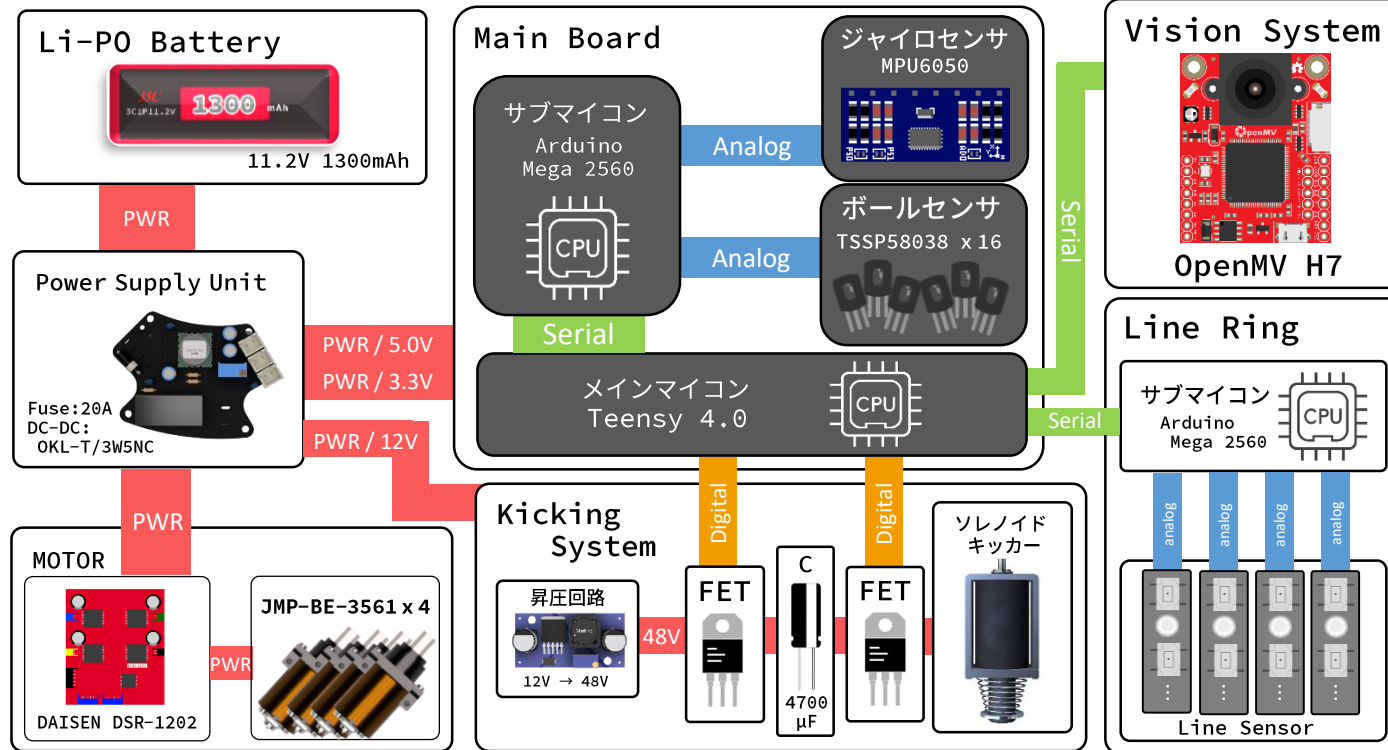
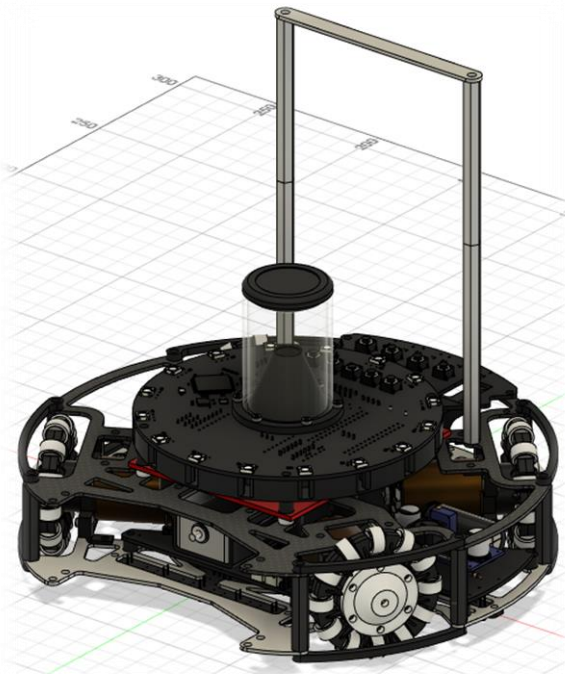




ロボットの概要



メインマイコン: Teensy 4.0

モーター制御や画面表示などの、ロボット制御を行っています。以前まではTJ3B Loaderを使用していたが、入出力ピン数の多さや、処理性能の高さからこちらを採用しました。



ボールセンサ: TSSP58038

ボールセンサにはTSSP58038を一機当たり16個搭載しています。ボールから発せられる赤外線を読み取り、サブマイコンで角度と距離を計算しています。入手性の高さや扱いやすさからこちらを使用しています。複数個載せることでボールの正確な場所の取得も可能にしました。



サブマイコン: ATmega2560

ボールセンサ・ジャイロセンサ、ラインセンサの処理にATmega2560を採用しました。センサーの値の読み取りや、動作に必要な計算を行っています。その他のマイコンと比べ、ピン数が多く、汎用性が高いため、このマイコンを採用しました。省スペース化のために回路を自作しています。



ラインセンサ: S4282-51

ラインセンサにはS4282-51(光変調フォトIC)を一機当たり25個搭載しています。コート上の白線の読み取りを行っています。外部からの光に影響を受けないため、誤作動が起こりにくく扱いやすいので、こちらを使用しています。また、複数個の搭載により、ロボットがどの位置の白線を踏んでいるのかの取得を可能にしました。



カメラ: OpenMV H7

ゴール・コートの角度・距離を計算しています。プログラムのしやすさや、入手性の高さ、他マイコンとの通信のしやすさからこのカメラを採用しました。



ジャイロセンサ: MPU6050

最初に向いていた方向からどのくらい角度がずれているのか計算し、ロボットの姿勢制御に利用しています。今まで利用していたセンサより小さく、エラーが少ないためこのセンサを採用しました。



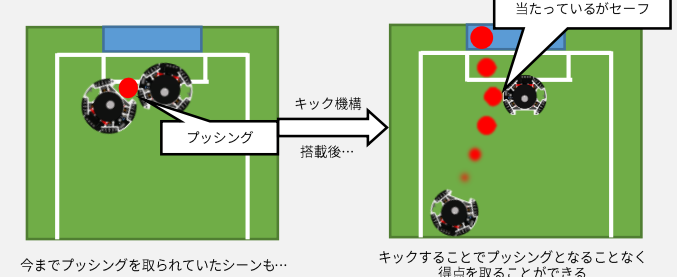
より早く、より正確にーラインセンサの改良

コート上の白線を認識するラインセンサは、円形+十字に配置したものに改良しました。以前よりもセンサーが外側に伸びたため、早くラインに反応することができるようになりました。また、反応した位置から角度と距離を求め、条件を分けることで最適な制御ができるようになり、ルール改定による白線外の空間の縮小にも対応することができました。



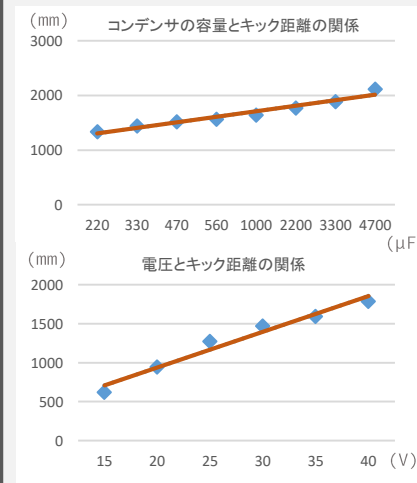
蹴り一つで状況を変えるーソレノイドキッカーの搭載

今季から、ソレノイドキッカーを用いたキック機構を搭載しました。キック機構を載せることで、ロボットが直接ボールを運ぶよりも強い力でシュートすることができます。キック機構を用いることで、自分のロボットとボールを離してシュートできるため、ゴールに入る前に相手ロボットと接触していてもブッシングを取られず、必然的にゴールに入る回数が増えることとなります。また、キック機構と他の機能を一緒に用いることで、より強力なシュートを打つことができます。例えば、カメラを用いてゴールの方向を向きながらボールをキックすることで、前を向いてキックするよりもはるかにゴールへのシュート率を上げることができます。また、自陣を守るロボットであっても、前方にボールをキックすることができるため、アタック・ディフェンス両方の機能を兼ね備えた強いロボットを作ることができます。



＜電圧・コンデンサの容量とキックパワーの関係＞

ソレノイドキッカーは、かける電圧とコンデンサの容量によってキックパワーが変化します。そこで、どの電圧をかけ、どのコンデンサを搭載することでより長距離にボールをキックできるのかという実験を行いました。



左上図より、コンデンサの容量を220μF～4700μFで変化した時、距離がほぼ一次関数的に伸びていくことがわかりました。また、左下図より、電圧を15V～40Vで変化した時も同様に、ほぼ一次関数的に変化していることがわかりました。このような結果から、コンデンサの容量を大きく、電圧を高くすることで威力を高められることがわかりました。そのため、僕たちのロボットでは、コンデンサの容量を、ロボットに載せることのできる最大サイズの4700μFに、電圧を45Vに設定しています。

ツールを駆使するー様々なサービスの利用

ロボット製作をスムーズに行うため、様々なツールを駆使して活動を行っています。例えば、ロボットの設計を早く、正確に行うために、Fusion360(機体設計)や、KiCad(回路設計)などのソフトを活用しています。ロボットの設計以外にも、チーム内で「今自分が何をすべきなのか」、「まだどのタスクが終わってないのか」をはっきりさせるために、「Asana」というツールを利用しタスクの明確化のみならず、日程の管理や情報伝達にも活用し、チーム内での活動を円滑に進めています。また、Githubなどのデータ共有サービスを利用し、チーム内での最新の進捗を常に素早く共有しています。これらのおかげで、わずかな活動時間でも大きな進捗を生み出しています。

駆使するツールはそれだけにとどまりません。例えば、SNSは新しい技術や情報を身に着ける重要なツールの一つです。僕たちは、RCJに参加するうえで、技術の共有をしていくことは必要不可欠であると考えています。そのため、Twitterのアカウント、チームのウェブサイトを作成し、私たちが持っている技術の公開・共有を行っています。プレゼンシート右上にTwitterのユーザー名とブログのQRコードを掲示しているので、ぜひ一度お訪ねください。



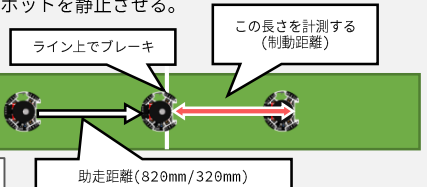
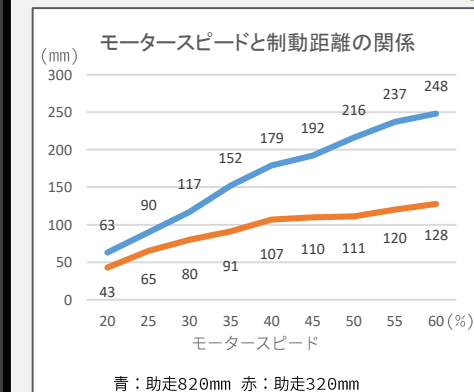
モータのスピードと制動距離の関係

2023ルールから、コートの白線外の場所が狭くなり、ラインアウト判定が「壁に触れたとき」に変更されたため、必然的にラインで正確に止まる必要が出てきました。しかし、僕たちのロボットは「速く、正確に」を目指しており、正確さをとるために速さを犠牲にしたくないので、速さ・正確さを両立させるギリギリを探する必要がありました。そこで、モータのスピードによってどのくらい白線の外に出してしまうのか、という実験を行いました。

＜実験＞

- 1) 白線からそれぞれ820mm、320mm離れた位置にロボットを静止させる。
- 2) 一定のモータースピードでロボットを助走させる。
- 3) 白線を踏んだ時点でモーターにブレーキをかけ、白線からはみ出た距離を計測する。
- 4) モータースピードを変化させて同様に実験する。

＜結果・考察＞



左のグラフのような結果となりました。ライン外空間の幅は120mmであるため、このグラフから見ると、助走が長いときは30%程度、助走が短いときは50%程度でモータを動かせばよいことがわかりました。しかし、この実験では、移動開始時は停止しているため、実際の試合では慣性によって、同じモータースピードでもラインアウトする可能性があります。そのため、この結果を参考にしつつ、どのような処理をすればラインアウトをしないギリギリで停止できるのかを調整していく必要があります。

スポンサー

僕たちがロボット製作をしていくうえで、金銭面や技術面からサポートしていただいております。この場を借りてお礼申し上げます。

JLPCB 様

基板の発注に際する発注費用や送料などの面でサポートをしていただいております。

佐賀大deラボ 様

3Dプリンタ、レーザーカッターなどの機械を利用させてもらうだけでなく、技術交流の場としても活用させていただいております。