

チームNo. 150 チーム名 青大ロボコン研P+青工MKN 所属：青森大学ソフトウェア情報学部/青森工業高等学校

### チーム紹介、目標、意気込み

今年のチームメンバーは高校生4人、大学生4人の計8人、全員初参加です！0からのスタートですが頑張ります！初めてのETロボコンなので、L/R両コースを完走することを目標にチーム一丸となって臨みたいと考えています！

今回のコースでは、RコースのS字カーブが難しい部分だと考えているので、そこをクリアできるように調整を重ねていきたいと思っています。  
難所についても、少しでも手を付けて次に活かせるようにしたいと考えています。

### モデルの概要

- 今回制作したモデルは技術教育で配布されている資料chap-03.pdfと昨年のモデルを基に分析・設計しました。
- コースをライン形状によって直線と緩いカーブと急なカーブに分割して、区間ごとに走行速度とPID制御のパラメータを変更してゴールまで完走を目指します。
- 区間終了判定は（走行距離 $\geq$ 区間距離）です。ロボットの走行距離が区間距離を超えた場合、PIDパラメータと走行速度を変えて次の区間を走行します。
- サンプルプログラムはLコース1分19秒かかったのですが、私たちが設計したプログラムはLコース35秒まで縮まりました。

### モデルの構成

#### 1.機能モデル

- 今回はLR両コースの完走を目標に設計する
- コースをライン形状で区間で分割し、区間毎に走行設定を変更する。（走行速度とPID制御パラメータ）
- 走行距離が区間距離以上になった場合に区間を切り替える
- 機能の実現方法をユースケース記述とアクティビティ図で示す

#### 2.構造モデル

- 機能モデルのアクティビティ図やユースケース記述を基に、機能実現に必要な部品を抜き出す
- 部品の責務や部品同士のつながりをオブジェクト図で示し、クラスの候補を見つける
- 部品間のメッセージのやり取りをコミュニケーション図で示し、部品間の繋がりやモレヌケがないことを確認する。
- オブジェクト図とコミュニケーション図を基に部品の仕様をクラス図で示す。

#### 3.振る舞いモデル

- コースを完走するという機能をシーケンス図を使って示し、時間軸に沿って動作が正しくやり取りされるか確認する。
- シーケンス図が複雑になることを避けるため「相互作用の利用」を使ってシーケンス図を分割しシンプルに表現する。

#### 4.要素技術

コース完走に使用する要素技術を以下に記す

- PID制御：走行体の旋回動作の操作量の制御
- 外乱対策：カラーセンサーのノイズ除去
- 走行距離の計算：区間終了判定に使用
- スタート直後のコースアウトのリスクを減らす
- 転倒判定：転倒時に左右モータを停止

## (1)大会の目標、選択課題

今回の大会では、チームメンバー全員が初参加ということもあり、目標をLR両コースの完走ということにした。

よって、今回はLR両コースの完走をメインに設計する

## (2)走行戦略

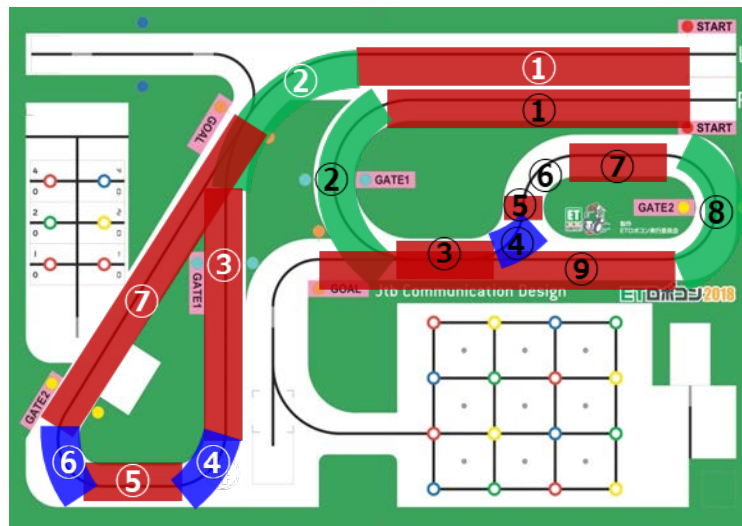
コースをライン形状で区間で分割し、区間毎に走行速度とPIDパラメータを変更する。(各区間の設定は(10)参照)

### 区間分割方法

次の3種類に分割する

- 直線(赤)
- 緩いカーブ(緑)
- 急なカーブ(青)

L/Rコースを区間分割した図を右に示す



LコースRコースの区間距離  
(Lコース 白字、Rコース黒字)

### 区間終了判定

走行距離 > = 区間距離

例) Lコース区間②  
走行距離 > = 1.455(m)

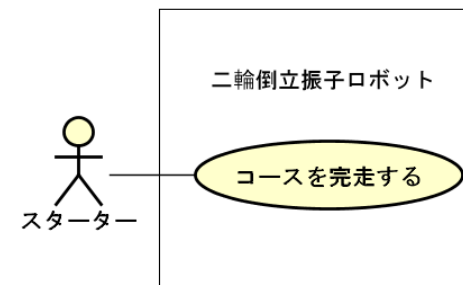
条件を満たした場合、次の区間に切り替える

右表はL/Rコースの区間距離を示す

区間	Lコース(m)	Rコース(m)
①	直 2.505	直 2.100
②	緩 1.455	緩 1.748
③	直 1.766	直 0.498
④	急 0.537	急 0.539
⑤	直 0.576	直 0.057
⑥	急 0.665	急 0.571
⑦	直 2.528	直 0.673
⑧		緩 0.972
⑨		直 2.574

## (3)提供する機能

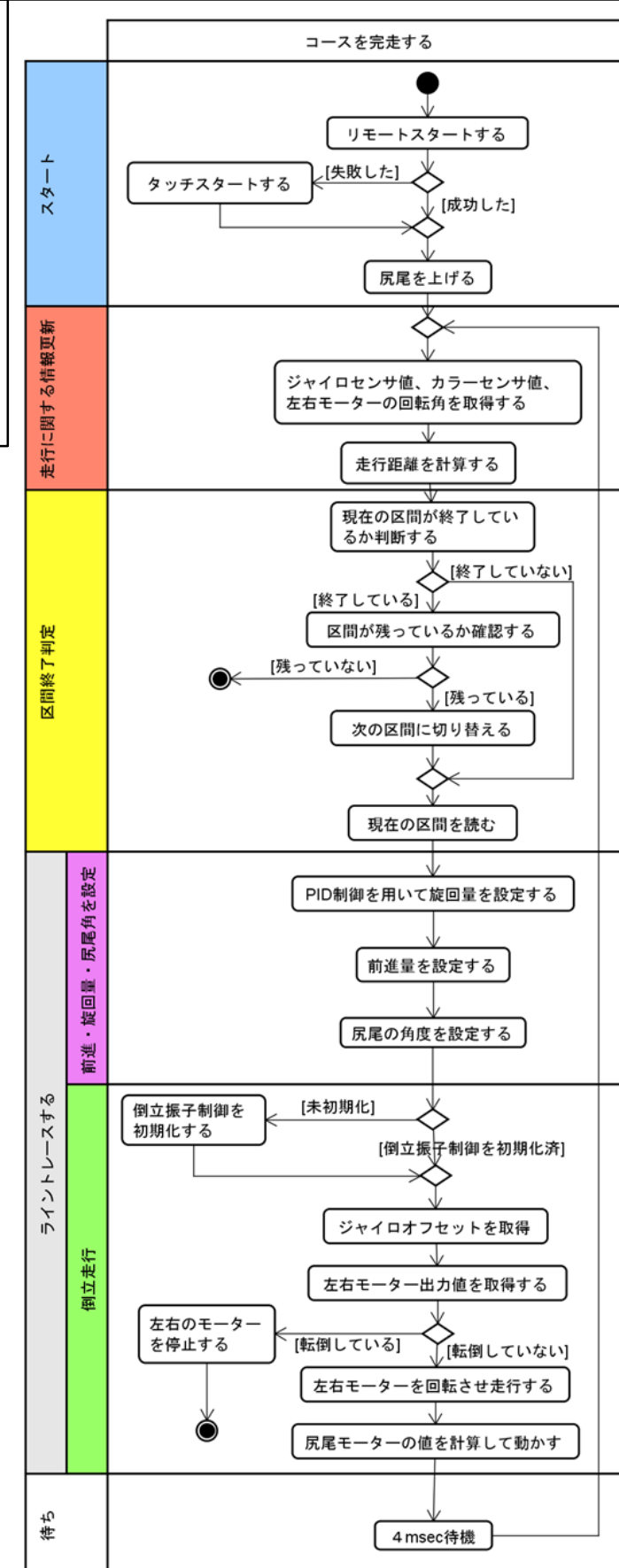
選択課題から、「コースを完走する」ためのユースケース図を記載する



## (4)機能の実現方法

機能「コースを完走する」の実現方法をユースケース記述とアクティビティ図で示す。

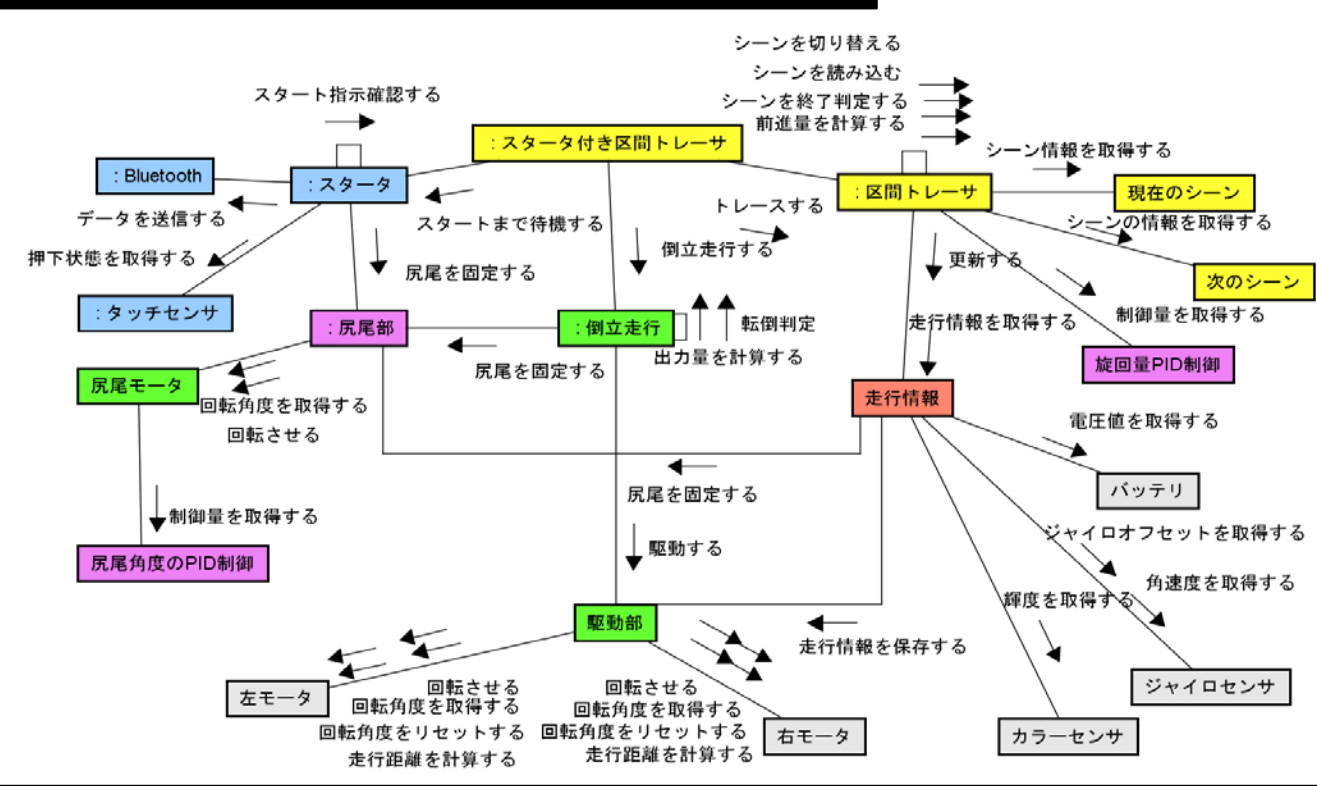
名前	コースを完走する
概要	LまたはRコースをラインレースしゴールまで走行する
事前条件	スタート地点で完全停止する
事後条件	ゴールゲートを通過する
基本フロー	<ol style="list-style-type: none"> <li>1) 走行体はリモートスタートする</li> <li>2) 走行体は尻尾を上げる</li> <li>3) 走行体はモータ、センサの値を取得する</li> <li>4) 走行体は現在の区間の状態を確認する</li> <li>5) 走行体はラインレースを行う</li> <li>6) 走行体は3～5を繰り返す</li> </ol>
代替フロー	<ol style="list-style-type: none"> <li>a. 基本系列 1)で失敗した場合 <ol style="list-style-type: none"> <li>1. タッチセンサでスタートする</li> <li>2. 基本系列 2)に戻る</li> </ol> </li> <li>b. 基本系列 4)で区間が残っていない場合 <ol style="list-style-type: none"> <li>1. 走行を終了する</li> </ol> </li> </ol>
例外フロー	<ol style="list-style-type: none"> <li>A) 走行体が基本系列 5) で転倒した場合 <ol style="list-style-type: none"> <li>1. 走行体は左右モータを停止する</li> </ol> </li> </ol>



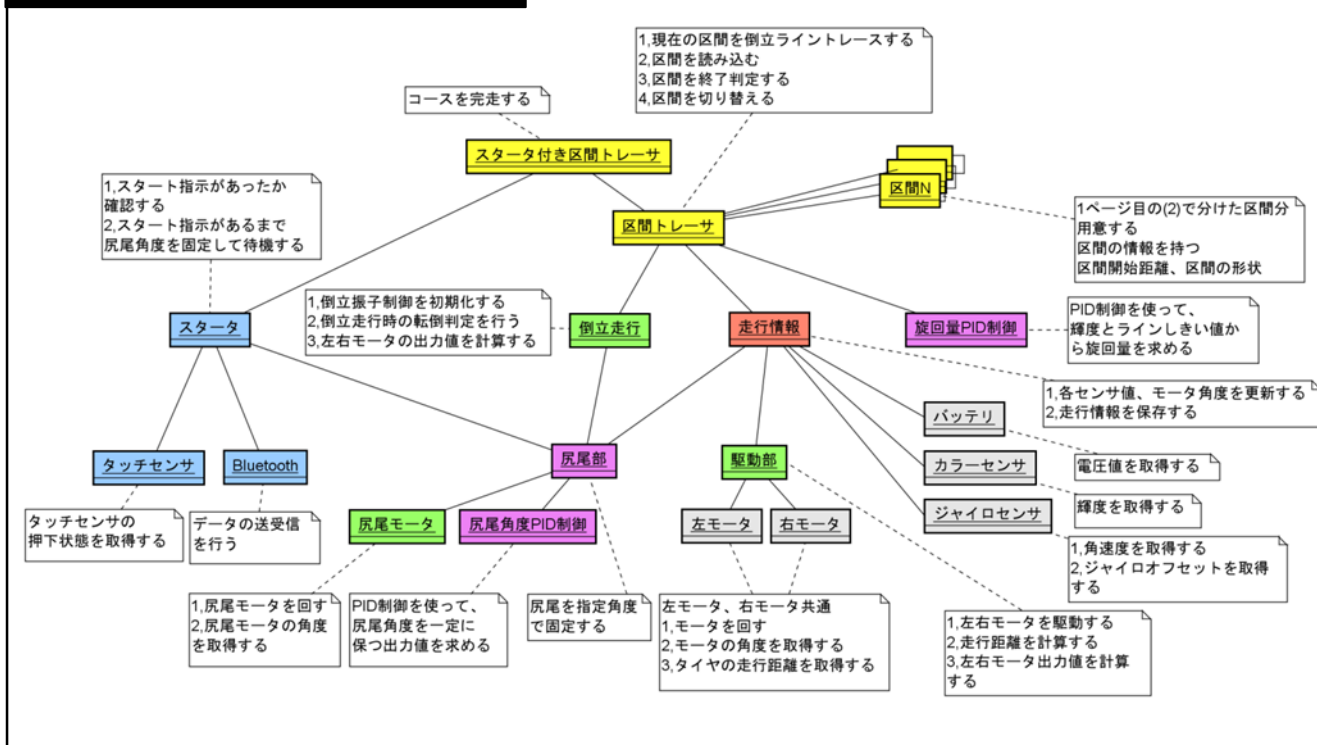
## (5) 機能実現に必要な部品

【役割】や【情報】	【部品】の候補
コースを完走する	スタータ付き区間トレーサ
スタート地点で完全停止する	スタータ, 尻尾部
スタート指示する	スタータ, Bluetooth, タッチセンサ
タッチセンサの押下状態を取得する	タッチセンサ
尻尾を指定角度で固定する	尻尾部, PID制御, 尻尾モータ
走行情報を更新する	走行情報, ジャイロセンサ, カラーセンサ, バッテリ, 駆動部, 左モータ, 右モータ, 尻尾部
走行距離を計算する	駆動部
区間を読み込む	区間トレーサ, 区間
区間を切り替える	区間トレーサ, 区間
区間終了判定する	区間トレーサ, 区間, 走行情報
倒立状態でライントレースする	倒立走行, PID制御, 駆動部, 尻尾部, 走行状態
転倒判定する	駆動部
左右モータを駆動する	駆動部

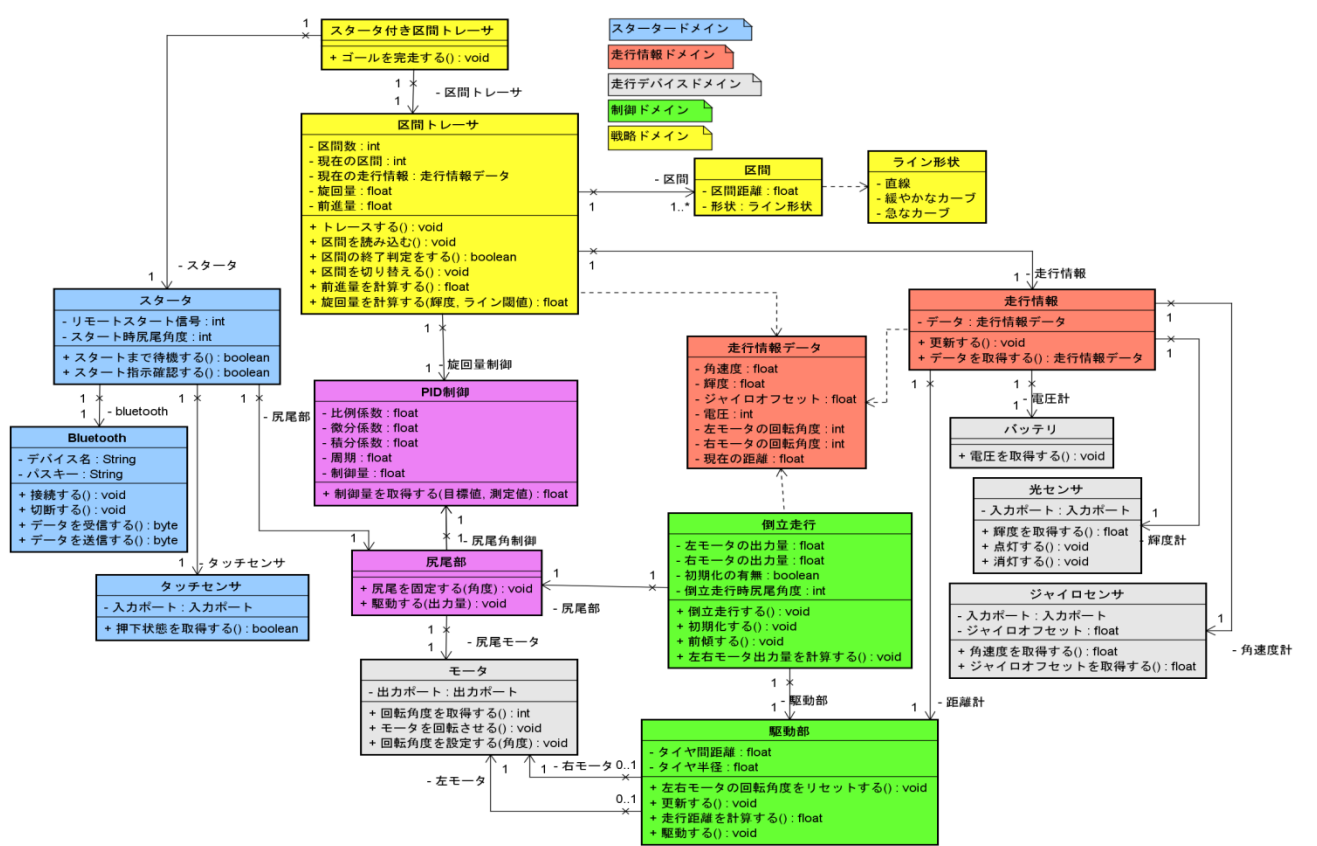
## (7) 部品による機能実現を検討する



## (6) 部品を定義する

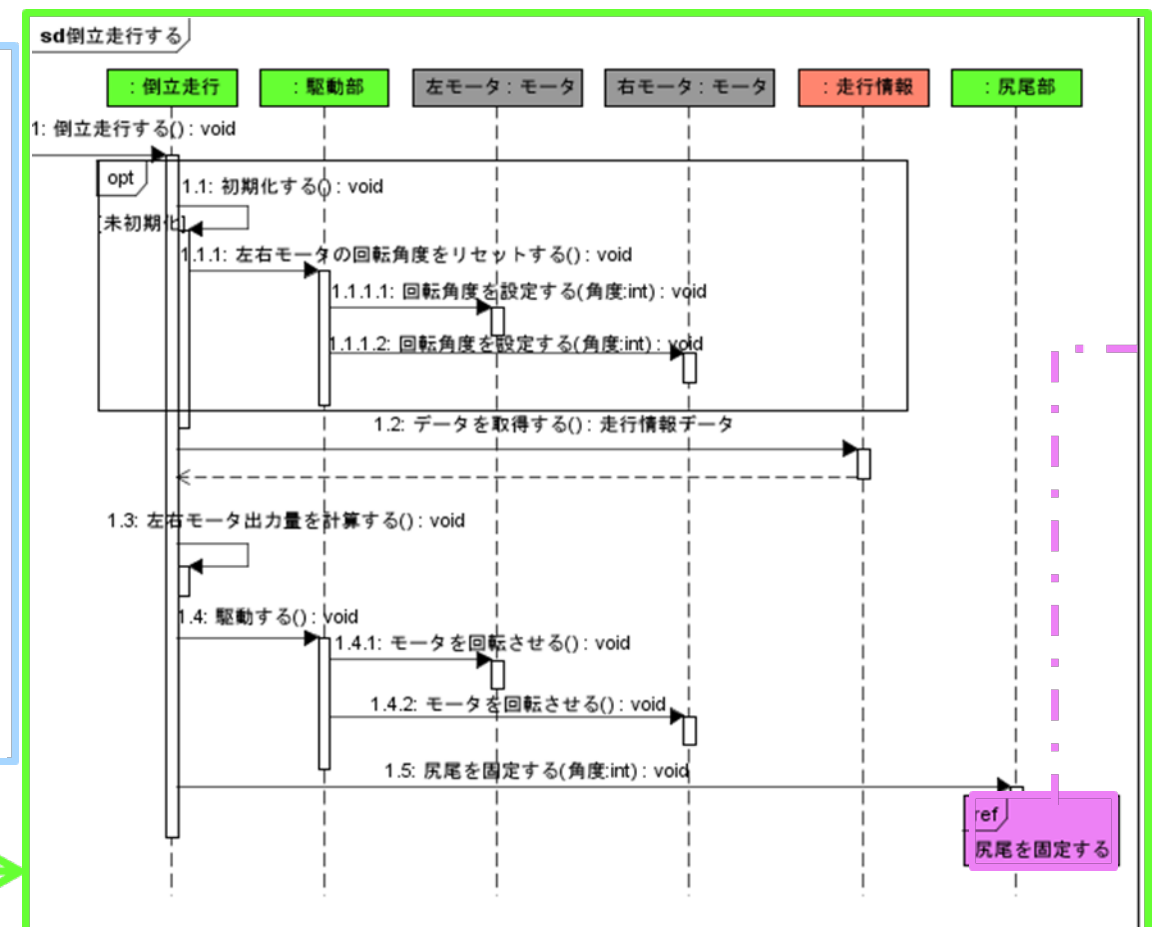
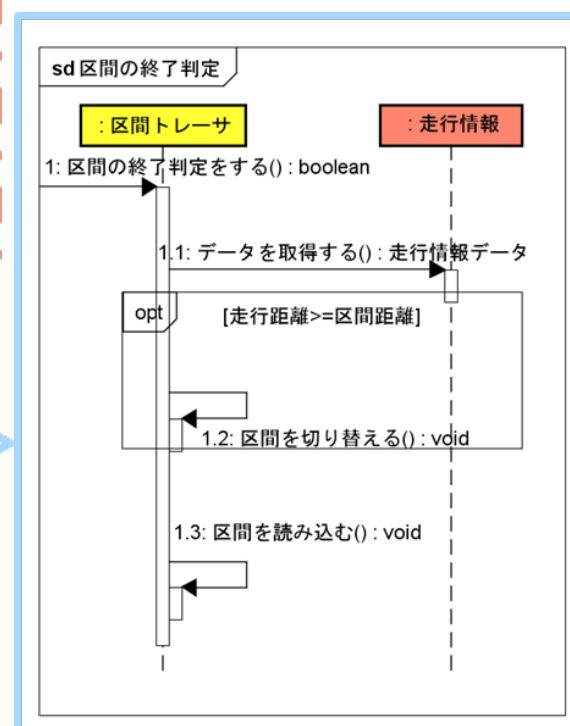
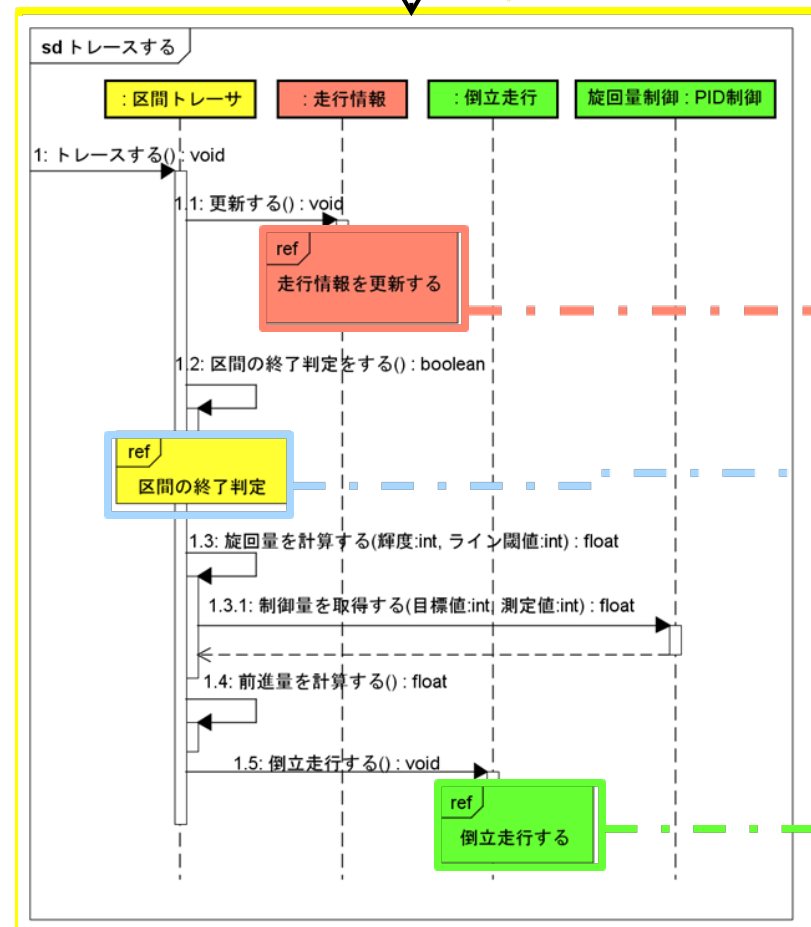
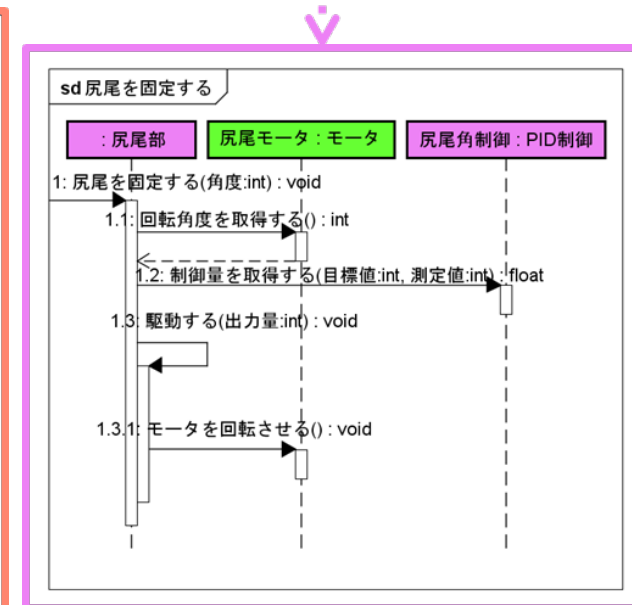
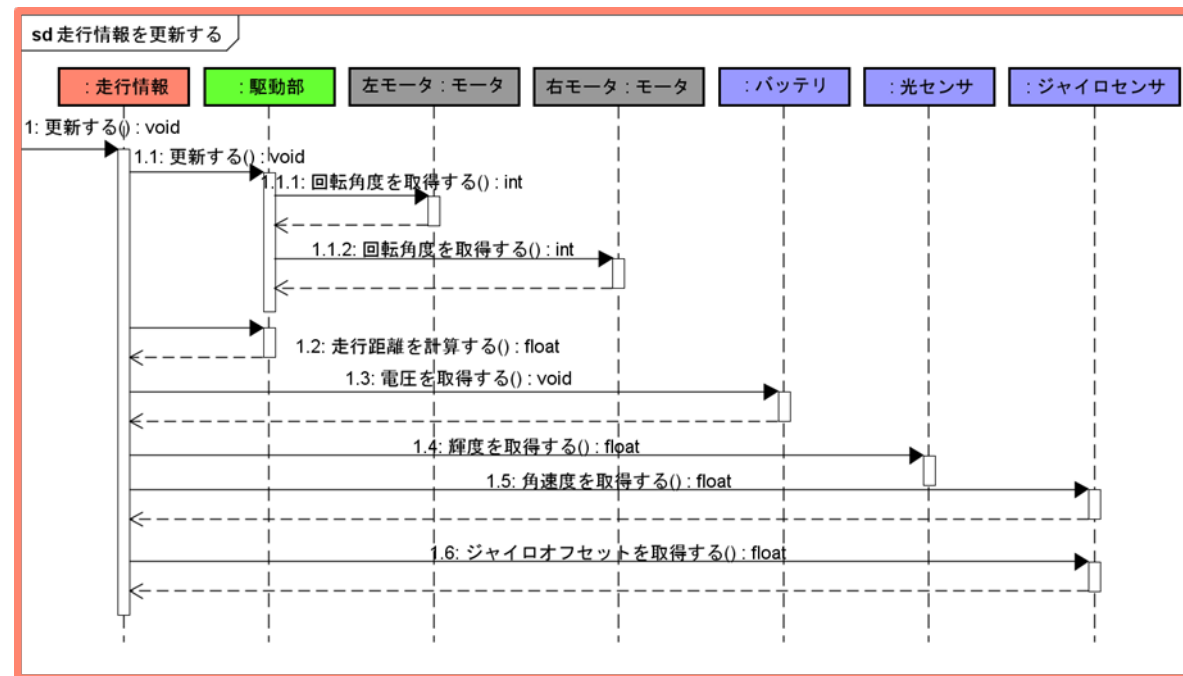
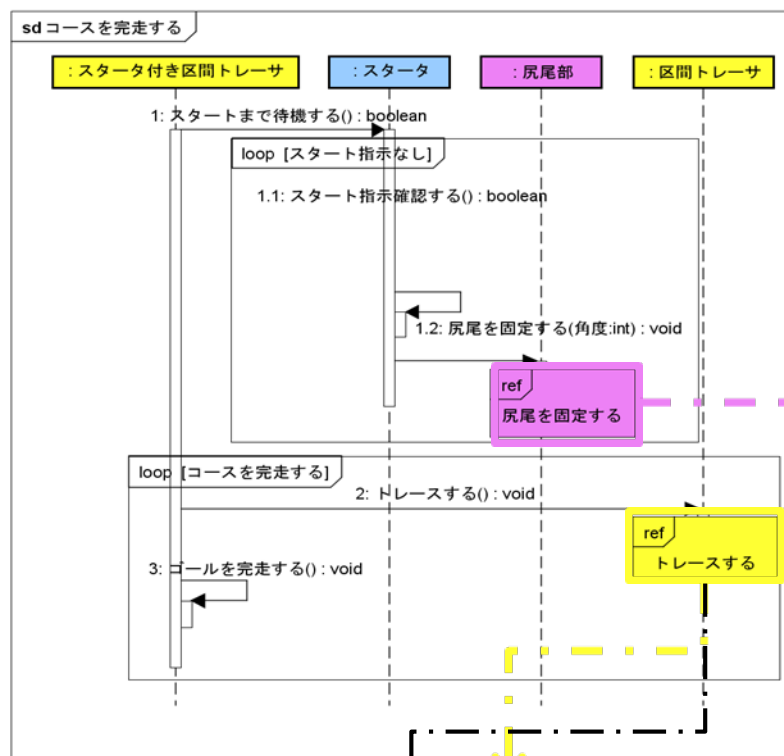


## (8) 部品の仕様を定義する





## (9)部品による機能実現を確認する



## (10)PID制御

走行体の旋回動作の操作量の制御を行う際にPID制御を行う。

$$\text{操作量} = K_P e_n + K_I \Delta t \sum_n e_n + K_D \frac{e_n - e_{n-1}}{\Delta t}$$

$e_n$ : 偏差 = (目標値-測定値)  
 $K_P$ : 比例係数  
 $K_I$ : 積分係数  
 $K_D$ : 微分係数  
 $\Delta t$ : 時間ステップ(0.004[s])

区間	速度	Kp	Ki	Kd
直線	95.0	50.0	16.0	7.3
緩いカーブ	80.0	50.0	16.0	7.3
急なカーブ	70.0	62.5	16.0	9.3

※右表は現時点の調整したソ  
走行速度、PIDパラメータを  
記している。

分割した区間に合わせて走行速度・PIDパラメータを  
変え、コースアウトしないようにする。  
 走行速度、PIDパラメータは目安として三段階に分け  
 る。上の表は区間ごとのPIDの値を示している。

※尻尾のPID制御は省略する。

## (11)外乱対策

ローパスフィルタを用いることでノイズの  
影響を抑え、急な誤作動が起こらないよう  
にする。計算式を以下に示す。

$$y_i = \alpha y_{i-1} + (1 - \alpha) x_i$$

$x_i$ : 今回のカラーセンサの測定値

$y_i$ : 出力値

$y_{i-1}$ : 前回の出力値

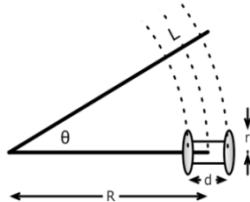
$\alpha$ : 係数(0.9)

## (12)走行距離

走行距離 ≥ 区間距離を満たす場合に、区  
間毎に走行設定を変更する。計算式を以  
下に示す。

$$L_n = \frac{L_n^l + L_n^r}{2}, L_n^l = \frac{2\pi r}{360} \phi_n^l$$

$$L_n^r = \frac{2\pi r}{360} \phi_n^r$$



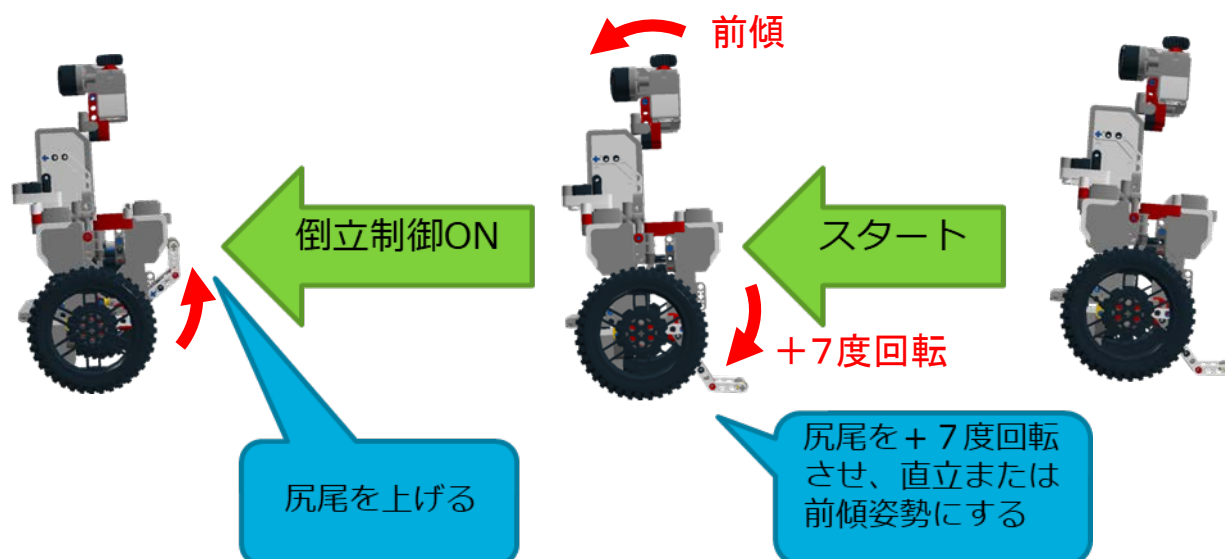
$L_n$ : 走行体の走行距離[m]  $\phi_n^r$ : 右モータ回転角度[度]

$L_n^l$ : 左モータ走行距離[m]  $\phi_n^l$ : 左モータ回転角度[度]

$L_n^r$ : 右モータ走行距離[m]  $r$ : タイヤ半径 (0.049[m])

## (13)スタート直後のコースアウトのリスクを減らす

走行体がスタート位置からバックしないように、スタート直後に尻尾モータを+7度  
回転させて直立に近い前傾姿勢にする。これまではスタート直後に走行体がバックし  
て倒れたりしたが、変更を加えたことでバックせず倒立走行に移行するようになった。



## (14)転倒判定

走行中に点灯すると左右モータ出力が100また  
は-100になることが左グラフで確認できる  
(2013年の資料を使用)

左右モータ出力が100または-100と等しい状  
況が500回連続(2秒間)発生した場合、走行体  
が点灯したと判定し、左右モータを停止する。

