

### ◆チーム紹介

個性豊かな新入社員6名で構成されたFreshers3です！  
真面目さと緩さをバランスよく組み合わせ、マイペースで作業に取り組んでいました。

### ◆チーム目標

チーム目標は「**コース完走**」です。  
難所も含めて全コースを走り切れることが最重要だと考え、この目標にしています。

### ◆意気込み

ここでの実装・設計・チームマネジメント経験などが業務に活かせるように、積極的に取り組んでいきます！

### モデルの概要

#### ➤ 選択課題

「**コースを完走する**」です。

#### ➤ モデル全体概要

- 詳しくは「モデル構成」に記載しますが、全体として**選択課題が実現できること、実現するためのフロー、システム開発・改修の容易さ**を意識し、設計を行いました。
- 開発にあたり注力した要素技術にはアイコンを配置し、可読性を高めています。アイコンにご注目いただくと、理解が容易になると考えます！

#### ➤ 効果・実績

- 安定した完走を実現した。
- 安定完走のための安定スタート、安定走行を実現した。

### モデルの構成

#### ➤ 機能モデル(p2)

- コースを完走するという目標を達成するために**必要な要求、また要求を満たすために必要な要件**を要求図で洗い出しました。

洗い出した**要件がシステム内でどのような流れで実行されているのか**を、ユースケース図、ユースケース記述を用いて示しました。

#### ➤ 構造モデル(p3)

- 1つの要求を実現するために複数パッケージを編集する事態を回避**できるパッケージの構成を考えました。
- プログラムの改修の際、**編集範囲が最小になる**ようなクラス構成を考えました。

#### ➤ 振る舞いモデル(p4、5)

- ユースケース図で抽出した各機能、それに対応するパッケージをもとに、アクティビティ図、シーケンス図の設計を行いました。

#### ➤ 工夫点(p6)

- 要素技術の詳しい説明を示しています。

### モデルのここに注目！

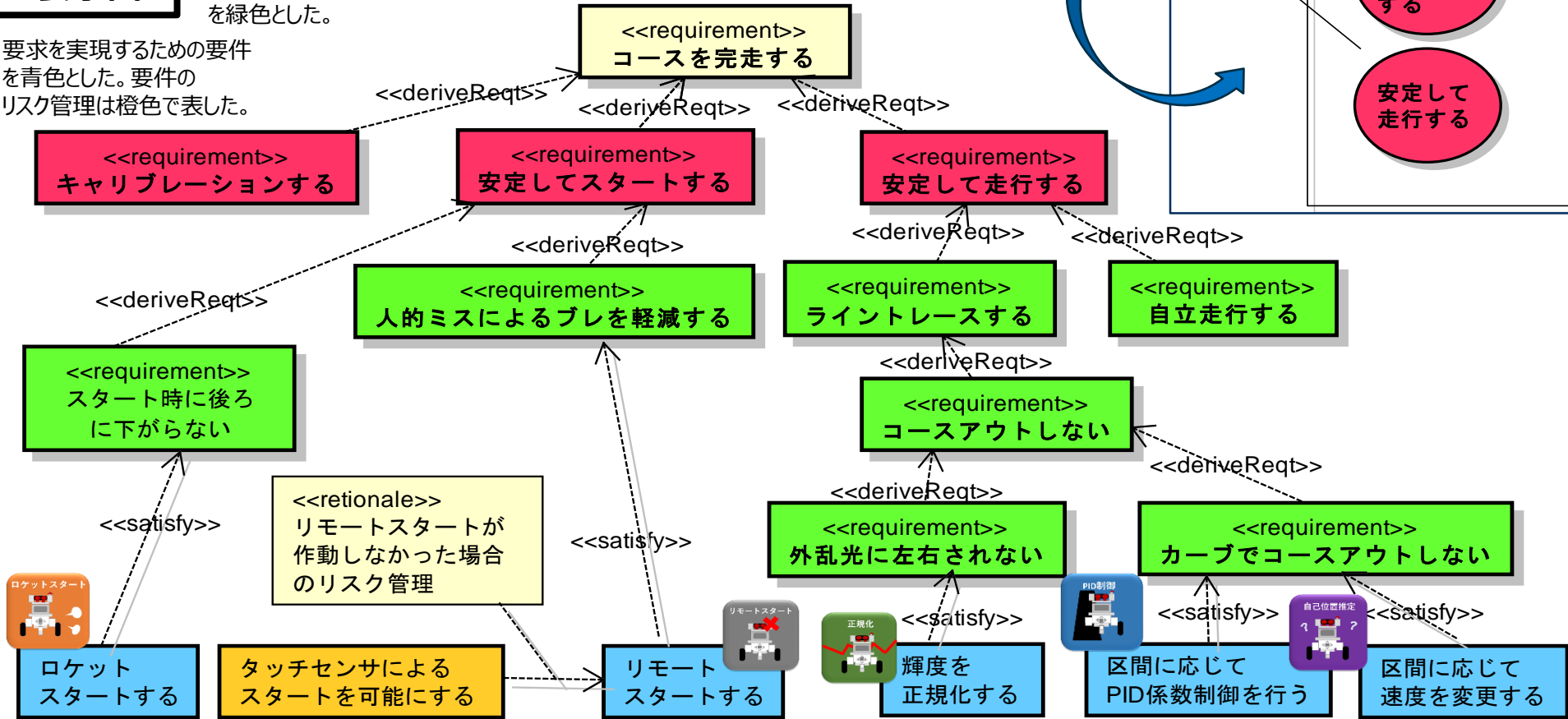
- 目標が確実に達成できる**機能モデル。
- 1つの要求を実現するために**複数パッケージを編集する必要が無い**、別々の要求を実現するための**各開発者が、同一パッケージを編集する必要が無い**構造モデル。
- 各要件がどのような流れで実行されるのか理解できること、本当にそれらの要件が実現できる**振る舞いモデル。

要求図を以下に示す。  
最上位要求として、選択課題「**コースを完走する**」を設定した。  
要求の抜け漏れを防ぐため、時系列で洗い出すことにした。  
そのため、最上位要求を満たすための機能要求を「**安定してスタートする**」「**安定して走行する**」とした。  
二つの機能要求を満たすために、非機能要求を細分化して洗い出した。  
非機能要求を達成するために、各非機能から要件を考えた。

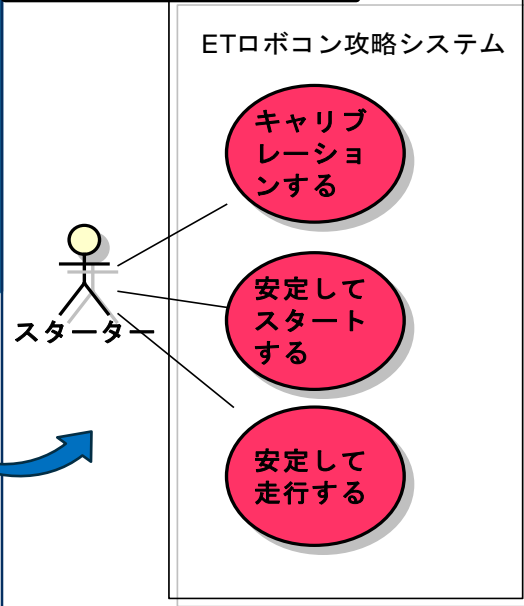
1.要求図

機能要求を赤色、非機能要求を青色とした。

要求を実現するための要件を青色とした。要件のリスク管理は橙色で表した。



2.ユースケース図



3.ユースケース記述

項目	内容	①
ユースケース	キャリブレーションする	
概要	黒色閾値、白色閾値、目標輝度値を取得・設定する	
アクター	スターター	
事前条件	各センサ値初期化終了	
事後条件	キャリブレーション終了	
基本系列	1. 光センサ黒色値を取得する 2. 光センサ白色値を取得する 3. 目標輝度値を設定する	

項目	内容	②
ユースケース	安定してスタートする	
概要	bluetoothに接続し、走行開始コマンド受信を判定する	
アクター	スターター	
事前条件	キャリブレーション終了	
事後条件	走行開始コマンド受信	
基本系列	1. bluetoothを接続する 2. 走行開始コマンドを受信する	
代替系列	基本系列1.において bluetooth未接続だった場合 1. タッチセンサ押下を判断する	

項目	内容	③
ユースケース	安定して走行する	
概要	走行体が走行開始コマンド受信後、コースを完走する	
アクター	スターター	
事前条件	走行開始コマンド受信	
事後条件	ゴールゲート通過	
基本系列	1. 輝度の正規化を行う 2. 自己位置推定から現在区画を取得する 3. 区間PID係数を取得する 4. PID係数を元に旋回量を決定する 5. 区間前進速度を取得する 6. 倒立走行を行う	

4.要素技術の概要

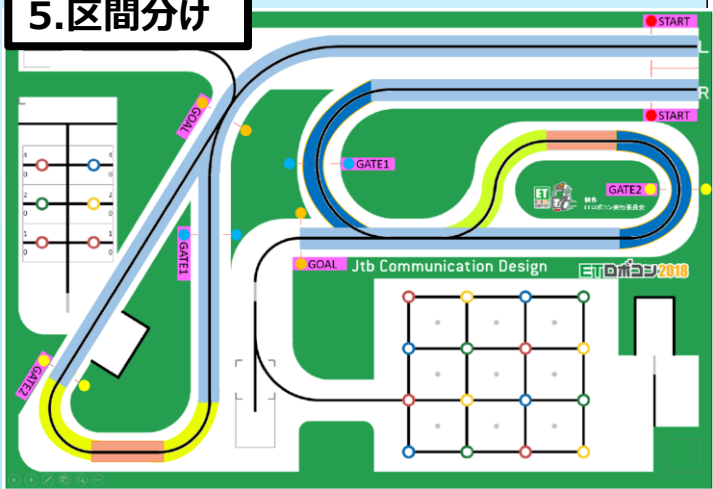
要求図で洗い出した要件のうち、要素技術として用いるものの概要を「4.要素技術の概要」に示した。  
モデル全体の一貫性を保つために、アイコンを作成し使用する。

「カーブでコースアウトしない」要求を満たすために、コースを直線とカーブで区間分けした。色別で「5.区間分け」に示す。

どちらも、詳細はp6.工夫点で記述した。

要素技術	概要
リモートスタート	スタート時の人的ミスを減らすため、Bluetoothによるリモートスタートを行う
ロケットスタート	スタート直後のバランスを保つために、前傾姿勢でスタートする
輝度正規化	外乱の影響を受けずに安定した走行を行う
自己位置推定	左右車輪回転角度値から、現在位置を把握する
PID制御	滑らかなラインレースを行う

5.区間分け



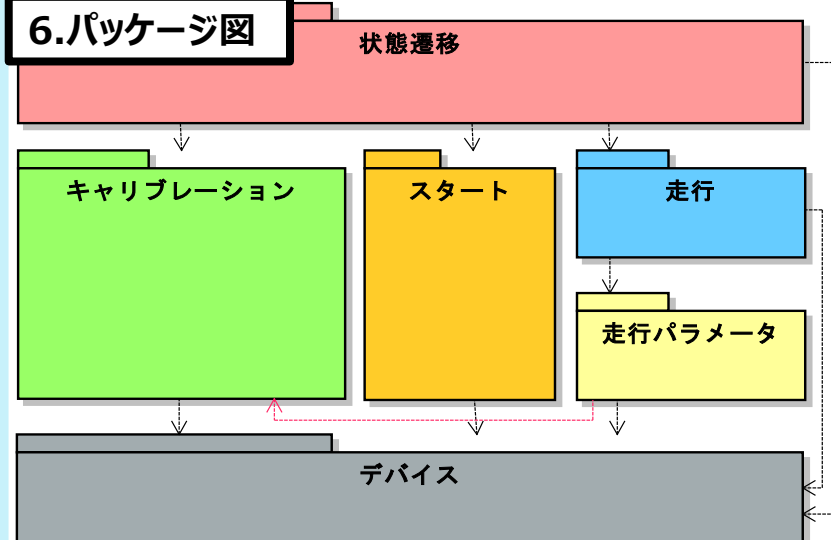
## 機能モデル

# 構造モデル

## 振舞いモデル

工夫点

## 6.パッケージ図



保守性の向上を目的として、機能ごとにパッケージを分類した。

機能ごとにパッケージを作成し、状態遷移パッケージで管理する。センサやモータ等のデバイスドライバは、デバイスパッケージで管理する。(※)

パッケージ図の概要を右に示す。

※可読性向上のため、交差している線は赤色で示した。

パッケージ名	概要
状態遷移	キャリブレーション、スタート、走行の状態を管理する
キャリブレーション	キャリブレーション実施、データ取得を管理する
スタート	Bluetoothによるリモートスタート等を管理する
走行	走行に必要な制御を管理する
走行パラメータ	区間速度や区間PID係数等を管理する
デバイス	各センサのAPI呼び出し等を管理する

## 7. クラス図

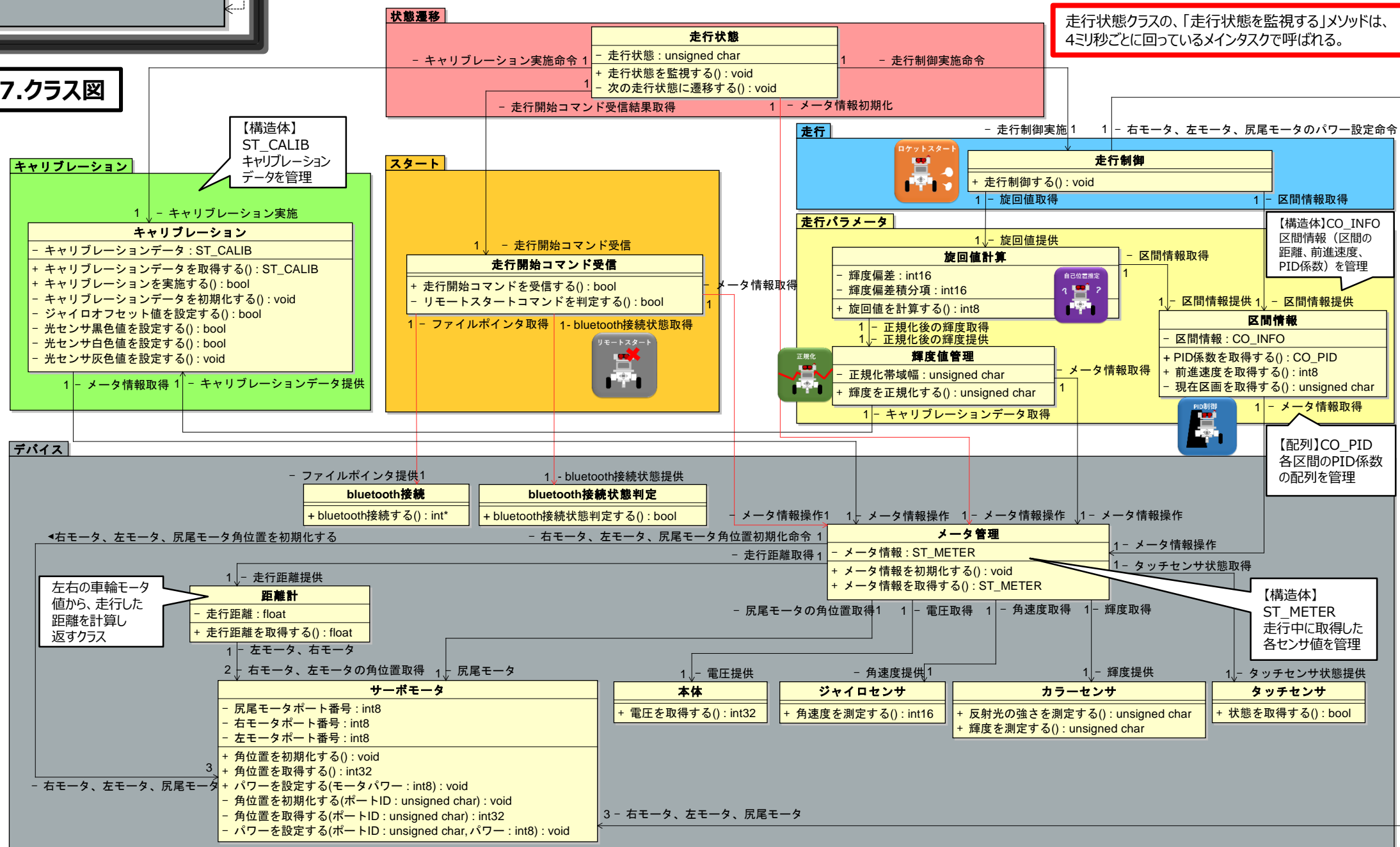
選択課題「コースを完走する」を達成するために必要な機能・要素技術を部品化し、クラス図で記述した。

保守性を向上させるために、部品を独立化することを目的とした。  
結果、完走するためのパラメータ調整がスムーズになった。  
新たな要素技術の追加にも対応できるようになった。

また、ST\_CALIBやST\_METER等の構造体を用いることで、実装者が各センサなどの働きを意識せずとも実装できるようになった。

※可読性向上のため、  
交差している線は赤色で  
示した。

要素技術と対応する箇所には  
アイコンを配置した。





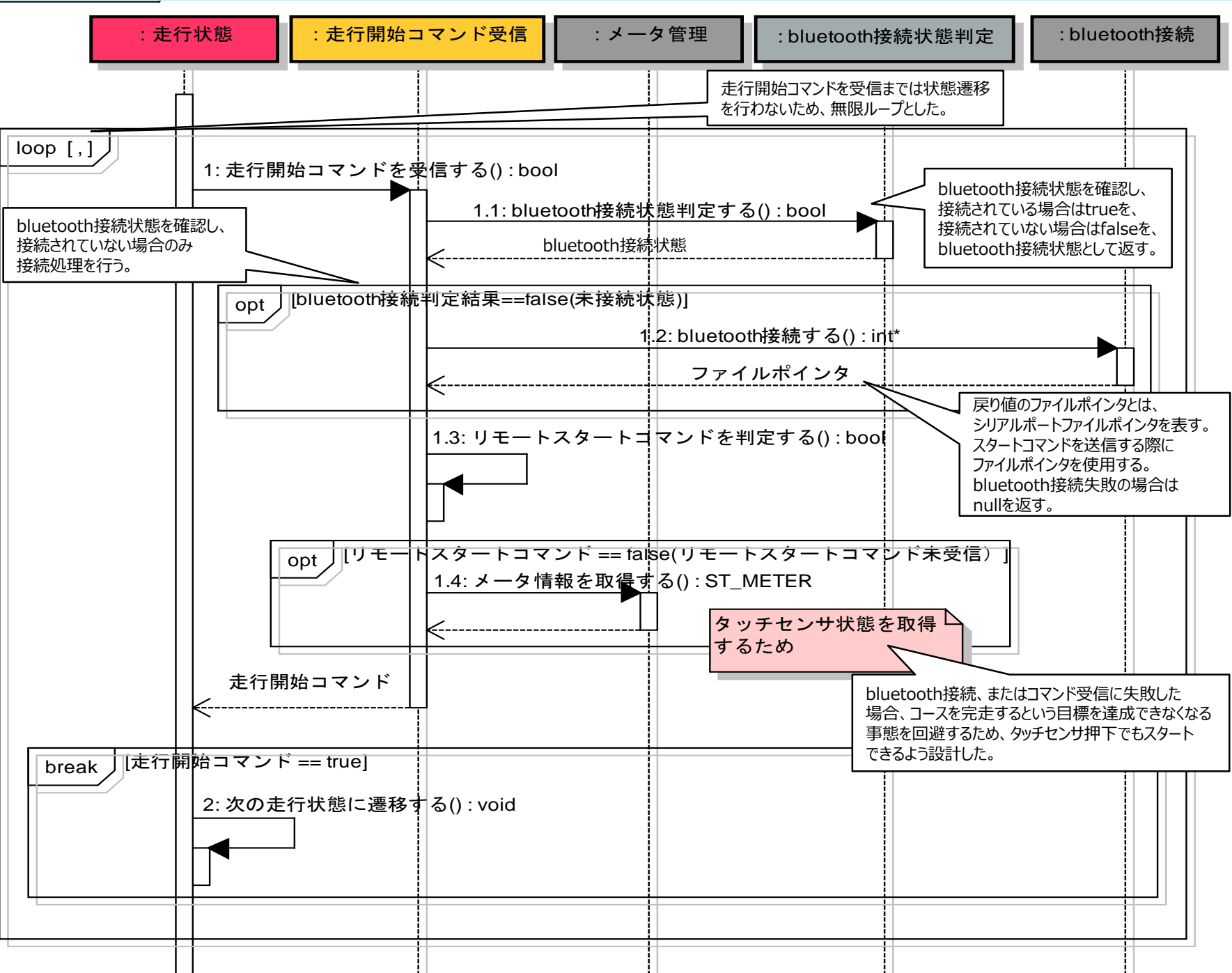
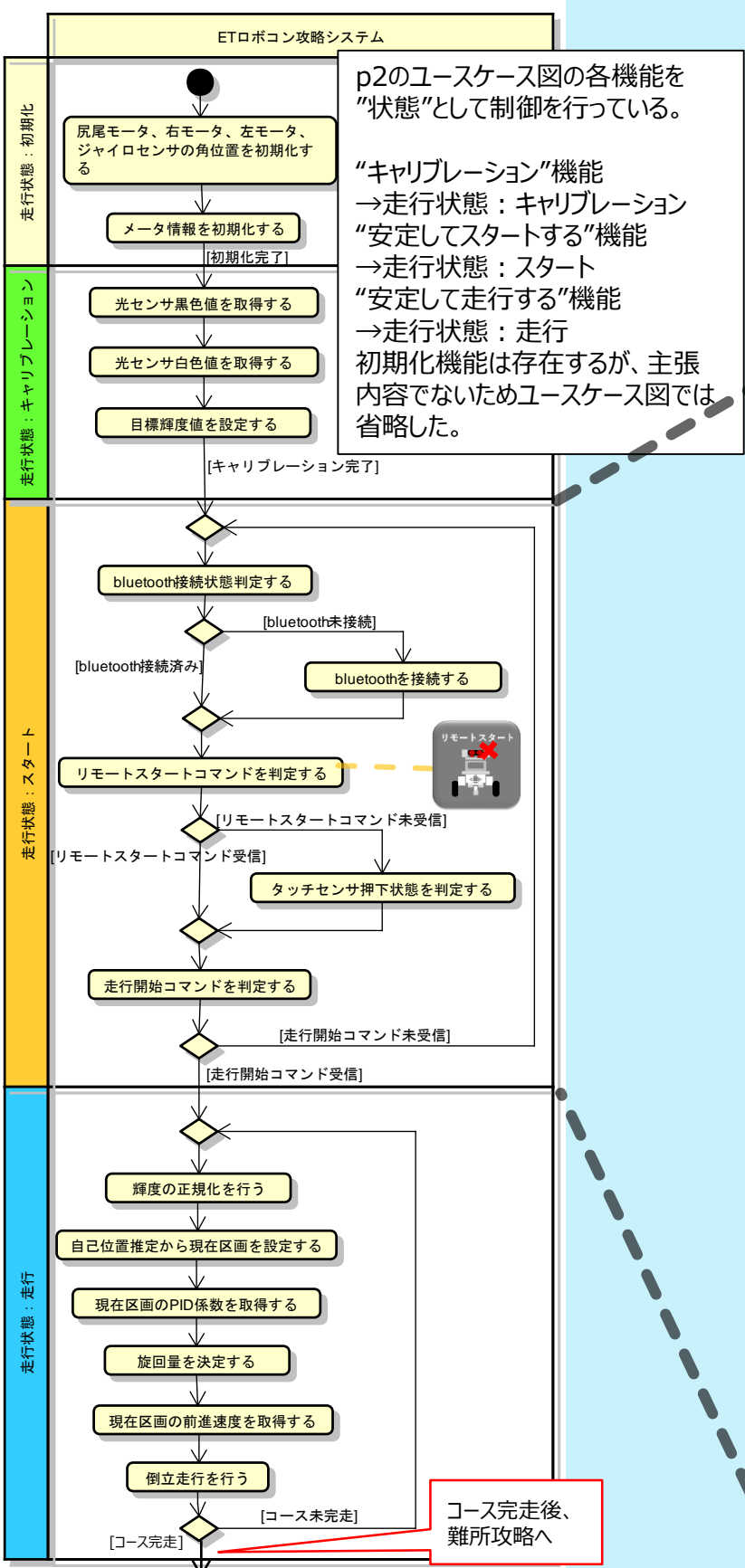
8.アクティビティ図

システム全体の流れを示す。

9.シーケンス図(走行状態：スタート)

選択課題「コースを完走する」を達成するために抽出したクラスについて、各クラスの振舞いをアクティビティ図とシーケンス図で示す。  
振舞モデル(1)では、スタート時の振舞いを示した。

bluetooth接続を行い、接続成功の場合、リモートスタートコマンド受信で「走行状態：走行」へ遷移する。  
接続失敗の場合、タッチセンサ押下で「走行状態：走行」へ遷移する。ユースケース図の“安定してスタートする”機能に対応した。

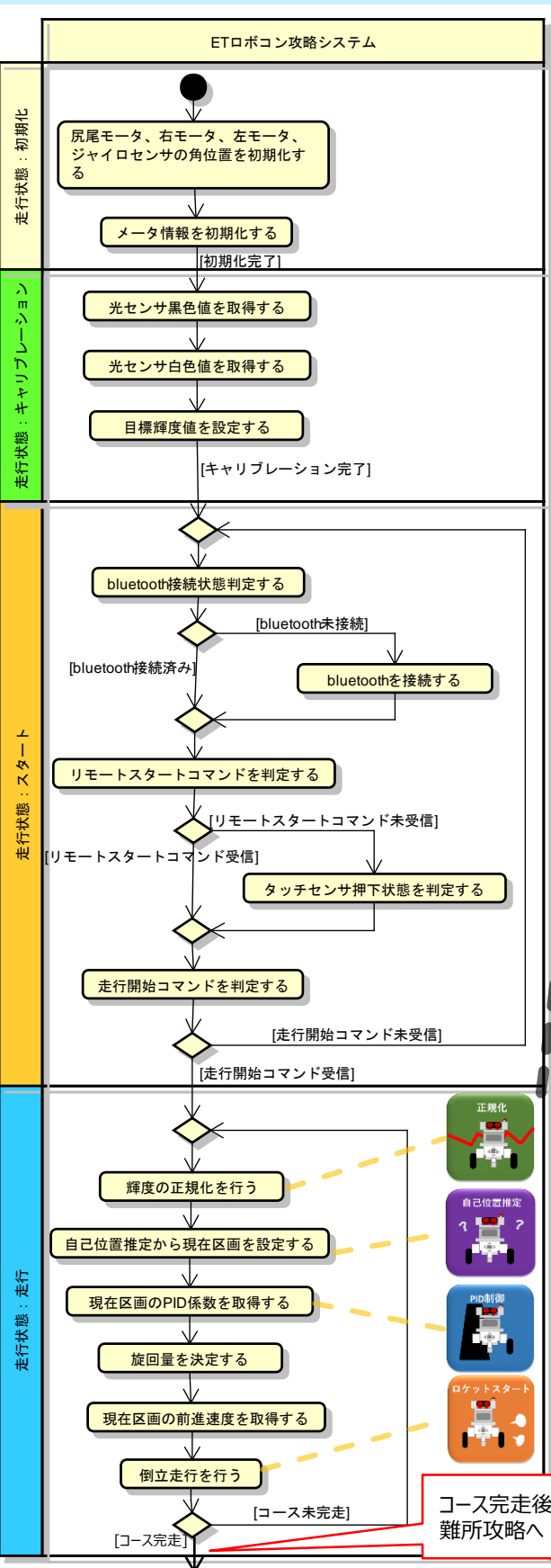
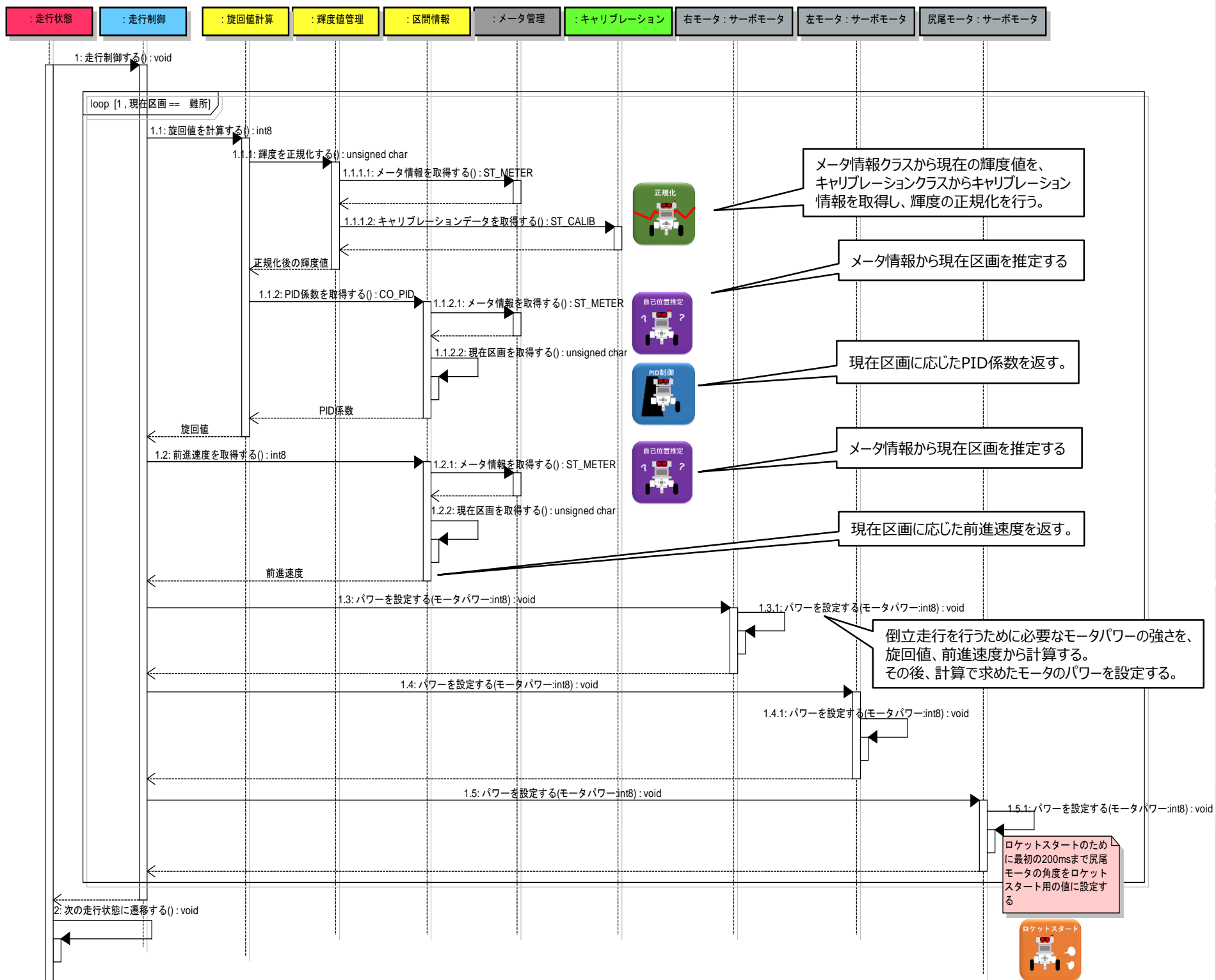


## 10. アクティビティ図

## 11. シーケンス図(走行状態：走行)

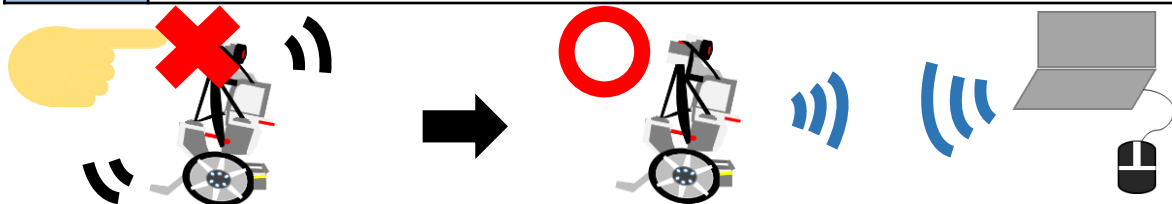
振舞モデル(2)では、走行時の振舞いを示した。

自己位置推定から現在区画を推定、旋回値と前進速度を求める。旋回値計算の際は輝度正規化により、外乱による影響を低減する。“安定して走行する”機能に対応している。難所攻略に対応できるよう設計した。



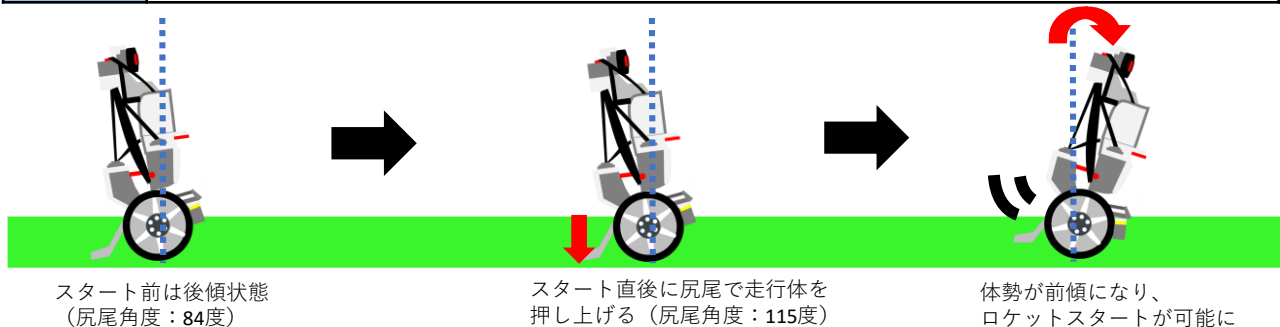
## ① リモートスタート

課題	タッチセンサによるスタートは人間の手で行うため、力加減によりスタートの際にバランスを崩す。
対策	bluetoothを用いてスタートする。また、bluetoothによるスタートが失敗した場合は、タッチセンサでもスタートできるように二重の対策を行う。
結果	人為的ミスが減り、スタート時にバランスを崩さなくなった。



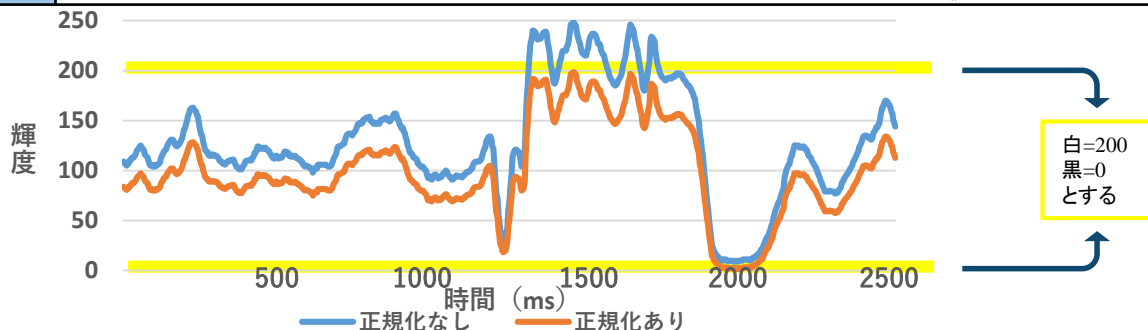
## ② ロケットスタート

課題	尻尾を使った倒立状態は後傾姿勢なので、スタート直後にバランスを保とうとし、後退してしまう。
対策	スタート直後に尻尾で前傾姿勢にすることで状態を安定させる。
結果	スタート直後のバランスを保つ動作が無くなり、安定しているロケットスタートが可能になった。



## ③ 輝度の正規化

課題	外乱によりカラーセンサで取得した輝度の変動が激しく、PID制御に影響が出てしまう。
対策	取得した輝度を一定の幅（Max：200、Min：0）に収まるように補正する。
結果	外乱の影響を受けず、安定した走行が可能になった。
数式	$f(x) = (\text{Max} - \text{Min}) * (\text{Light} - \text{Black}) / (\text{White} - \text{Black})$ Light：現在の輝度 Black：黒のキャリブの値 White：白のキャリブの値

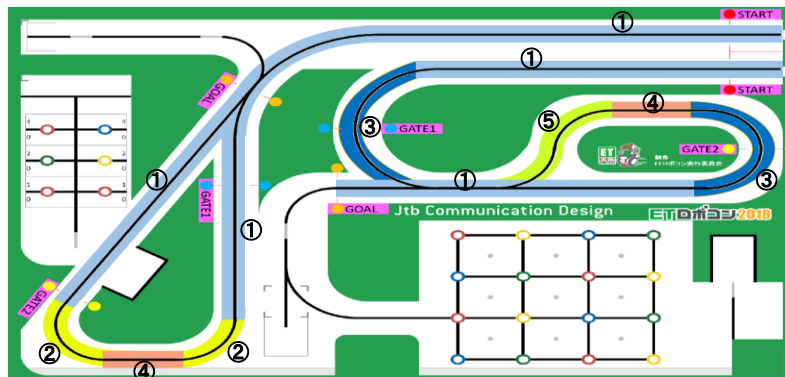


## 達成課題：コース完走



## ④ 自己位置推定

課題	直線とカーブでPID値・速度を変更するためには、現在位置(走行距離)が必要となる。
対策	左右のモータパワーから、走行距離を算出する。
結果	直線とカーブでPID値・速度を変更することが出来るようになり、カーブでもコースを外れないトレースが可能になった。
数式	$D = (\pi R_H (E_r + E_l)) / 2 * 360$ D：走行距離[cm] R <sub>H</sub> ：走行体の車輪の直径[cm] E <sub>r</sub> ：右モータ回転角度[deg] E <sub>l</sub> ：左モータ回転角度[deg]



コースシナリオ

コース番号	コース形状	速度
①	直線	高速
②	中カーブ	中速
③	大カーブ	低速
④	直線	中速
⑤	S字カーブ	低速

カーブの間で急激な速度変化に対応できるように、「中速」にしている

## ⑤ PID制御

課題	ON/OFF制御では無駄な動きが多く、滑らかなトレースが実現できない。
対策	PID係数制御を用いて、目標値付近を維持する適切な旋回量を算出する。
結果	安定した滑らかなライントレースが実現した。
数式	$K_p e + K_i \int e dt + K_d \frac{de}{dt}$ K <sub>p</sub> ：比例ゲイン K <sub>i</sub> ：積分ゲイン K <sub>d</sub> ：微分ゲイン e：光センサの目標値

