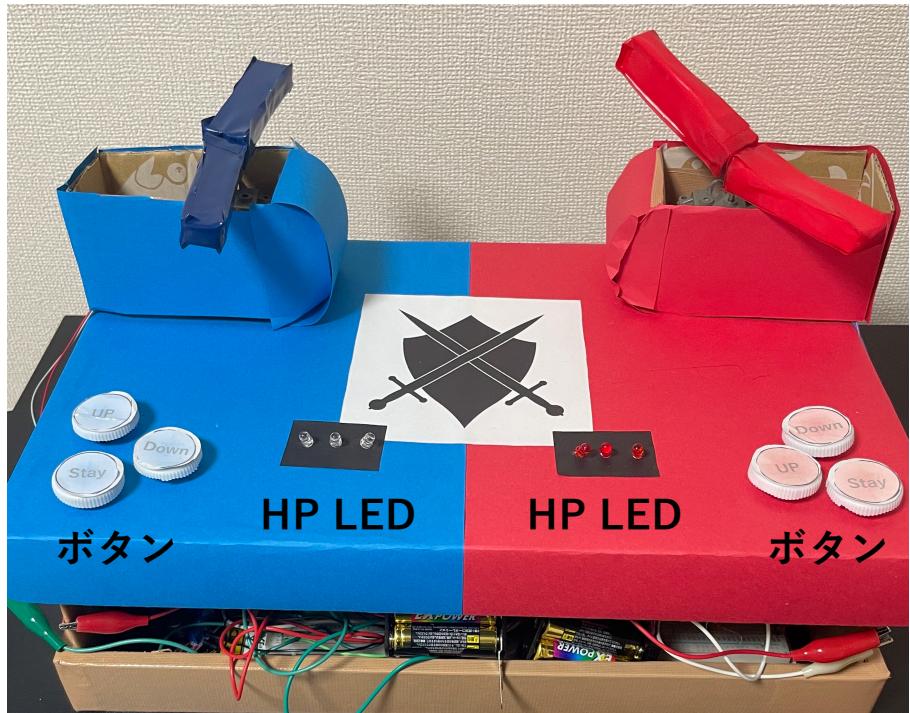


目標

モータの動きから押すべきボタンを判断する  
直感的で分かりやすいゲーム

## 動作原理

2P ヘリコプター



1P ヘリコプター

プロペラを直流モータによって回転

モータドライバー

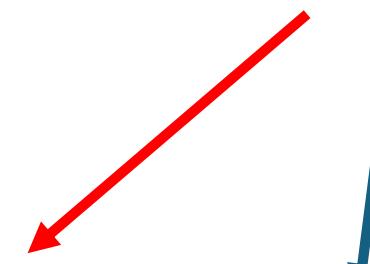
00:ストップ

10:正転

01:逆転

11:ブレーキ

$\text{PORTD}=0b\textcolor{red}{1000}\textcolor{blue}{01}\textcolor{green}{11};$

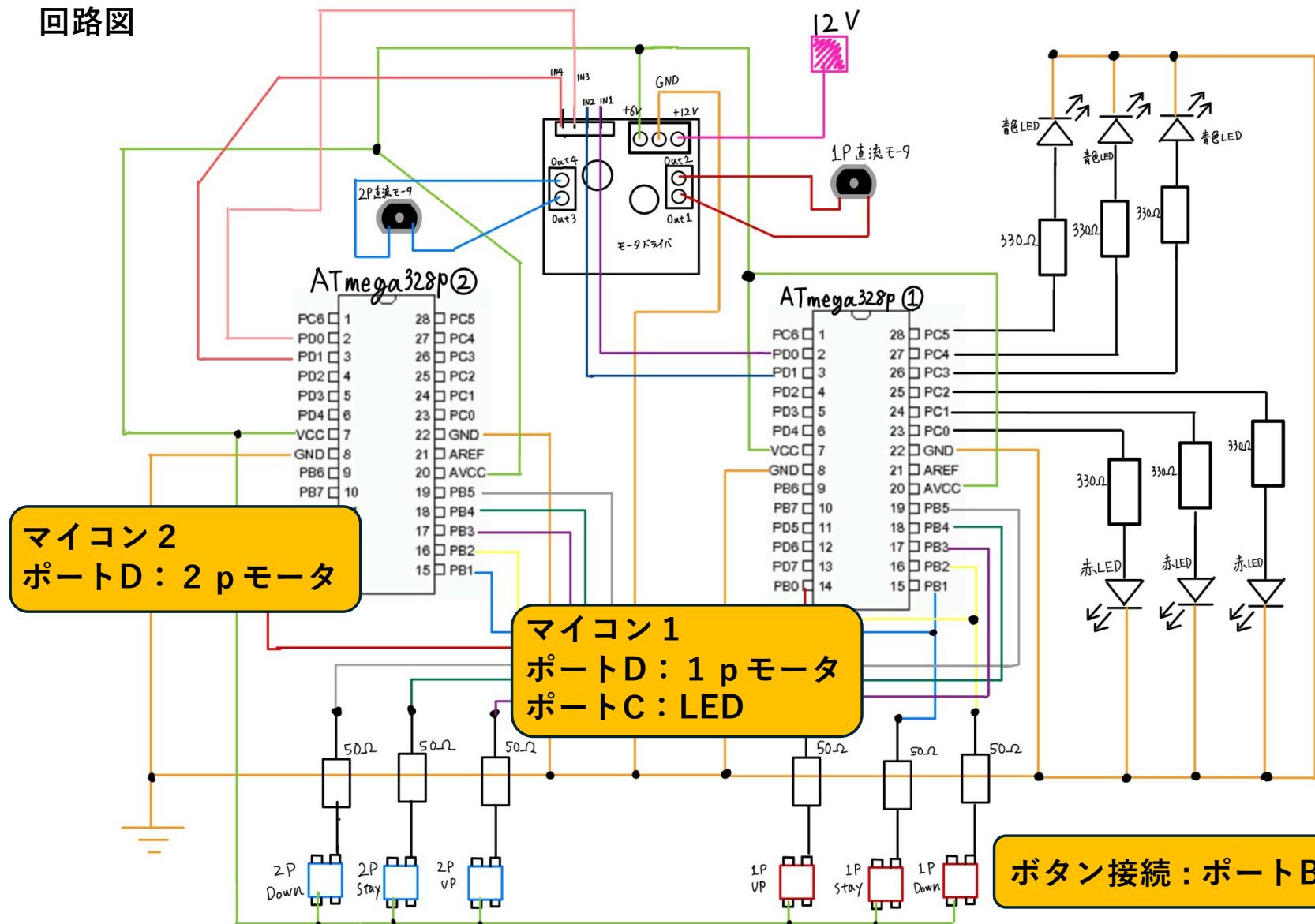


ピン6-7に繋いだ  
モータが正転



ピン2-3に繋いだ  
モータが逆転

## 回路図



## 部品リスト

### ・回路

1. ATmega328p × 2
2. モータードライバー(L298NDC) × 1
3. 直流モーター(FE130AV-2270-38-R) × 2
4. LED: 青 × 3
5. LED: 赤 × 3
6. 抵抗 (330Ω) × 6
7. 抵抗 (50Ω) × 6
8. ボタン × 6
9. 単三電池 × 4
10. ブレッドボード
11. ワニ口クリップ
12. ジャンパー線

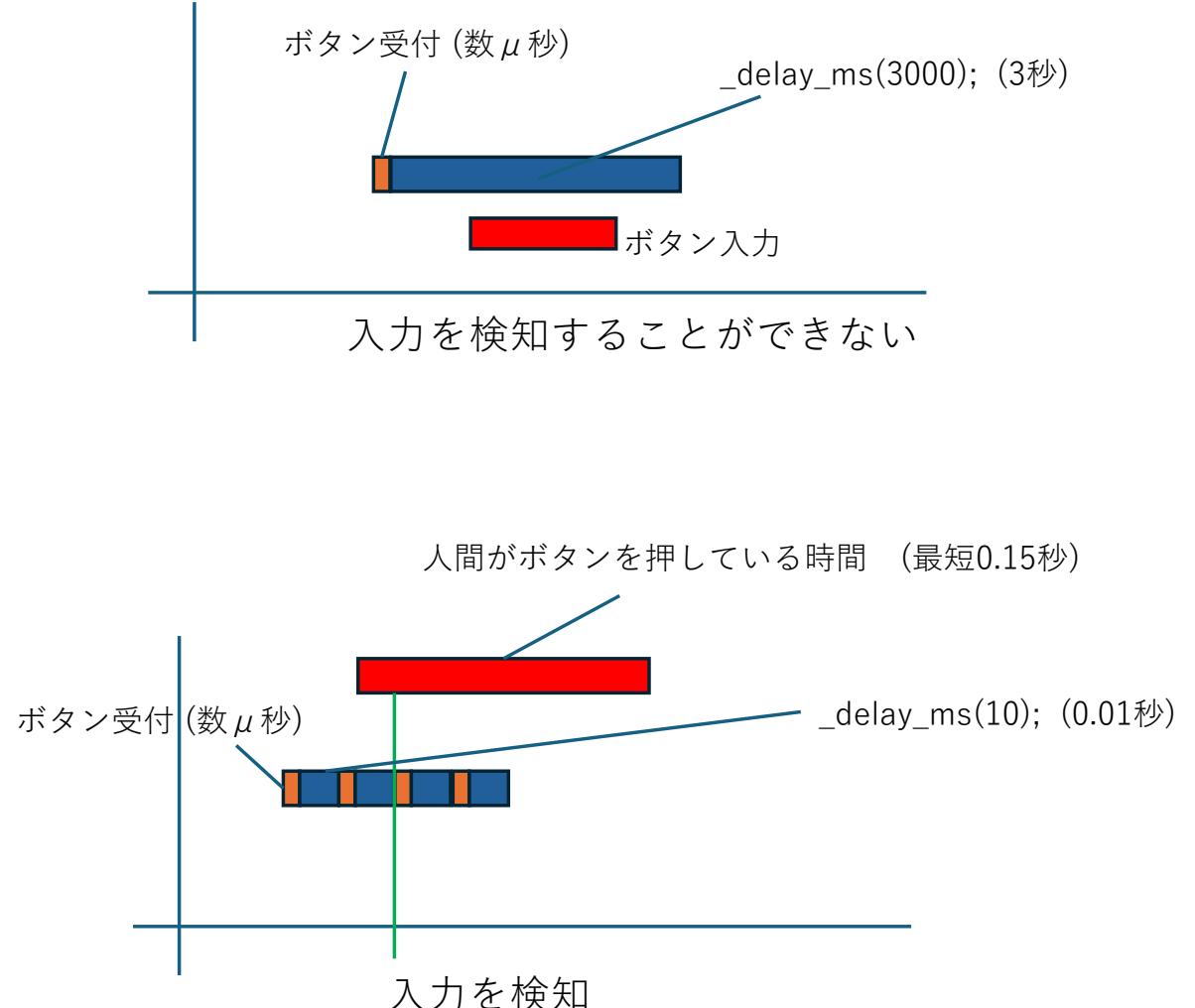
### ・デザイン

1. マスキングテープ: 赤
2. マスキングテープ: 青
3. 色画用紙: 赤
4. 色画用紙: 青
5. 折り紙: 黒
6. 発泡スチロール
7. 草子箱
8. ダンボール
9. マグネット

## 苦労した点

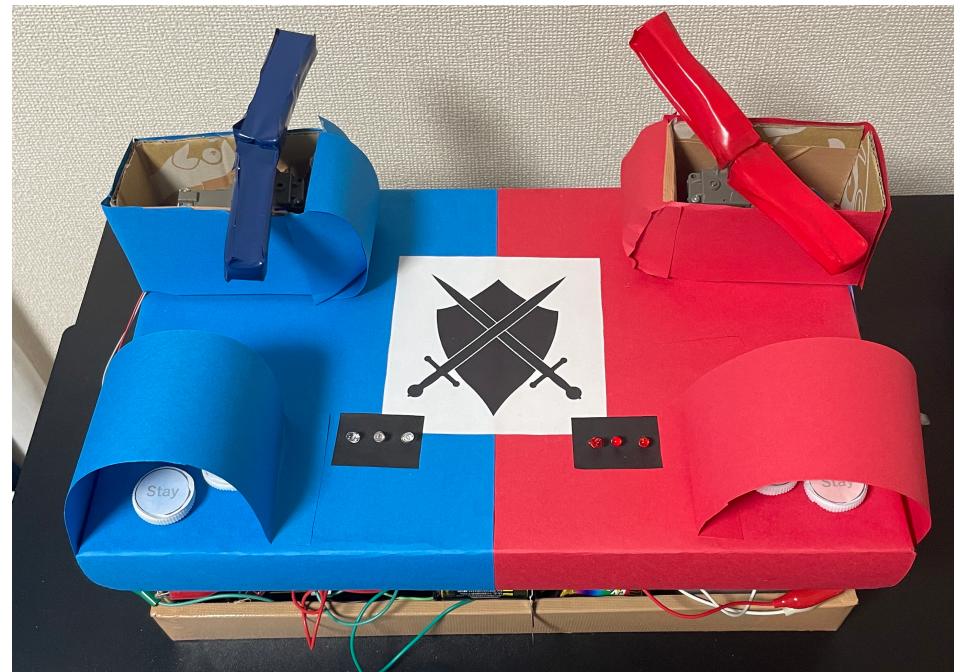
### ボタン入力に受付時間を設ける

```
//ボタン関数
int button1(int t)
{
    int q=0;
    int t_1=0;
    while(t_1<t)
    {
        if(bit_is_set(PINB,PB0))
        {
            q=1;
            break;
        }
        else if(bit_is_set(PINB,PB1))
        {
            q=2;
            break;
        }
        else if(bit_is_set(PINB,PB2))
        {
            q=3;
            break;
        }
        _delay_ms(10);
        t_1 = t_1+10;
    }
    return q;
}
```



## アピールポイント

- ・独自のゲーム性
- ・ボタン入力に受付時間がある。
- ・連打、長押し、同時押しされても、正常に動作。
- ・回路をいつでも取り出せるデザイン
- ・LEDによって判定結果がわかる。
- ・モータの動きから押すべきボタンを判断する。
- ・誰でも直感的に遊ぶことができる。

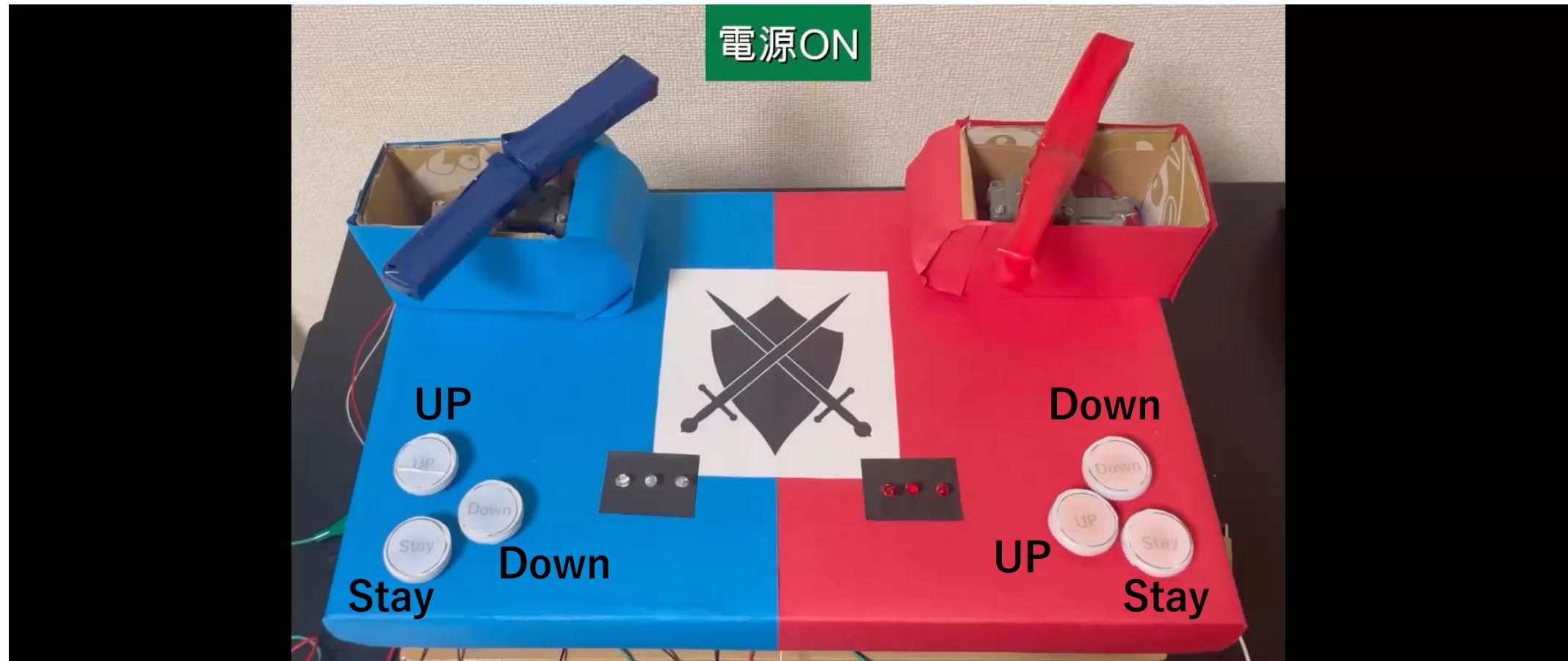


## プレイ動画

攻撃側のプロペラが回転、UP：時計回り、Stay：回転しない、Down:反時計回り

防御側は攻撃の回転を見て3秒以内にボタン選択

一致：防御成功、不一致：防御失敗 LED消灯



※時間の都合上、同じ動作の部分は早送りしております。