

チーム紹介・目標・意気込み

卒業研究活動の一環として、一関高専 電気情報工学科と制御情報工学科の個性豊かな5年生7名で活動しています。

チーム名は学校の英称 (National Institute of Technology, Ichinoseki College) の略称より付けました。

一関高専としては久しぶりの参戦ですが、高専で学習した**モデリングとプログラミング**の両方の技術力を武器に、まずはコースの完走、そして**CS進出**を目標に頑張ります。

モデルの概要

[選択課題: コースを完走する]

選択した課題を実現するために、

- ・スタート指示管理
- ・走行距離に応じて走行設定を変えるシナリオトレース

の大きく2つの機能を備えた走行体を考え、モデルを作成しました。

機能を実現するために必要な部品を探した後に、それぞれの関係や振る舞いを示しています。

色使いを統一し、各図間で表記に相違がないようにすることで、全体を通して一貫性を持たせ、可読性を高めることをモデル作成における一番の重点としました。この表現技法についてはモデルを読み進める前に、ぜひ「**5) 工夫点**」を一読下さい。

モデルの構成

各ページにおける内容は以下の通りです。

ページ	タイトル	項目
1)	機能モデル	A.提供する機能 B.機能要件 C.部品の定義
2)	構造モデル	D.機能構成 E.部品の仕様定義
3)	振る舞いモデル	F.部品による機能実現の確認 G.システム全体の状態 H.システム全体の振る舞い
4)	振る舞いモデル	I.システムの振る舞い
5)	工夫点	ドキュメント: 一貫性を担保する表現技法

- 1) 機能モデル** : ユースケース図、記述によってシステムが利用者に提供する機能を要求分析し、処理の流れをアクティビティ図に示しました。その上で機能の実現に必要な部品を探し、オブジェクト図でまとめました。
- 2) 構造モデル** : クラス図を用いて部品の仕様を定義し、クラスの役割とまとまりを明確にしました。
- 3) 振る舞いモデル** : クラスに割り当てた働きの関係をコミュニケーション図に示した。またシステムの状態遷移をステートマシン図に示しシステム全体の振舞いをシーケンス図で示しました。
- 4) 振る舞いモデル** : シーケンス図を用いて処理の流れを時間軸で考え、機能の動作を確認しました。
- 5) 工夫点** : 機能を実現する上でどのように可読性の高いモデルを作るかをまとめました。

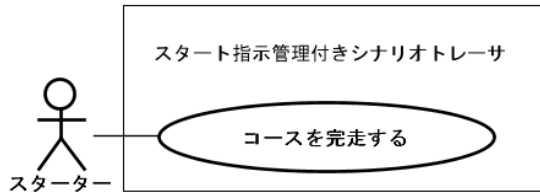
1) 機能モデル



teamNITIC

A.提供する機能

スターターにコースを完走する機能を提供する。



B.機能要件

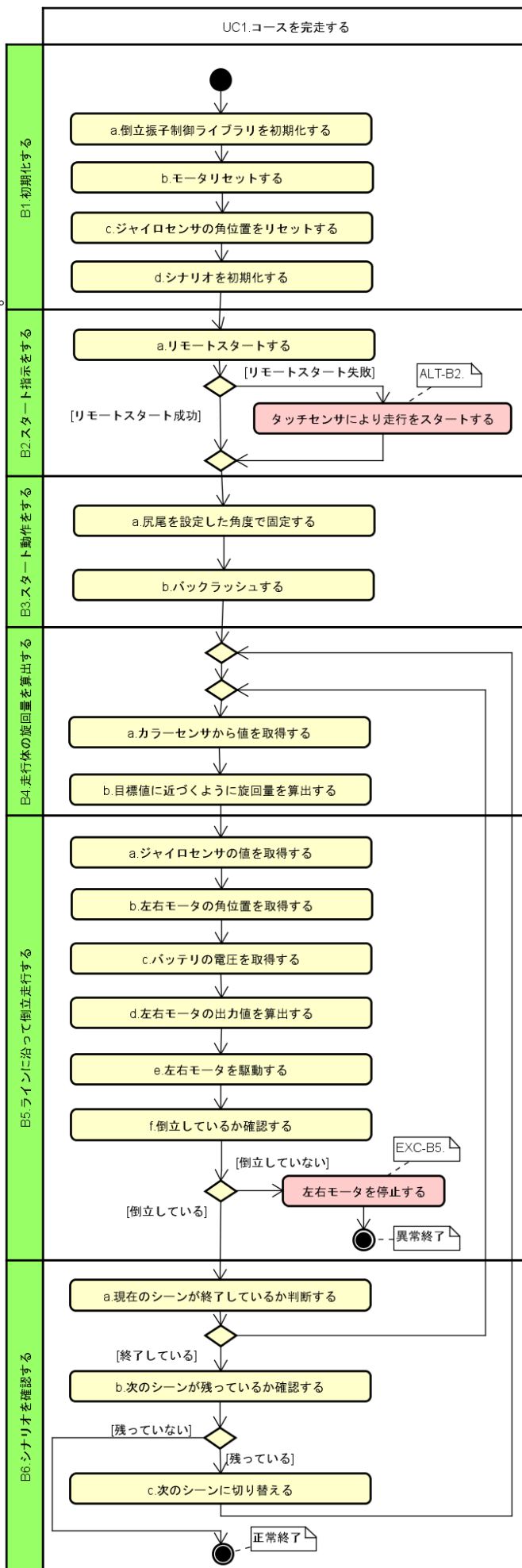
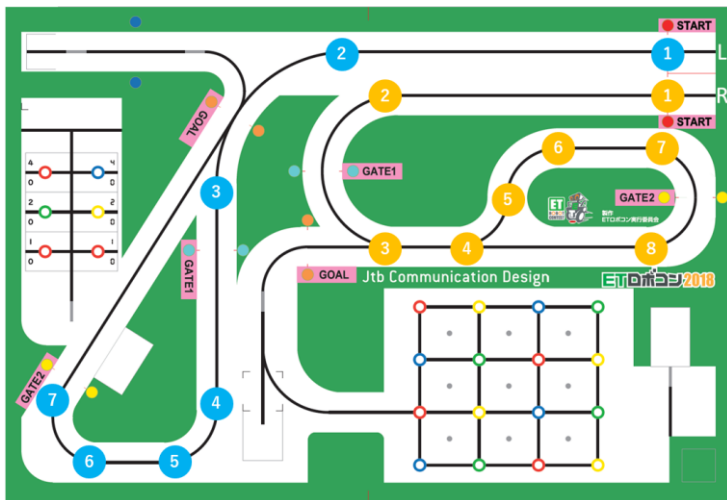
機能を実現する方法と処理順序をユースケース記述とアクティビティ図で示す。

項目	内容
ユースケース	コースを完走する
概要	LコースまたはRコースをライントレースしてゴールまで走行する
アクター	スターター
事前条件	キャリブレーションし、走行体がスタートエリアのライン上に完全停止状態で設置されている
事後条件	ゴールゲートを通過する
基本系列	B1.初期化する B2.スタート指示をする B3.スタート動作をする B4.走行体の旋回量を計算する B5.ラインに沿って倒立走行する B6.シナリオを確認する B7.基本系列B4.～B6.を繰り返す
代替系列	ALT-B2.基本系列B2.においてリモートスタートが失敗した場合タッチセンサにより走行を開始する
例外系列	EXC-B5.動作途中にて走行体が転倒した場合、動作を停止する

補足

シーンとシナリオについて

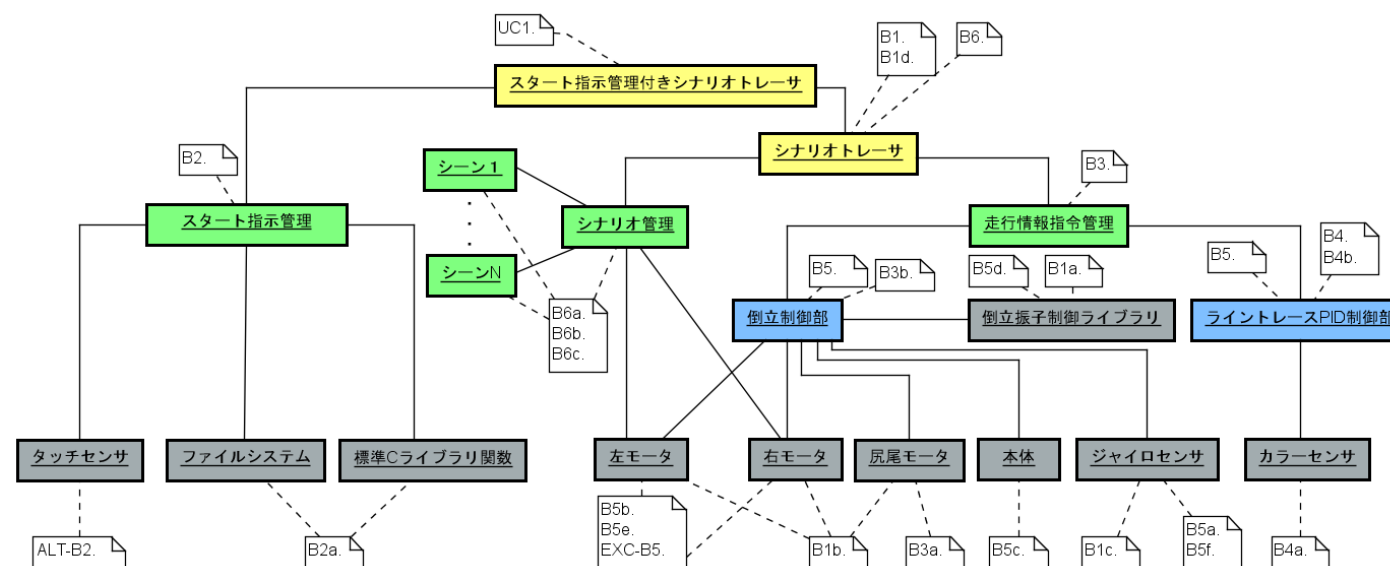
コースを以下のように分割し、区間ごとに走行設定を変える。
区間毎の走行設定を「シーン」とし、実際に走行させる上での順序をまとめたものを「シナリオ」とする。次の図の番号はシーンが切り替わるポイントである。



C.部品の定義

機能を実現するために必要な部品を以下の表に示し,各部品の責務をオブジェクト図に示す。

	「働き」や「情報」	部品の候補
UC1.	コースを完走する	スタート指示管理付きシナリオトレーサ
B1.	初期化する	シナリオトレーサ
B1a.	倒立振子制御ライブラリを初期化する	倒立振子制御ライブラリ
B1b.	モータリセットをする	左モータ、右モータ、尻尾モータ
B1c.	ジャイロセンサの角位置をリセットする	ジャイロセンサ
B1d.	シナリオを初期化する	シナリオトレーサ
B2.	スタート指示をする	スタート指示管理
B2a.	リモートスタートする	ファイルシステム、標準Cライブラリ関数
ALT-B2.	タッチセンサにより走行をスタートする	タッチセンサ
B3.	スタート動作をする	走行情報指令管理
B3a.	尻尾を設定した角度で固定する	尻尾モータ
B3b.	バックラッシュする	倒立制御部
B4.	走行体の旋回量を算出する	ライントレースPID制御部
B4a.	カラーセンサから値を取得する	カラーセンサ
B4b.	目標値に近づくように旋回量を算出する	ライントレースPID制御部
B5.	ラインに沿って倒立走行する	倒立制御部、ライントレースPID制御部
B5a.	ジャイロセンサの値を取得する	ジャイロセンサ
B5b.	左右モータの角位置を取得する	左モータ、右モータ
B5c.	バッテリーの電圧を取得する	本体 (インテリジェントブロック)
B5d.	左右モータの出力値を算出する	倒立振子制御ライブラリ
B5e.	左右モータを駆動する	左モータ、右モータ
B5f.	倒立しているか確認する	ジャイロセンサ
EXC-B5.	左右モータを停止する	左モータ、右モータ
B6.	シナリオを確認する	シナリオトレーサ
B6a.	現在のシーンが終了しているか判断する	シナリオ管理、シーン
B6b.	次のシーンが残っているか確認する	シナリオ管理、シーン
B6c.	次のシーンに切り替える	シナリオ管理、シーン



このオブジェクト図に基づいて構造モデルを作成する。

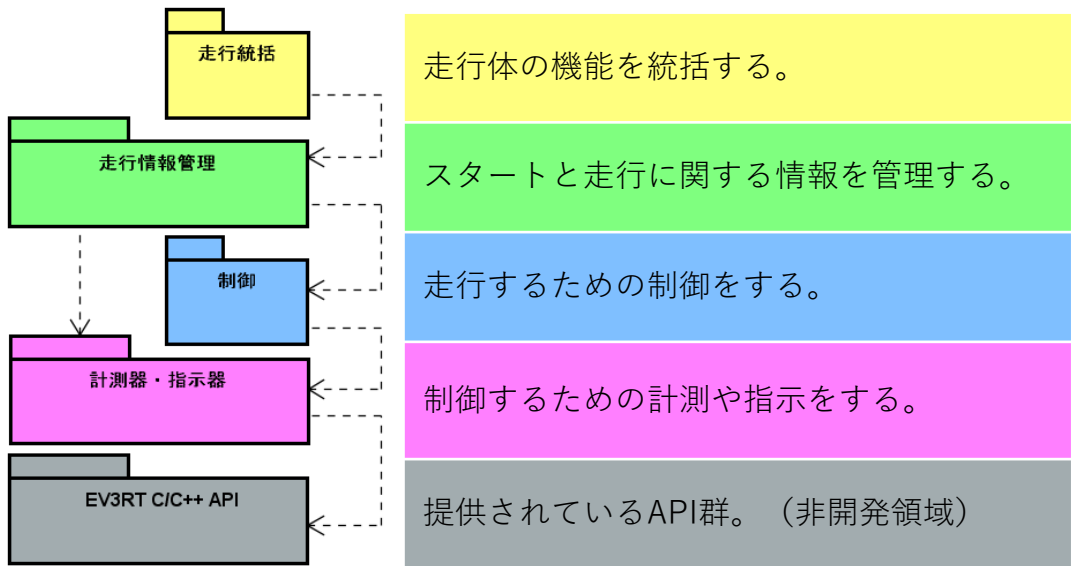
2) 構造モデル



teamNITIC

D.機能構成

機能の構造を責務に応じて階層化する。
この際、APIへ計測や指示を行う階層を新たに設ける。



走行体の機能を統括する。

スタートと走行に関する情報を管理する。

走行するための制御をする。

制御するための計測や指示をする。

提供されているAPI群。(非開発領域)

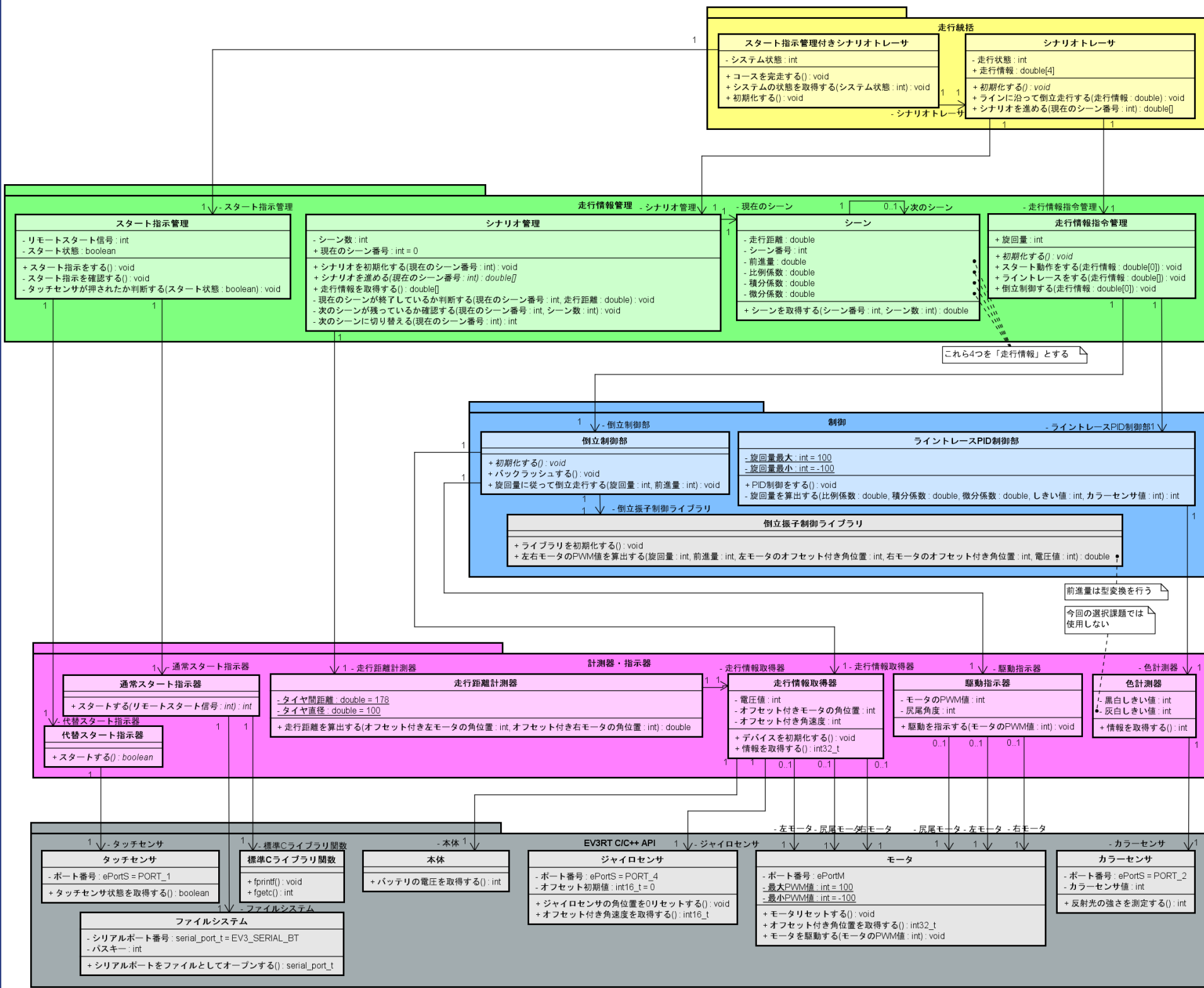
補足 各パッケージ内のクラス役割

各パッケージ内のクラスの役割を示す。パッケージやクラスは色で対応している。

クラス名	説明
スタート指示管理付きシナリオトレーサ	走行体のスタートや走行の統括をする
シナリオトレーサ	走行体の走行に関する統括をする
スタート指示管理	スタートの指示を管理する
走行情報指令管理	走行体の走行指示を管理する
シナリオ管理	シーンの組み合わせに基づくシナリオを管理する
シーン	区間別の走行情報を保持する
ライントレースライントレースPID制御部	PID制御をする
倒立制御部	倒立制御をする
倒立振子制御ライブラリ	左右モータの出力値を算出する
通常スタート指示器	リモートによる通常スタートを指示する
代替スタート指示器	タッチセンサによる代替スタートを指示をする
走行距離計測器	走行体の走行距離を計測する
走行情報取得器	走行体の走行に必要な情報を取得する
駆動指示器	モータに駆動指示する
色計測器	色を計測する
標準Cライブラリ関数	Bluetooth通信をする
ファイルシステム	
タッチセンサ	押下状態を取得する
本体	バッテリーの電圧を取得する
ジャイロセンサ	角速度を取得する、角位置をリセットする
モータ	駆動する、角位置を取得する
カラーセンサ	反射光を測定する

E.部品の仕様定義

機能の構造をクラス図に示す。ここから状態遷移についてはH.に、時系列による振る舞いはI.に示す。



このクラス図に基づいて振る舞いモデルを作成する。

3) 振る舞いモデル

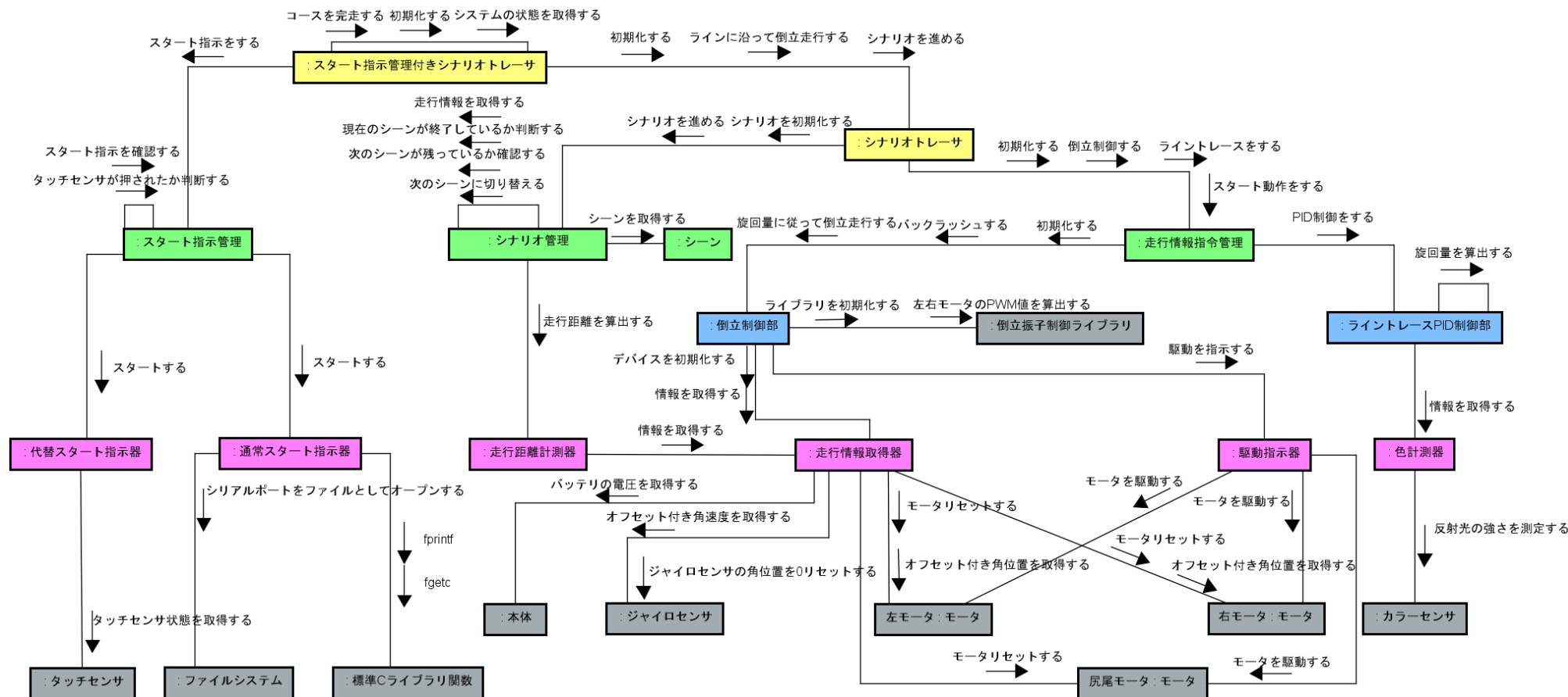


teamNITIC

F.部品による機能実現の確認

部品同士の関係をコミュニケーション図に示す。なお、引数と戻り値については省略する。
また、コミュニケーション図から各クラスの操作数を確認し、右表にまとめる。

以下の表は倒立振り制御ライブラリとEV3RT C/C++ APIは考慮しない。

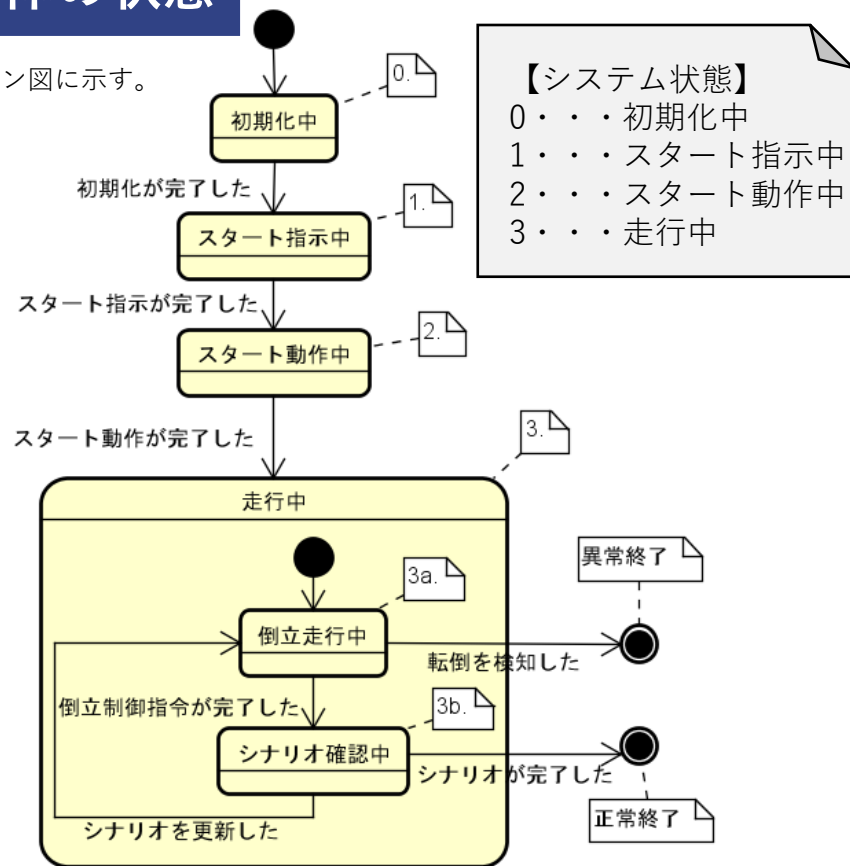


クラス名	操作数
スタート指示管理付きシナリオトレーサ	3
シナリオトレーサ	3
スタート指示管理	3
走行情報指令管理	4
シナリオ管理	6
シーン	1
ライントレースライントレースPID制御部	2
倒立制御部	3
通常スタート指示器	1
代替スタート指示器	1
走行距離計測器	1
走行情報取得器	2
駆動指示器	1
色計測器	1

これによりクラス図との整合性が取れる。
1つのクラスの操作数が6以下になるように構成する。

G.システム全体の状態

機能の状態をステートマシン図に示す。

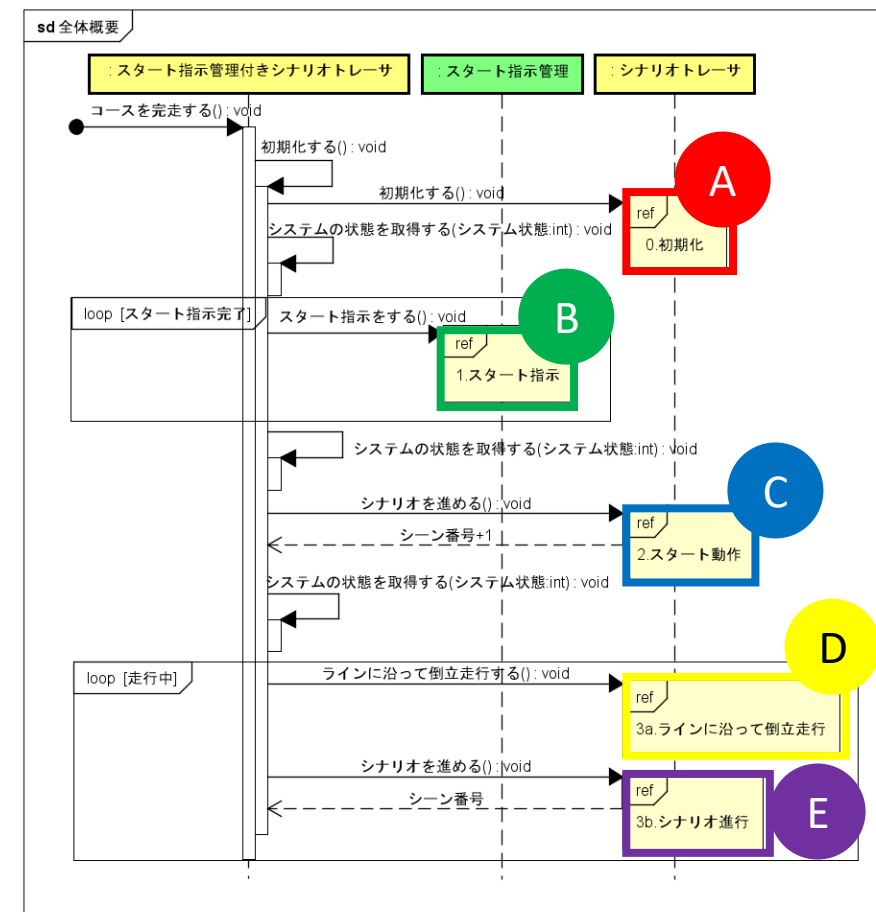


H.システム全体の振る舞い

全体の動作をシーケンス図に示す。

右図はG.で示したステートマシン図の流れを具体的に示したものであり、ref.で表した部分は次ページを参照。
記号や色は次ページと対応している。

ステートマシン図		シーケンス図
初期化中	0.	初期化
スタート指示中	1.	スタート指示
スタート動作中	2.	スタート動作
倒立走行中	3a.	ラインに沿って倒立走行
シナリオ確認中	3b.	シナリオ進行



4) 振る舞いモデル

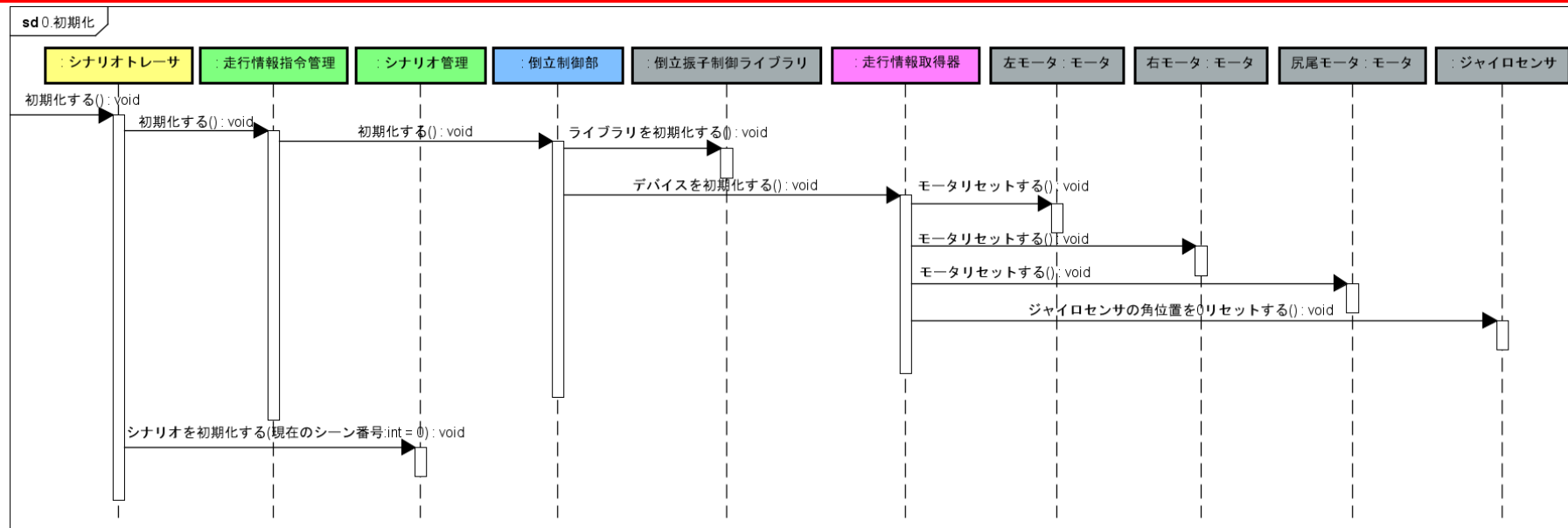


teamNITIC

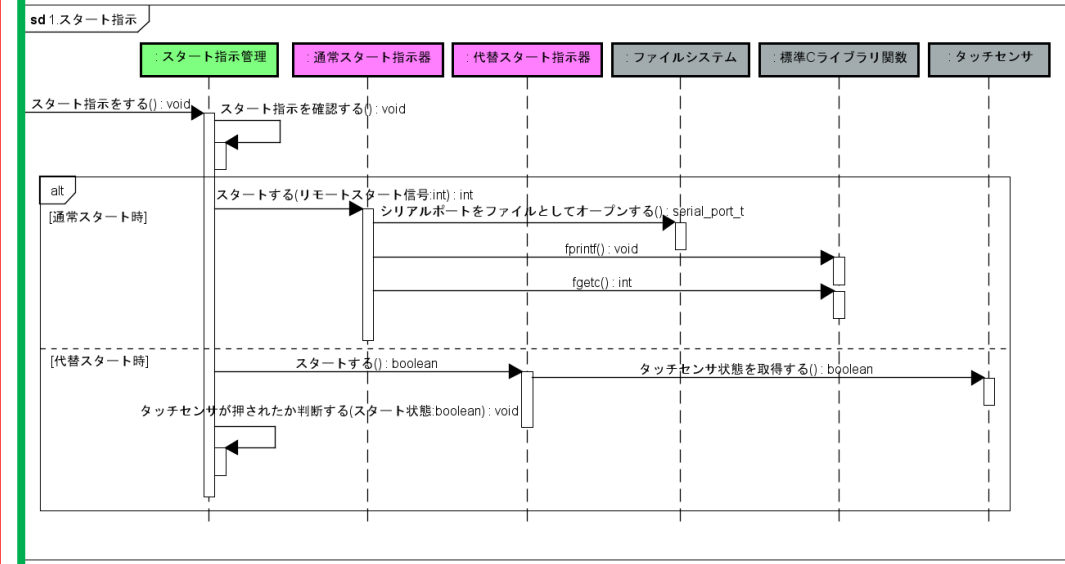
1. システムの振る舞い

前ページに示した全体概要シーケンス図をさらに詳しく表す。記号や色は前ページと対応している。
なお、倒立振り制御ライブラリ及びEV3RT C/C++ APIからの戻り値については省略する。

A

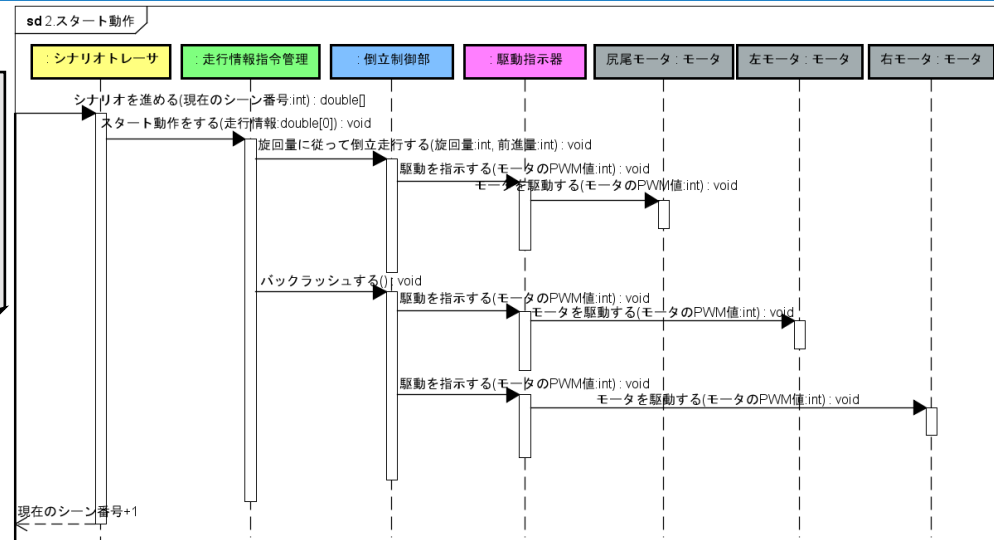


B

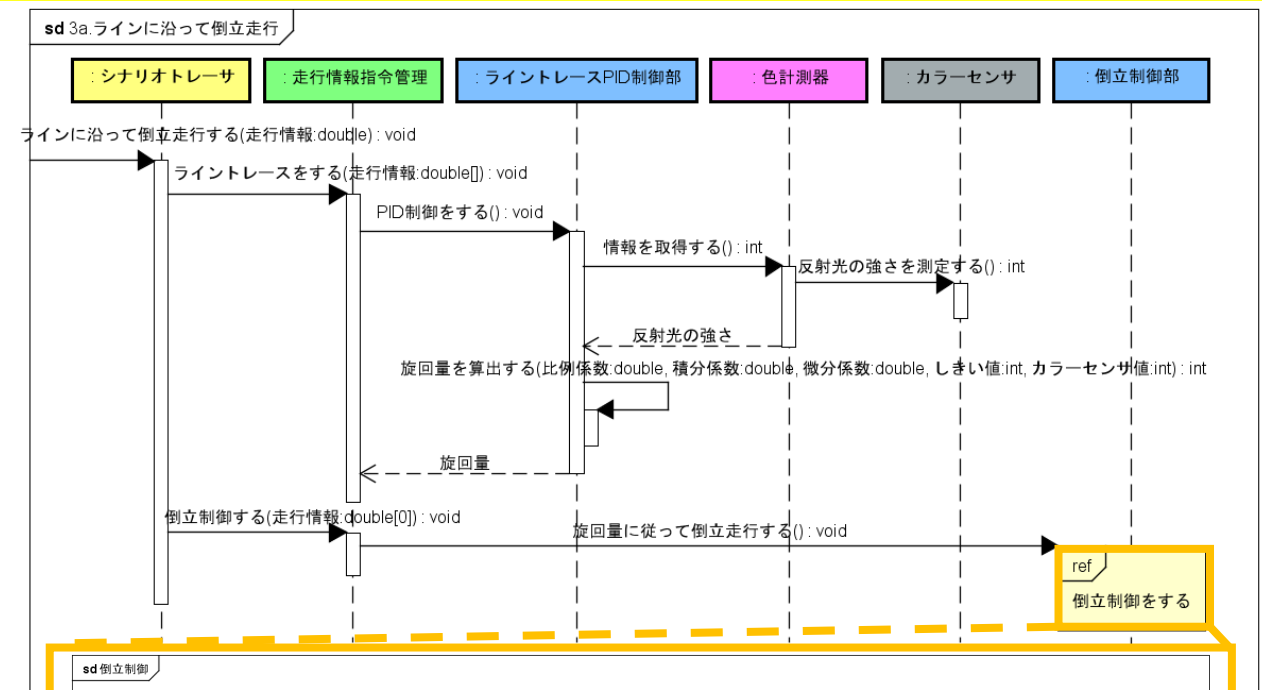


C

スタート動作は
シナリオ番号:0
として実行する。



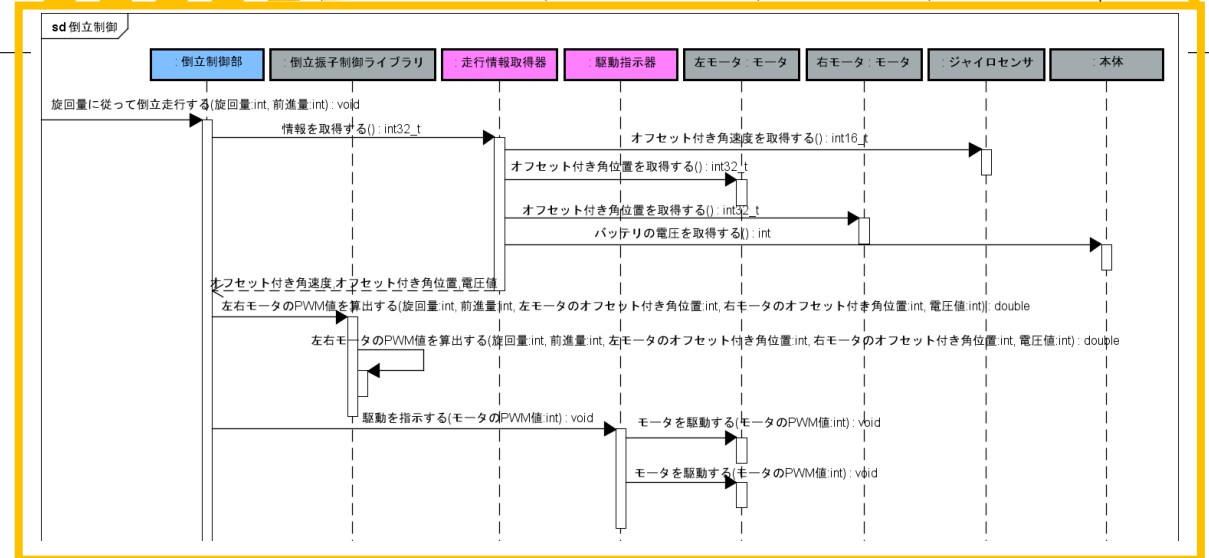
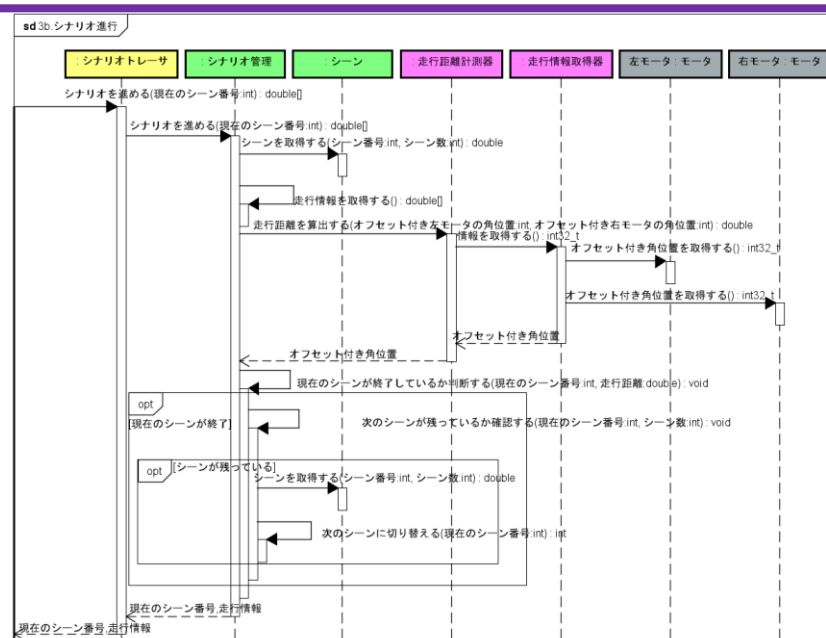
D



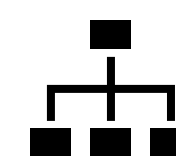
E

コースに応じて
シナリオ番号:1
から実行する。

※コースの区分については
「1) 機能モデル-補足」を参照。



> 誰が読んでもわかるモデル作成とは？



モデルは提供する機能を明確に要求分析し、プログラムを書く上での設計書となる。
複雑かつ可読性の低いモデルは開発を効率よく進めることができず、信頼性を失っていると考ええる。
今回モデルを作成する上で注意した点を示す。

1. レイアウトの統一

ヘッダーやタイトル、枠の色をページごとに変えることなく全体で統一することで読み手がモデル以外に気を取られないようにした。

2. 自然な誘導

図やページが変わるごとに前の内容を見返すということを極力なくすために、INDEX化や色分け、アイコンをつけるといった方法をとった。
具体的な実践内容を以下に示す。

・「1) 機能モデル」

部品の定義にてアクティビティ図で示した処理の項目に**INDEX**（例：B1a.）をつけた。

その後のオブジェクト図にはINDEX番号で示すことで省スペース化した。

・「2) 構造モデル」

パッケージごとに**色**を使い分けた。色は最上部のパッケージから色相環時計回り順に「黄→緑→青→紫」を使用した。

なお、最下層に灰色を使用した。これは非開発領域であり開発領域と区別するために意図的に行った。

このページの説明で同じ色を使用したため、D.～F.の項目間を読みやすくした。

・「3) 4) 振る舞いモデル」

ステートマシン図と全体概要シーケンス図をINDEX番号で対応させたことで処理の流れをわかりやすく示した。

また、シーケンス図がページをまたぐ構成になったため、記号による**アイコン**を用いた。

これにより読みたい図へスムーズに誘導できるようにした。

これらの技法を取り入れることにより一貫性を担保し、可読性を高め、開発をスムーズにできると考える。