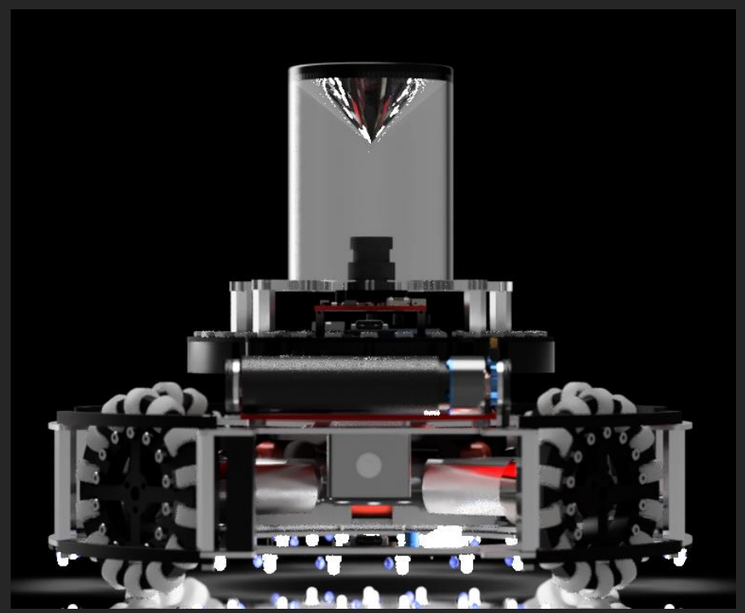
ARTEMIS
MUNAKATA.E.P.C


## チームメンバー
松田 魁琉 / 目野 優輝 / 石原 廉太郎 / 熊抱 崚太

## 所属ブロック
九州ブロック 北九州ノード

@munako_artemis


# ロボットの概要 – 2024 Season Munako Artemis' Robot



## ➢ 搭載している部品

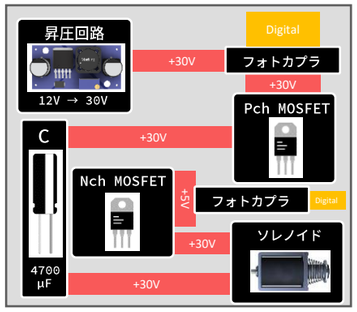
- **バッテリー** : KyPom 3Cell 1300mA
- **モータードライバ** : DSR-1202
- **モーター** : IG22 19:1 x 4
  → 応答性の高さから採用しました。
- **メインマイコン** : Teensy 4.0 x 1
  → シリアル用ピンの多さから採用しました。
- **サブマイコン** : Seeeduino Xiao x 3
  → ジャイロセンサ・ラインセンサ・ボールセンサの処理用。
- **ボールセンサ** : TSSP58038 x 8
  → ボールから発せられる赤外線を読み取り、サブマイコンで計算を行ってボールの角度を距離を算出しています。
- **ラインセンサ** : S4282-51 x 16
  → 光変調フォトICを使用し、白線の判別をしています。
  ベクトルの考えを用いて白線の角度を算出しています。
- **ジャイロセンサ** : L3GD20H x 1
  → 主にロボットの姿勢を制御するために用いています。
- **カメラ** : OpenMV H7 x 1
  → カメラ導入コストの低さから採用しました。
  今回の機体では全方位ミラーを用いて360度の方向を監視しています。
- **昇圧回路** : XL6009
  → ソレノイド駆動電源の生成で使用しています。(12V → 30V)
- **ソレノイド** : CB1037
  → 所謂「キッカー」という機構で用いています。
  ボールをキックし推進力を与えることができます。
- **ドリブラー用BLDC**
  → AliExpressで購入した比較的小さいBLDCです。
  ボールを自ロボット側に回転させることでボールを保持します。
  主に安定したキックをするために使用しています。

## ドリブルとキックで「魅」せる
### - ドリブラーとキッカーの搭載 -

2024シーズンの私たちの新しい取り組みとして、「ドリブラー」「キッカー」の搭載を行いました。ドリブラーは主にボールの保持、キッカーはボールのキックという役割を果たします。

### キッカー

キッカーの仕様は右図の通りです。
電源回路で生成した12Vを昇圧回路で30Vに昇圧しています。また、
コンデンサへの充電部分、ソレノイド駆動部分をFETによって分断することでマイコンへのノイズ対策をしています。
Digital(黄色部分)はメインマイコンに接続されており、フォトカプラに3.3Vの電圧をかけることでそれぞれの信号を出力させています。
キッカーによりゴール前におけるプッシング発生率を大幅に減少させることができます。




### ドリブラー

ドリブラーにはAliExpressで購入したBLDCを用いています。3Dプリンタで制作した回転体にゴムチューブを取り付け、それをBLDCを用いて回転させています。
そのままだとドリブラーがボールの高さより上に上がってしまい、ボールをうまく回転させることができなかったため、回転部分と固定部分にバネの両端を取り付けました。そうすることによって、
回転部分が回転した時にバネが捻られ、
元に戻そうとする力が働くことで結果的にドリブラーがボールをを押さえつけるという機構を実装することができました。
今大会では、ドリブラーを用いて、
ボールをゴールまで運んだり、
キックを安定させるための機構として
使用しています。

## 円形で**全方位**を見る
### - 円形配置による全方位の監視 -

今シーズンのロボットにはセンサを円形に配置したセンサーを多く見ることができます。

### ボールセンサ

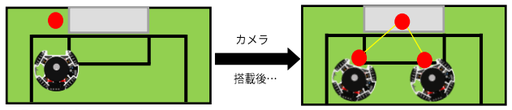
ボールセンサ(TSSP58038)を円形上に8個配置しています。ボールの角度・距離を取得するために、まずは一番反応しているセンサーを取得し、次にその周囲3つのセンサーのデータを元にベクトル計算を行うことで1度単位での正確な位置取得が可能になりました。

### ラインセンサ

光変調フォトIC(S4282-51)を円形に16個配置し、2個ずつに分割してサブマイコンで処理をしています。各センサーの値を見て白線の角度を算出し、前回の白線の角度と照らし合わせることで、「ロボットが白線を踏んでいる」「ロボットが白線を超えた」という状況を判別することができるようになりました。

### カメラ

正確には「カメラに用いる全方位ミラー」が円形となっています。
カメラの上部に円錐状のミラーを設置することで、全方向のコートの状況を正確に把握することができます。
自チームでは、敵ゴール、自ゴールの二つを色認識で取得し、
三角関数などを用いて角度・距離を算出しています。
これらのデータで、ロボットが敵ゴールを向きながらキックしたり、
自ゴールに戻りキーパーとして動いたりすることが可能になります。
特に敵ゴールを向きながらのシュートは強力で、中立点からのボールも正確にシュートすることができます。




何もないと入らないこのシュートも…

カメラでゴールの角度を計算することで
ゴールに向かってシュートすることができる

## 白線外など**言語道断**
### - プログラムによる速度調節 -

昨年度から白線外の空間が縮小し、白線外に出ないような
工夫が必要になりました。すべての速度を落とすという対策もありますが、これだと全体的なスピードを失うことになります。
そこで、ロボットの移動する角度を縦、横に分解し、その方向にどのくらいの時間移動したかをタイマーで取得することで、
速度の変化の割合を変化させるという処理を実装しました。
これによって、コート内では素早く動き、白線付近では比較的スローな移動にすることができました。

```
100     //ある角度に一定時間移動していた時は速度を落とす。
101     if(cos(radians(deg)) <= cos(radians(150)))
102     {
103         LimitOfBackMove.start();
104         if(LimitOfBackMove.get_value() >=500)
105         {
106             if(LimitOfBackMove.get_value() >= 2300)
107             {
108                 LimitOfBackMove.stop();
109                 LimitOfBackMove.reset();
110             }
111             else
112             {
113                 if(Ball_Distance <= 90)
114                 {
115                     move_speed = move_speed * 0.85;
116                 }
117             }
118         }
119     }
```

実際のプログラム
LimitOfBackMove というタイマー変数によってある方向に対する移動時間の検知を行い、速度の割合を減少させている(115行目)。

## **金属製**でロボットを強固に
### - CNCによる金属加工 -

モーターの変更によりタイヤを強固に固定する必要がありました。初めは3Dプリンタを用いて製作していましたが、部品が折れるなどの問題が頻発したため、OriginalMind社のCNCを用いることで、時間のかかる金属加工を自動で行い、タイヤ・タイヤ固定具を強固に、そして安定させることができました。

(右図：ロボットに取り付けた金属部品)

## 強いロボットは**環境**から
### - 私たちの開発環境 -

私たちはロボットの開発のために以下のツールを使用しています。
環境を統一して活動することでロボットの開発を円滑に進めることができます。

<ハード>
ロボット設計 ： Fusion360
基板設計 ： KiCad 6.0

<ソフト>
プログラム制作：Visual Studio Code
Arduino IDE

<データ共有>
Githubを用いてチーム内で
リアルタイム共有を行っています。

<コミュニケーション>
Discord、Asanaを用いてタスクや進捗をチーム内で共有しています。

## スポンサー

私たちが制作するロボットにはJLCPCB様の基板を使用しています。
JLCPCB様にスポンサーになっていただき、格安で基板製造をしていただいております。また、JLCPCB様とのタイアップ記事も制作しているのでぜひ
JLCPCB及び私たちのウェブサイトをご覧ください。

JLCPCB　ARTEMIS MUNAKATA.E.P.C