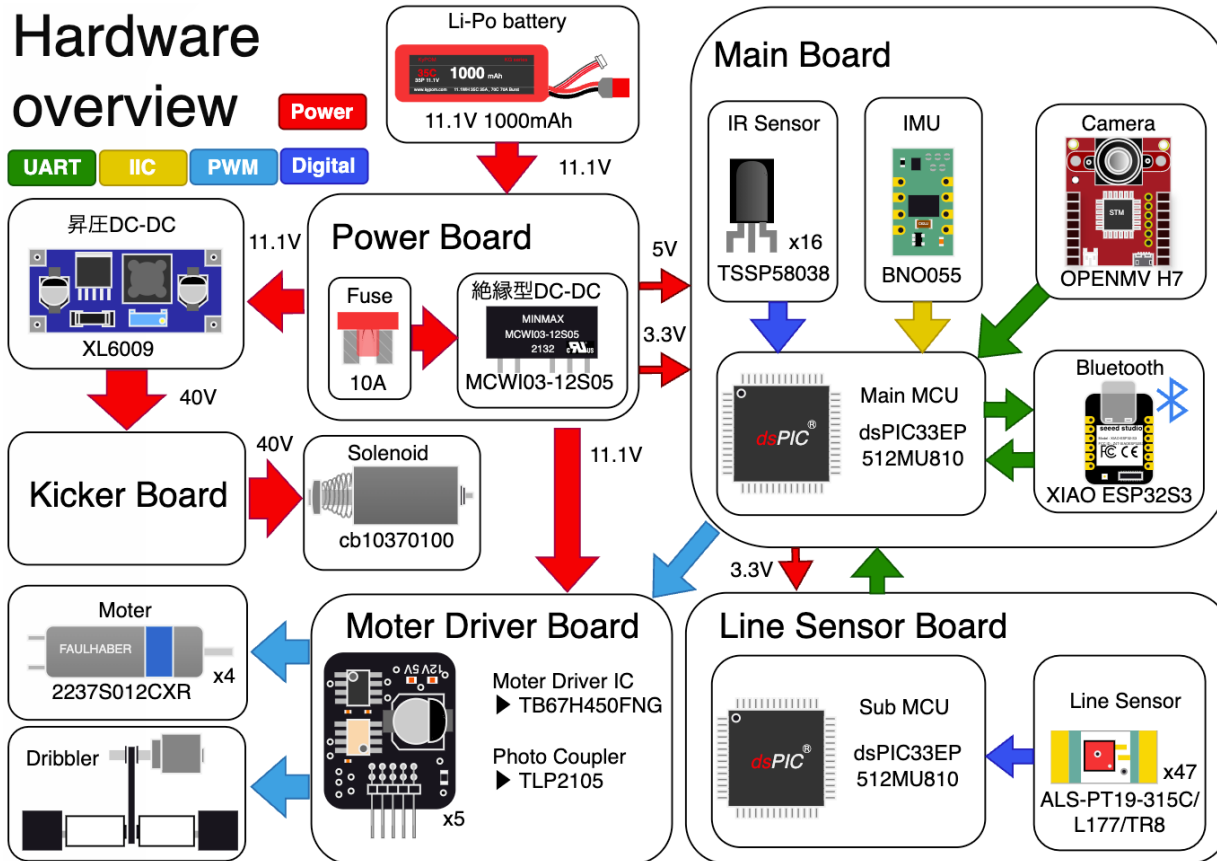


3Dデータ

## Hardware overview



## チーム活動・開発環境

私たちロボマッスルは、東京都立産業技術高等専門学校のロボカップ研究部に所属しており、RCサッカーライトウェイトに出場するため、2年生3人、4年生1人で活動している。

【ロボット設計】 Autodesk Fusion



主に機体と回路の設計に使用  
機械部品と回路の干渉をアセンブリで確認できるため、ロボットを組み立てる前から、物理的なミスに気づくことができる  
設計データは、オンラインのワークスペースに保存されるため、他のメンバーが常に最新のデータを確認することができる

【チーム連絡】 Discord



主にチーム連絡やタスクの管理に使用  
過去の投稿が残るため共有した内容の保管ができる

また、フォーラムという機能では小さなトピックごとに情報を分けられるので、必要な情報をすぐに得られる

【機体制御】 MPLAB X IDE



主に機体のプログラムを書くのに使用  
メインで使用しているMicrochip Technologyが出しているIDE  
コンフィギュレーターでコードを生成できたり、COMポートのデータ確認機能でビジュアルデバッグができたりするため便利

【データ公開】 Git Hub

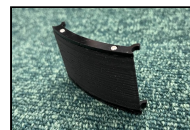


主に機体のコードを公開するために使用  
バージョン管理ができるため過去の変更点などを確認できる

また、私たちは他チームとの交流や自分たちの技術の交換は必要不可欠だと考えている。そのため、開発状況やロボットの紹介などをブログやX(旧Twitter)などを通して発信している。

## 工具不要のカバー

側面のカバーパーツは整備性を考え、工具等を使用せず手で着脱ができるようにした。仕組みとしては、丸型のスペーサーに側面パーツの爪を引っかけることで本体と側面パーツを固定しており、ねじ等は接続に使用していない。また、爪の長さなどを工夫することで機体同士の接触による衝撃では外れないが、外すときには手のみで容易に外すことができる。



外装パーツ

## 足回り

駆動用のモーターにはFAULHABERの2237S012CXR、ギアボックスは14:1を使用している。これによりmaxonモーターの118715番のものと比較した場合回転数では劣ってしまうが、トルクが高いため機体同士で押し合いになった場合押し返すことができる。また、オムニホイールをφ53の大型なものにすることにより、回転数が低いことをカバーしている。



自作オムニホイール

## 機体の材質

軽量化を考え、3Dプリンターで製作したABS製の部品や塩化ビニル、POMでできている比較的軽量の部品を使用しているが、強度が必要な箇所はアルミを使用し上部に横載するものを支えている。これにより、重量を抑えつつ強固な機体を作ることができた。  
アルミ製の部品にはアルマイト加工を施したものを使用し、回路が短絡することを防いでいる。

## PICマイコンの使用

私たちのチームでは、複数のセンサーやモーターの制御、基板間通信を一つのマイコンで同時に実現するために、4つのUARTと16個のPWMなどの多様なペリフェラルを備えたPICマイコン「dsPIC33EP512MU810」を使用している。このマイコンは70MIPSの処理能力を持ち、複雑な演算を高速で行えるほか、DSP機能による高速な信号処理が可能です。また、PPS (Peripheral Pin Select) 機能を活用することで、基板設計時の配線ミスなどにも柔軟に対応できる。

他のマイコンと比較してみると、Arduinoはシンプルな制御に適しているが、高度な演算処理には向きません。Raspberry Piは画像処理や機械学習などに優れますが、リアルタイム性が求められる制御には適していません。STMマイコンは32bit MCUのARM Cortex-Mコアを搭載し、複雑なデータ処理やリアルタイム制御に優れ、高度な処理性能を必要とする場面で非常に有効だ。

しかし、これらのマイコンを採用しなかったのは、私たちは開発環境が整い、経験豊富な先輩方からのご指導が得られるからだ。これにより、スムーズな開発が可能となっている。

## 安定動作する回路を目指して

昨シーズンの機体は回路の動作があまり安定せず機体が誤作動を起こすことが頻発した。そのため回路の安定性を向上させ、安定して動くロボットを作るために以下のことを行った。

### ノイズを低減してマイコンを安定動作

- ・モータードライバなどから発生するノイズを、絶縁型DCDCコンバータとフォトカプラを使用しGNDを絶縁することで遮断
- ・マイコンの電源ピンにバイパスコンデンサを接続することでノイズを除去

### 故障時の対応を迅速に

- ・モータードライバやマイコンなど故障する可能性のあるICを一部ピンソケットで接続することで故障時に故障したICのみを取り外して換装できるようにした
- ・ケーブルコネクタの接触不良で回路が動かないなどの事故が頻発していたため、回路に電源が入力されているかを可視化する照度の低いLEDを搭載しコネクタの接触不良を可視化

## ブロック大会からの改良

### レーザーを使用したホールドセンサー

ブロック大会では、ホールドセンサーの発光部に砲弾型LEDを使用していた。しかし、LEDの光が拡散しやすいため、車検を一度で通過することができず、大会当日にタイムロスが生じてしまった。

そのため、全国大会の機体ではレーザーを使用することにした。レーザーの光はLEDの光に比べて直進性が高く、機体から光が漏れなくなる。これにより、ブロック大会の時よりもロボットがさらに多様な動きができるようになった。

### ドリブラーの搭載

小型で高速に回るモータで、グリップしやすいシリコンでできたローラーを回して、ボールにバックスピンをかけ、機体がボールを保持できるようにした。これにより、ブロック大会の時よりもロボットがさらに多様な動きができるようになった。

## ライン検出を高速に

今回ライン検出の処理を高速化するため、使用するモジュールをADコンバータからコンパレータに変更した。

逐次比較型ADコンバータではセンサー情報を一つずつしか変換できず処理時間が長かった。しかし、コンパレータでは全てのセンサー情報を並列処理できる。

これにより変換時間を20μsから1.3μsに短縮することに成功した。

## 回り込み処理の効率化

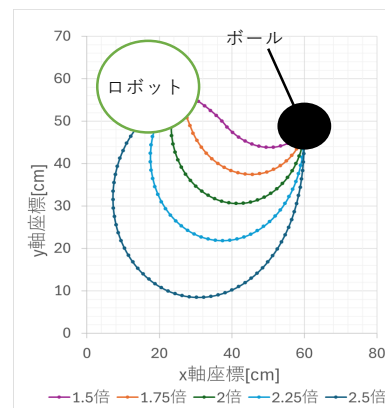
ロボットの移動方向を決める処理では高い制御周期を維持するため、効率的かつ理想的な軌跡を描き、ボールの後ろに回り込む方法を工夫する必要がある。そこで、以下の計算式を回り込み処理として実装した。

$$\text{進む方向} = n \times \text{ボールのある方向} \\ N: \text{任意の定数}$$

この計算式はロボットの正面を0°、右回りを-の方向、左回りを+の方向とし、ボールがある角度に一定の値を掛けた方向に移動させることで、自然な弧を描きながらボールの後方に回り込む動作を実現した。この計算をExcel上でシミュレーションした結果、nの値を1.75〜2の間にするとちょうどよい軌跡を描くことが分かった。



ラインセンサ基板の写真



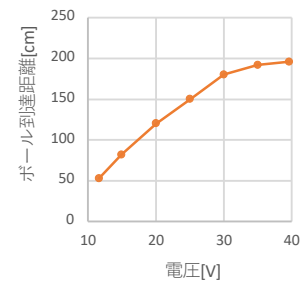
ゲインを変更したときのロボットの動きをシミュレーションした結果

## 高威力なキッカー

重量とサイズの制限を考えるとあまり大型のソレノイドを搭載することはできないが、なるべく高出力のキッカーを搭載したい。

XL6009昇圧DCDCコンバータモジュールを用い、ソレノイドに入力する電圧をバッテリー電圧の11.1Vから40Vに昇圧した。

これにより小型軽量のソレノイドでボールの到達距離を53cmから194cmに約4倍程度向上させることができた。



ボール到達距離の計測実験結果

## 敵避け精度の向上

これまでは、ゴールの色の塊をもとにシュート位置を決定していた。

具体的には、最も大きな色の塊の中心をシュートするエリアの一部として認識していた。しかし、背の低い敵がいると、ゴールの上部で色が繋がりがやすくなり、その部分がシュートするエリアに含まれてしまうことがあった。その結果、敵がいる場所もシュートできるエリアとして認識されてしまっていた。

この問題を解決するために、ゴール前に敵がいなかったかを確認する処理を追加した。具体的には、ゴール内の複数の位置を調べ、それぞれの色がゴールの色に近いかどうかを判断する。ゴールの色に近い場合、その位置をシュート可能な場所として認識し、ゴールの色でない場合はそこに敵がいると判断する。そして、敵の位置を避けて、シュートするエリアを再決定する。これにより敵避けの精度が向上し、得点率が向上した。

