ETロボコン2018

デベロッパー部門 プライマリークラス

_{チームNo.} 180 _{チーム名:} さんかわーズ _{所属:} 個人



チーム紹介、目標、意気込み

チーム名は、もちろん「三河ers」。 愛知県の三河地区に根差すことへの誇りを持って参加する。

目標は、開発者・設計者の思いをしっかり『伝える』こと。
やんちゃな動きと、ドキュメントで。

ハマへの憧れは終わらない!

モデルの概要

- 選択機能: 『コースを完走する』
- ・ 完走には大きく5つの工程で実行する機能が必要であると考えた。
- 機能モデルでは、これらのユースケースに加え、完走を妨げるネガティブアクターから、ミスユースケースと緩和ユースケースを抽出した。これらから、コース完走の全体の動きをアクティビティ図にまとめた。
- 構造モデルは、4つに大別されるまとまりのクラス図にまとめた。全体工程から、走破のシナリオを持つクラスを起動する形で、シナリオ部も開発分担も考慮して3つに分けている。
- 振舞いモデルはシーケンス図で示した。倒立走行でのライト レースの計時コース(*) 走破については詳しく示している。
- * スタート直後からゴールゲート通過までを「計時コース」、その後を「難所コース」「停車」と区分している。

モデルの構成

地区: 東海

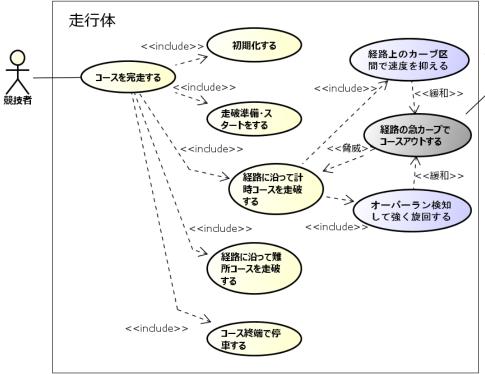
- 1. 機能モデル
- 計時コースを走破するミスケースとして、設計から許容できない競技コース環境の差異から発生する「急カーブでのコースアウト」を考え、カーブでの走行体の動きを抑える機能とオーバーランの程度が小さいうちに対処する機能を緩和ユースケースとして加えた。急カーブを判別するためのコースの区間分割も示した。

地域:愛知県岡崎市

- アクティビティ図は、走行体デバイス等の初期化から停車までの動きをまとめた。
- 2. 構造モデル
- 全体を「走行制御」「走破シナリオ」「走行手法」「入出力 デバイス」で括って、構造をクラス図で表した。コースの走 破戦略を司る「走行シナリオ」は、「計時コース」「難所 コース」「停車」のクラスに分けてデータも保持し、個別に 開発・設計できるよう考慮した。
- 3. 振舞いモデル
- シーケンス図では、まず最上位の「走行制御」全体の振舞い を示した。その中で、準備・スタートの部分と計時コース走 破の部分について、別に詳しく記載した。コースを区間分割 していて繰り返しとなるため、代表的な部分のみを記載した。
- 緩和ユースケースに当たる機能もこの計時コース走破シーケンスに織り込んでいる。
- 4. 工夫点
- 「初歩的な開発環境の中で安全で効率的な走行テストをする」という課題に対して、走行シナリオ部分で開発分担を考慮した分割にし、そのシーケンスは主たるメソッドを2つ設けて、個々の開発者が常に全体の工程を動かして開発にあたることで、安全性と効率の向上を狙っている。
- ※「難所コースの走破」に関わる部分は、各図で省いても誤解の 生じないものを多く省略している

機能モデル

ユースケース図およびユースケース記述



競技コース環境への走行体の不適合

ユースケース記述

[初期化する]

- ・競技者が走行体を起動すると、走行体は動作を開始する
- ・走行体は動作を開始すると、デバイスやライブラリを初期化する

[走行準備・スタートをする]

- ・競技者は走行準備を指示する
- ・走行体は走行準備の指示を受けると、キャリブレーションの結果を保持する
- ・競技者は走行開始を指示する
- ・走行体は走行開始の指示を受けると、静止状態から倒立して走行を開始する

[経路に沿って計時コースを走破する]

- ・走行体はコース内の経路(黒線)付近の状況を調べ、倒立を継続してライントレース走行をする
- ・自車位置を計算して、走破状況を把握する
- ・これらを繰り返すことで経路に沿って計時コースを走破する

「経路の急カーブでコースアウトする」という脅威に対しては、カーブ走行速度を抑えたり、オーバーラン検知でライントレースでの旋回量を多くし強めることで、コースからの逸脱の発生を緩和する

<以下は割愛する>

[経路に沿って難所コースを走破する]

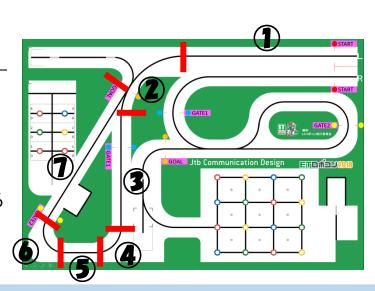
コース終端で停車する]

計時コースの区間分割 (Lコースの例)

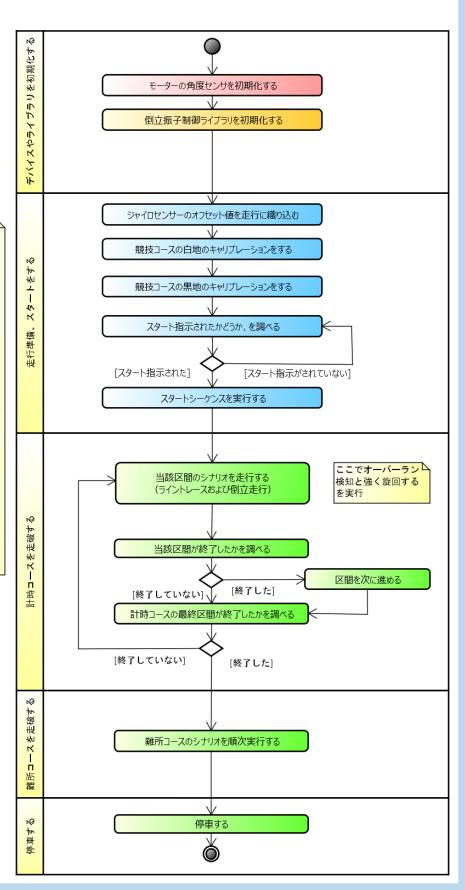
直線: ①、③、⑤、⑦

カーブ: ②緩、④急、⑥急

※各区間の経路距離は縮小コース図面を参考とする



<u>アクティビティ図</u>

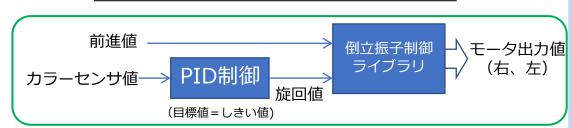


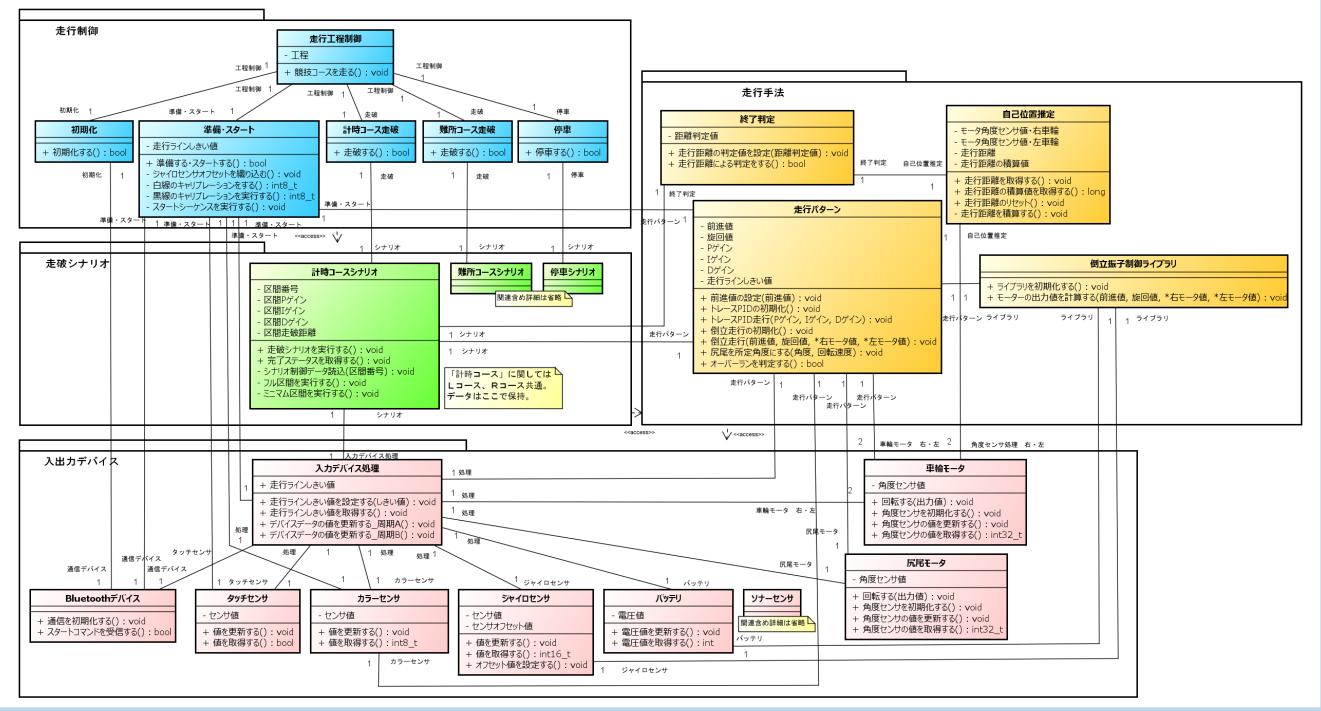
構造モデル

クラス図とパッケージ

走行制御走行全体の工程を管理、実行する走破シナリオ走破のためのシナリオ、データを処理する走行手法走破に必要な手法、判定を扱う入出力デバイスデータを含めデバイスに関わる処理を賄う

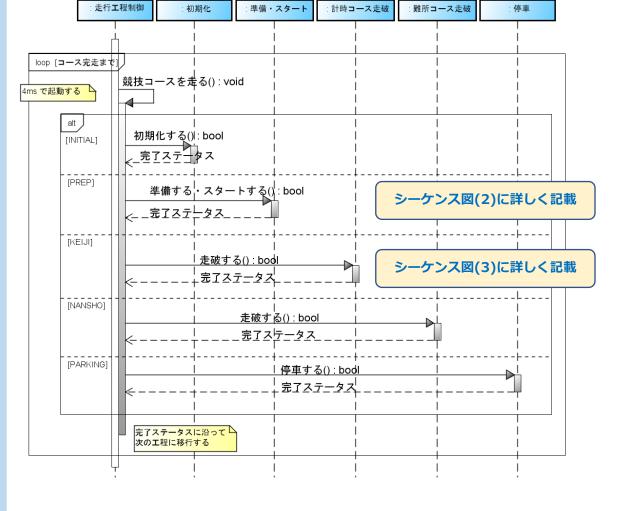
ライントレース・倒立走行の制御演算



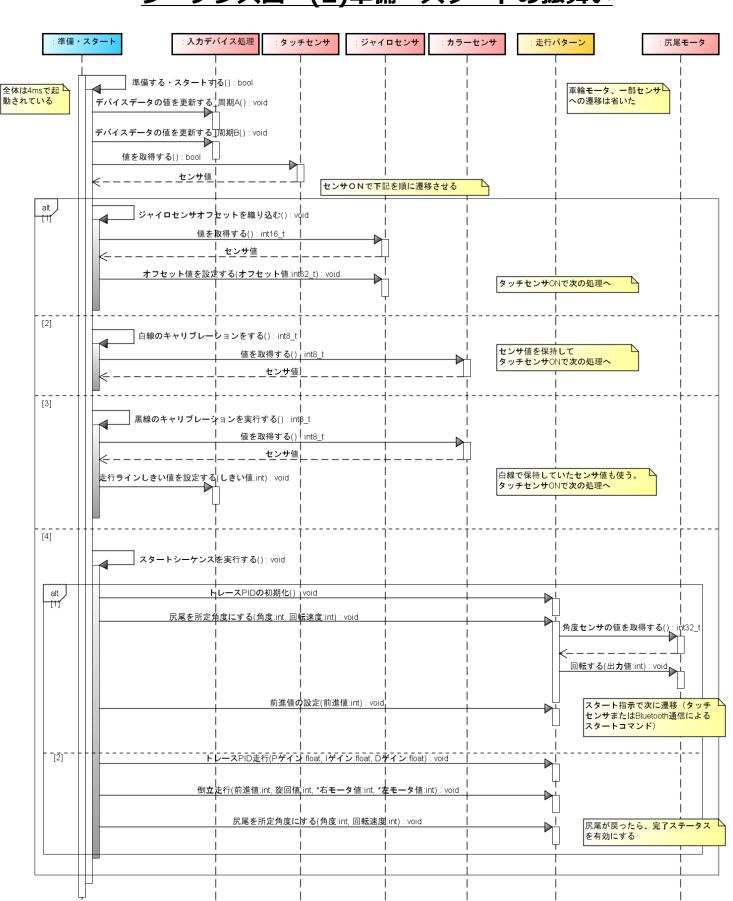


振舞いモデル(1)

シーケンス図 (1)走行制御の振舞い

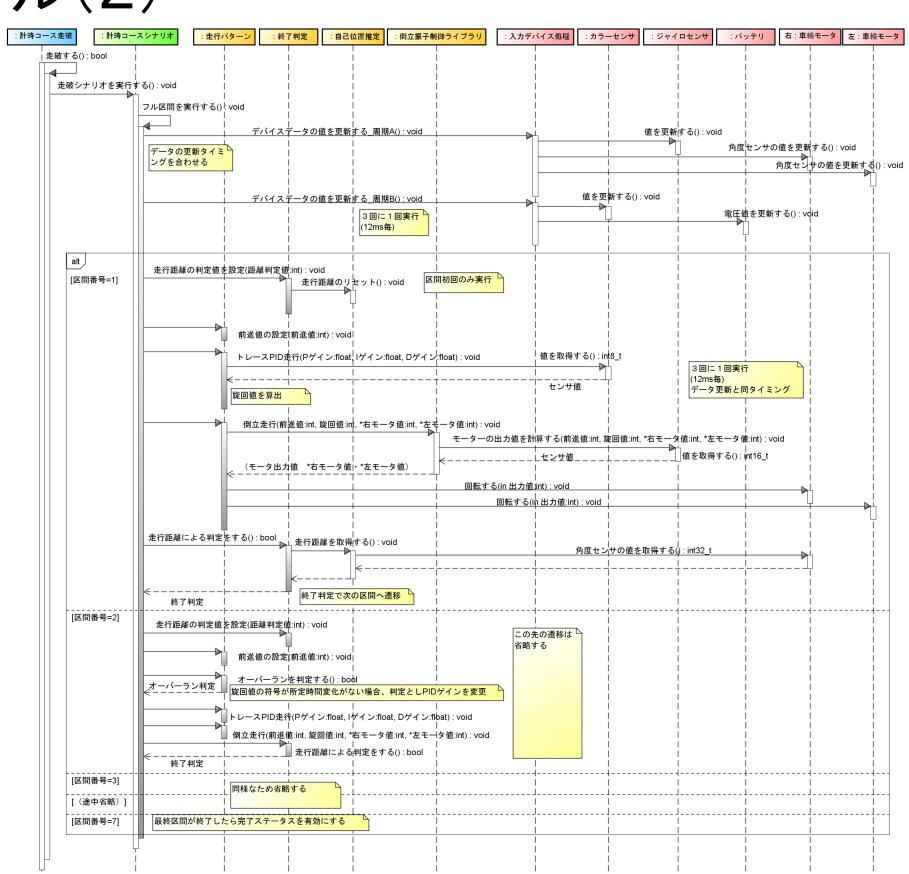


シーケンス図 (2)準備・スタートの振舞い



振舞いモデル(2)

<u>シーケンス図 (3)</u> <u>計時コース走破の振舞い</u>



工夫点

・課題:初歩的な開発環境の中で安全で効率的な走行テストをする

背景

- ・当チームは「個人」の集まりで、開発分担は単一ファイルレベルでのやり取りが主体。 (サーバあるいはリポジトリといった資源は、 スキル的にも準備、運用が難しい)
- ・開発者は分担分のみ動作するような構成で進めがちで、統合的なテストが結合等の関係で 不安定で非効率

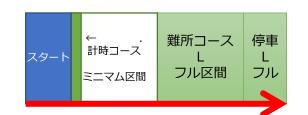
工夫とその効果

- ・走破シナリオのクラスを3つに分ける(開発分担を想定)「計時コース」「難所コース」「停車」それぞれのクラスに、
- "ミニマム区間を実行する"、"フル区間を実行する" という2つの主シーケンスのメソッドを用意し、 開発担当外のクラスでは"ミニマム区間"を動作させる
- *常に全体の工程を通した走行で開発を進めるため、 結合後のテスト等で不具合発生確率を抑えられる (Lコース、Rコース含めても同様と考える)
- ・スタートシーケンスは共通の走行制御クラスで動作させる
 - *全員が利用するため、リモート含め確実な動作の 求められるスタート部分の動作確認のN増しができる (ある程度安定的な動作は確立されている)

但し、走行パターンについては、結合の手間はある程度 残されるので、開発者間のコミュニケーションは重要。

開発者 B + C)

「停車(ガレージイン)開発」(Lコース)



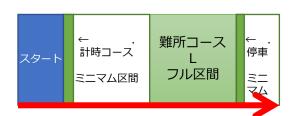
開発者 C)

「停車(ガレージイン)開発」(Lコース)



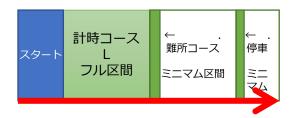
開発者 B)

「難所コース開発」(Lコース)



開発者 A)

「計時コース開発」(Lコース)



ベース)

スタート可のミニマム構成(コース不問)



コース全体走行のソフトウェア形態 (Lコース)

