9/2(金) 10:20〜11:30 セッションs3b

ここから始める箱庭の活用



久保秋 真（チェンジビジョン）
細合 晋太郎（東京大学）
高瀬 英希（東京大学）

演習の準備（お詫び）

- ハンズオンと予告しましたが時間が足りませんでした
  - ダウンロードだけで割と時間がかかります
- 事前にある程度準備しておけば一緒に進められます
  - （あるいは、事後に資料をみながらやってみるとよいでしょう）
  - WSL2、Docker、Unity、GitHubの利用準備
  - GitHubのhakoniwa-single_robotの実施
    - https://github.com/toppers/hakoniwa-single_robot/
    - 演習で使うデータのダウンロード（https://bit.ly/3QPzJm6）
  - GitHubのhakoniwa-single_robotとhakonwia-ros2simの実施
    - https://github.com/toppers/hakoniwa-ros2sim/

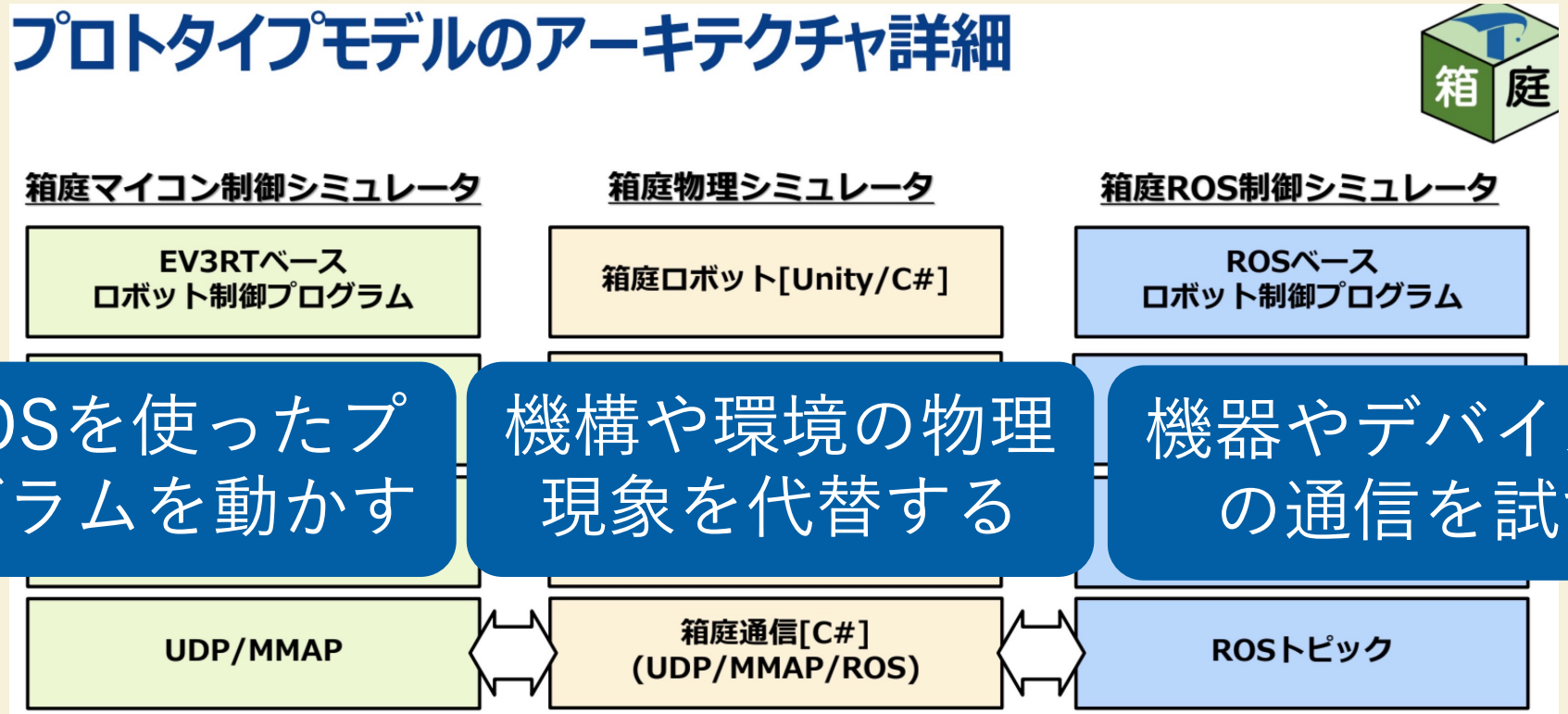
このセッションの概要

- 使う側からみた箱庭にできること
  - 箱庭は、なにで、どんなことに使えるか
- どんなふうに使っているか（久保秋）
  - 自動運搬システム（RTOSプログラムによる機器の制御）
  - 信号に従って走行する列車（複数の機器を一緒に動作させる）
- 自分たちのしごとで使うには（細合）
  - みなさんは、どんなことに使えそうか
  - みなさんが使うには、どんな準備や作業が必要か

久保秋　真（くぼあき　しん）

- 株式会社チェンジビジョン勤務
- モデリングツール「Astah*」の開発支援
- UMLやSysMLを使ったモデリングのコンサルティングや技術教育の開発・講師
- ETロボコンの本部モデル審査員
- アジャイル開発とモデル駆動開発（MDD）の双方に関心
- 情報処理学会、日本ソフトウェア科学会各会員
- 早稲田大学理工学術院、日本大学生産工学部、関東学院大学理工学部
  各非常勤講師
- トップエスイー講師、スマートエスイー講師
- 日本雨女雨男協会IT本部長
- 日本あんこ協会あんバサダー

使う側から見た箱庭にできること



RTOSを使ったプログラムを動かす

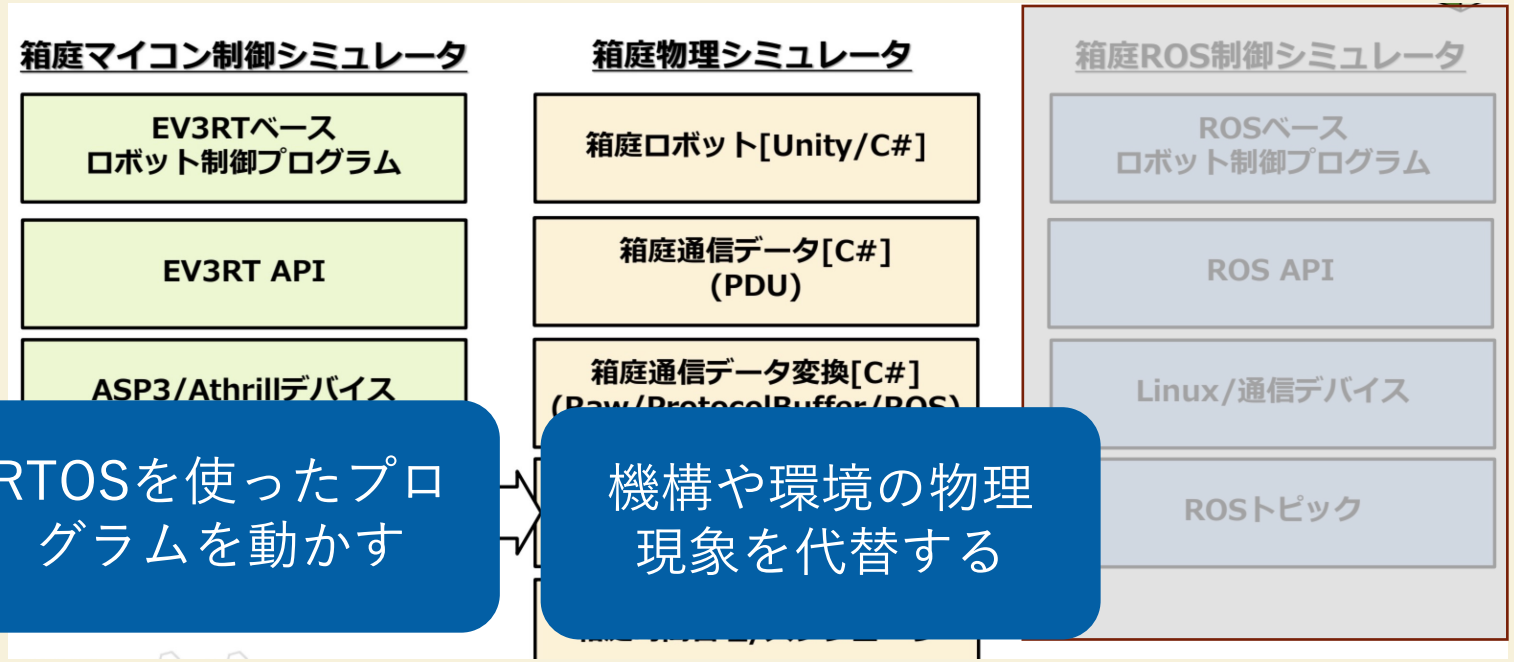
機構や環境の物理現象を代替する

機器やデバイス間の通信を試す

実機を必要とせず、 PCの上で
さまざまなシミュレータを組み合わせて
いろいろなことが試せる

【その1】

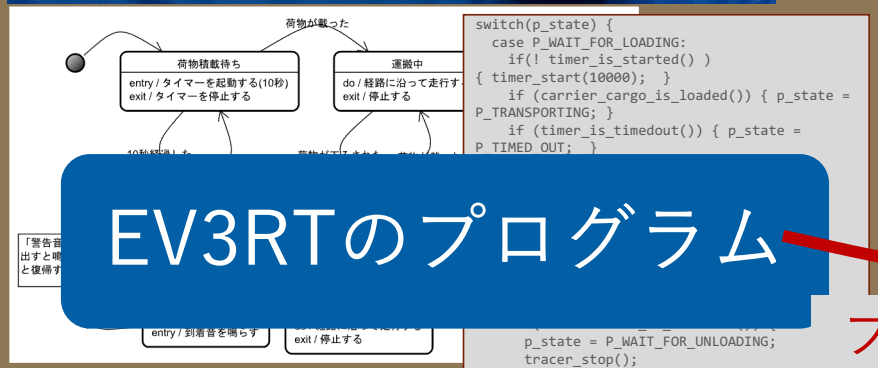
マイコン制御シミュレータ＋物理シミュレータ



事例）自動搬送システム

リアルワールド

EV3のロボット
と運搬コースや荷物

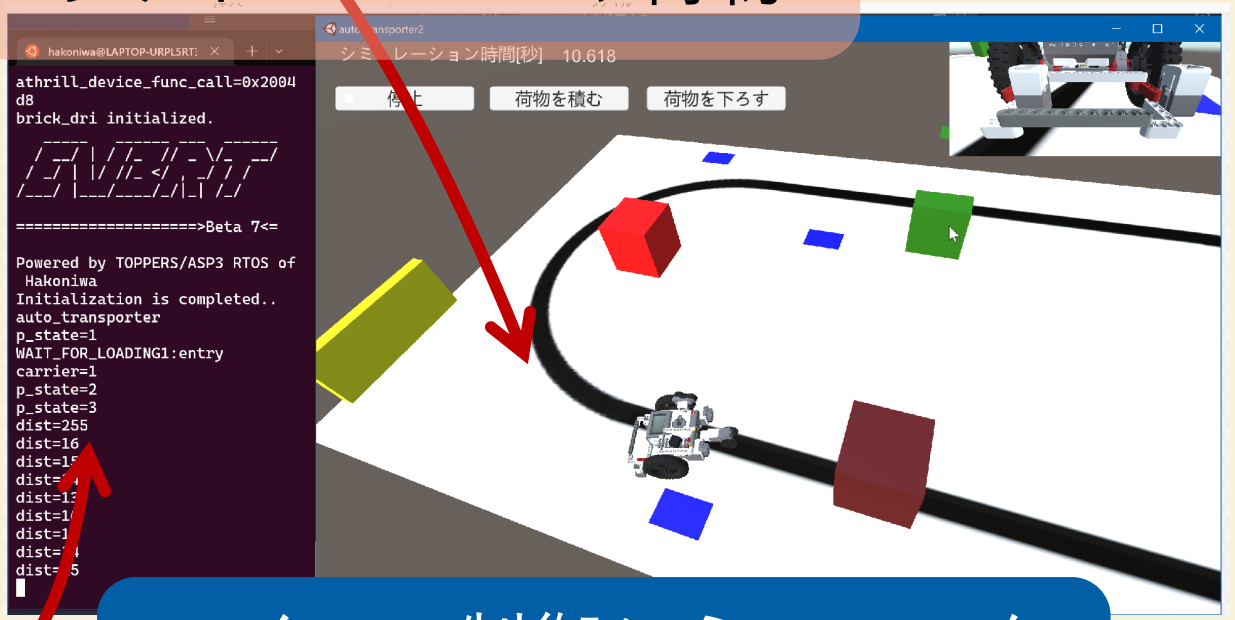



```
switch(p_state) {
  case P_WAIT_FOR_LOADING:
    if(! timer_is_started() )
{ timer_start(10000);  }
    if (carrier_cargo_is_loaded()) { p_state =
P_TRANSPORTING; }
    if (timer_is_timedout()) { p_state =
P_TIMED_OUT;  }
    p_state = P_WAIT_FOR_UNLOADING;
    tracer_stop();
    horn_arrived();
  }
```

EV3RTのプログラム

プログラムは
ほぼ共通

物理シミュレータ上のロボットとコースや荷物



マイコン制御シミュレータ上のプログラム

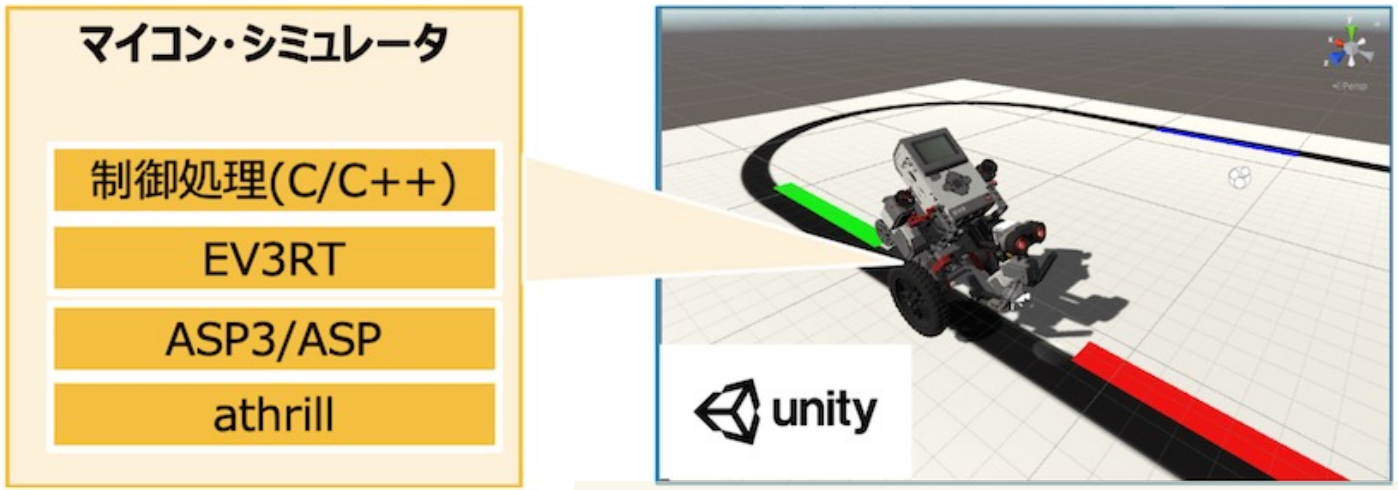
導入手順（1）

- マイコン制御シミュレーションの環境を導入する

箱庭プロトタイプモデルA：単体ロボット向けシミュレータ

TOPPERSプロジェクト箱庭WGでは，IoT／クラウドロボティクス時代の仮想シミュレーション環境である『箱庭』の研究開発を進めています．本活動での狙いやコンセプトを実証するために，プロトタイプモデルを実装してひろく公開しています．

本リポジトリでは，プロトタイプモデルのひとつである「単体ロボット向けシミュレータ」について，WSL 2とDockerを用いて最小の構成と手順で試行できる実行環境を提供しています．本プロトタイプモデルでは，ETロボコンを題材としており，組込みマイコンシミュレータathrill上での制御プログラムの振る舞いを，Unity上でのロボット上の挙動と連携させて検証を進めることができます．

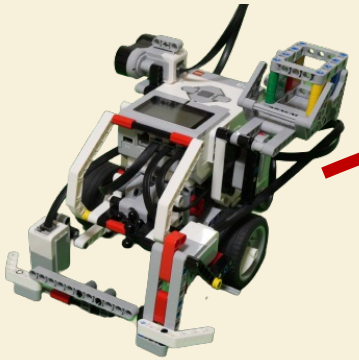


箱庭プロトタイプモデルA：単体ロボット向けシミュレータ
https://github.com/toppers/hakoniwa-single_robot/

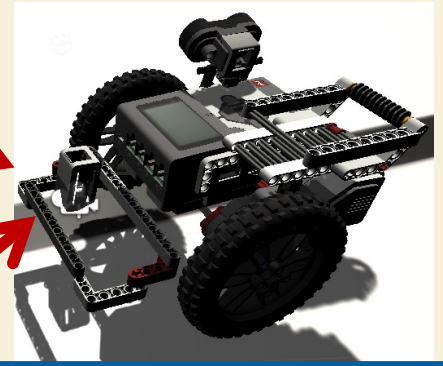
必要なもの
- Windows10～
- GitHub
- WSL2
- Docker
- Unity
- Blender

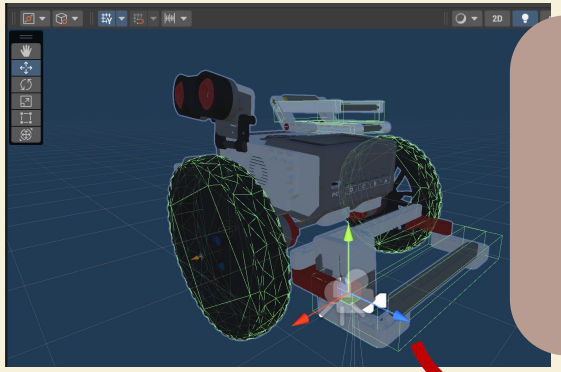
導入手順（2）

- 物理シミュレータ用のロボットを作る



Unity用のデータ作成
（CAD/Blenderなど）





物理動作のための
オブジェクトの設定
（可動部、センサなど）

Unity上で動作する
仮想ロボット

あらかじめ作成したプログラムと環境は下記から入手できる

https://bit.ly/3QPzJm6

導入手順（3）

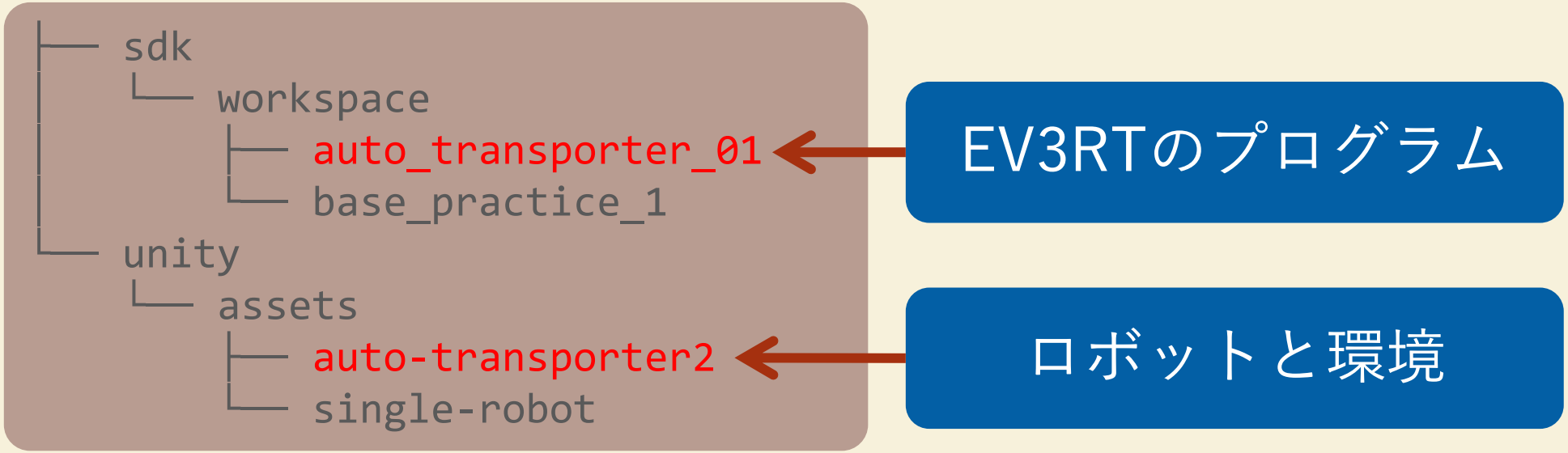
- 物理シミュレータ用の環境を作る

シミュレータ操作のためのUI（ボタンや表示）、カメラなど

運搬コース、荷物、配達先等のステーションなど

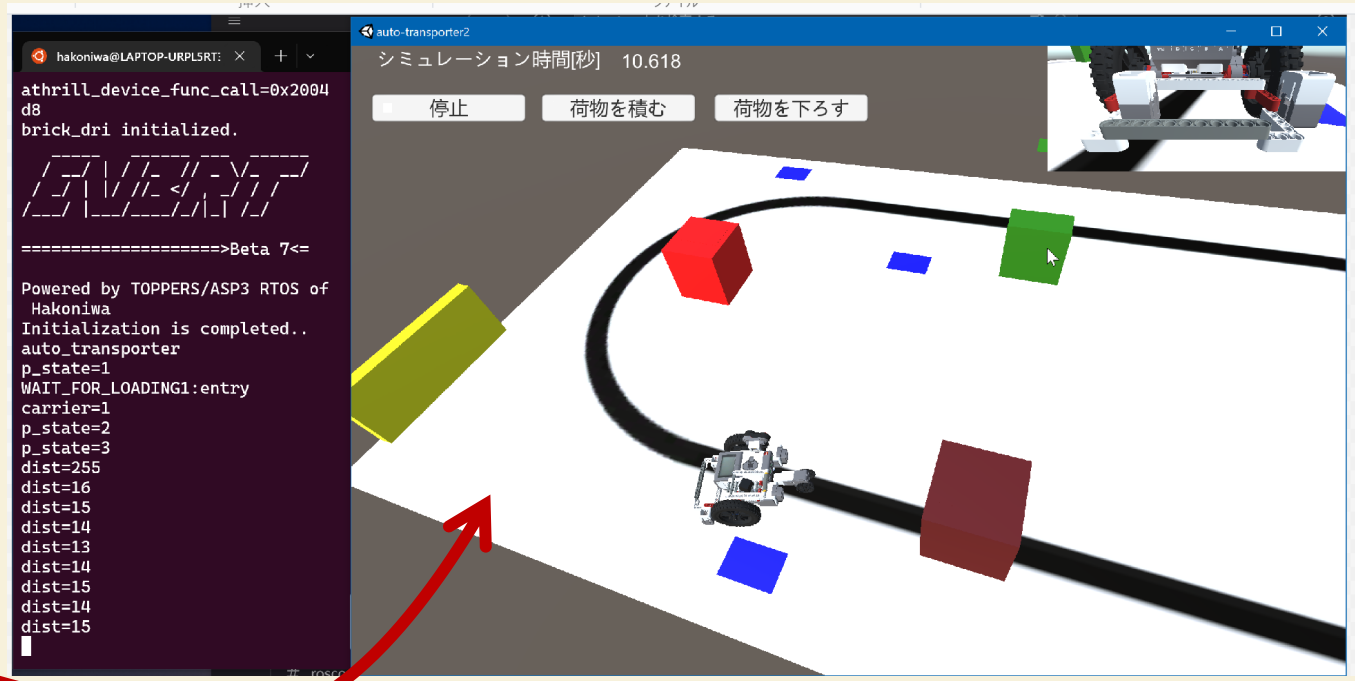
導入手順（4）

- ロボットと環境をアプリ化する
  - Windows用だと、.exeファイルとデータセットができる
  - アプリ化しないでUnityエディタで実行する方法もある
- プログラムとアプリを配置する



シミュレーションの実行

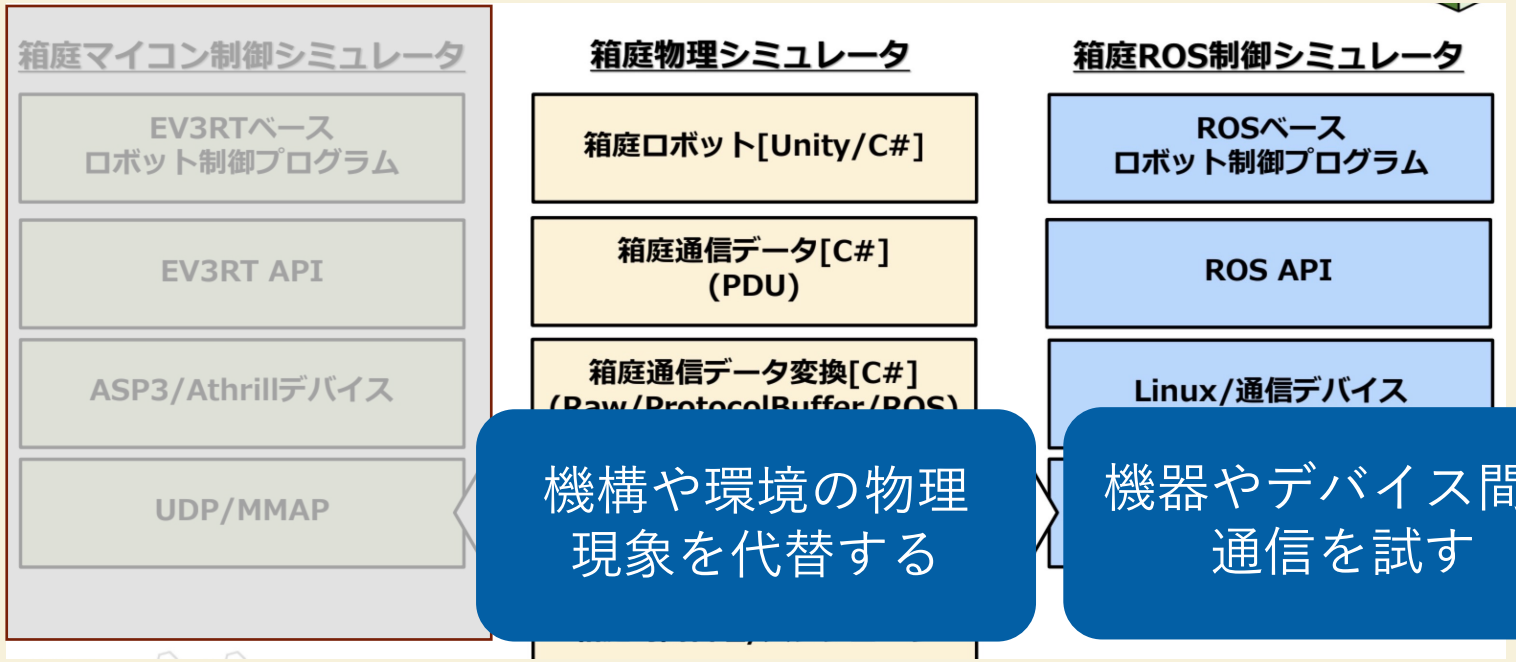
- 端末A
  - 開発用dockerコンテナを起動
  - Unityアプリが起動すると、ログが表示される
- 端末B
  - プログラムのビルド
- 端末C
  - Unityアプリの起動
  - プログラムとが起動し、シミュレーションが始まる



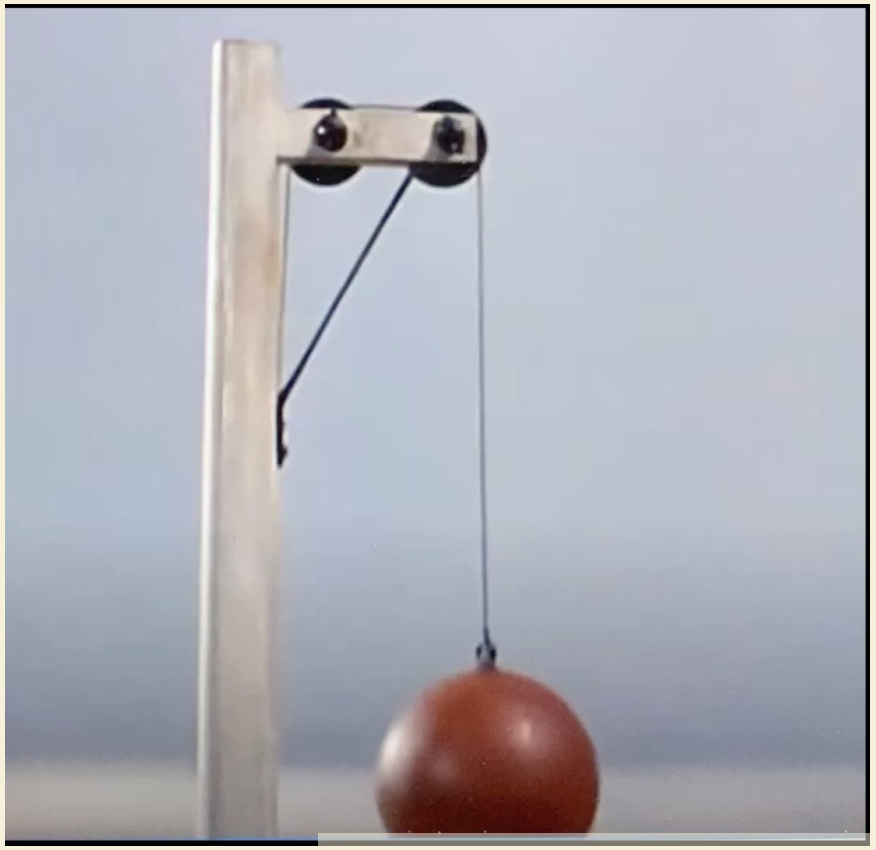
AutoTransporter_720.mov

【その2】

物理シミュレータ
＋ROS制御シミュレータ



昔の信号機やイギリスの信号機の例



The Signal Engineers - 1962 - Electrical Engineering on the Railwayより抜粋

https://www.youtube.com/watch?v=6Oc_50DnGG0

https://farm8.staticflickr.com/7085/13465755913_94348f4cea_z.jpg

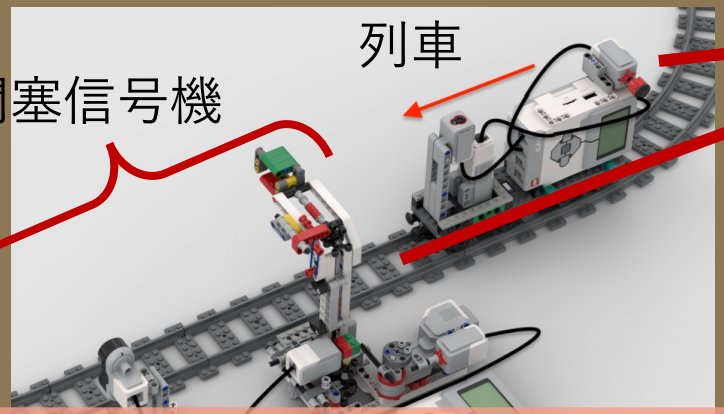
まねをして作った信号機の実機モデル

- YouTube
  - https://youtu.be/k168I_5-GNs
  - https://youtu.be/71gXzo7RDiw
- GitHub
  - https://github.com/kuboaki/ev3_train/
  - プログラム、LEGOのCADデータ、モデル図など

🤔 都心部では、鉄道の信号はGPS等を使った列車上の電子信号に変わりつつある。
一方で、運転士の目視による閉塞信号方式も、まだたくさん使われている

事例）信号機に従って走行する列車

リアルワールド



EV3の列車と信号機と線路

信号機のプログラム

列車のプログラム

物理シミュレータ上の
列車と信号機と線路

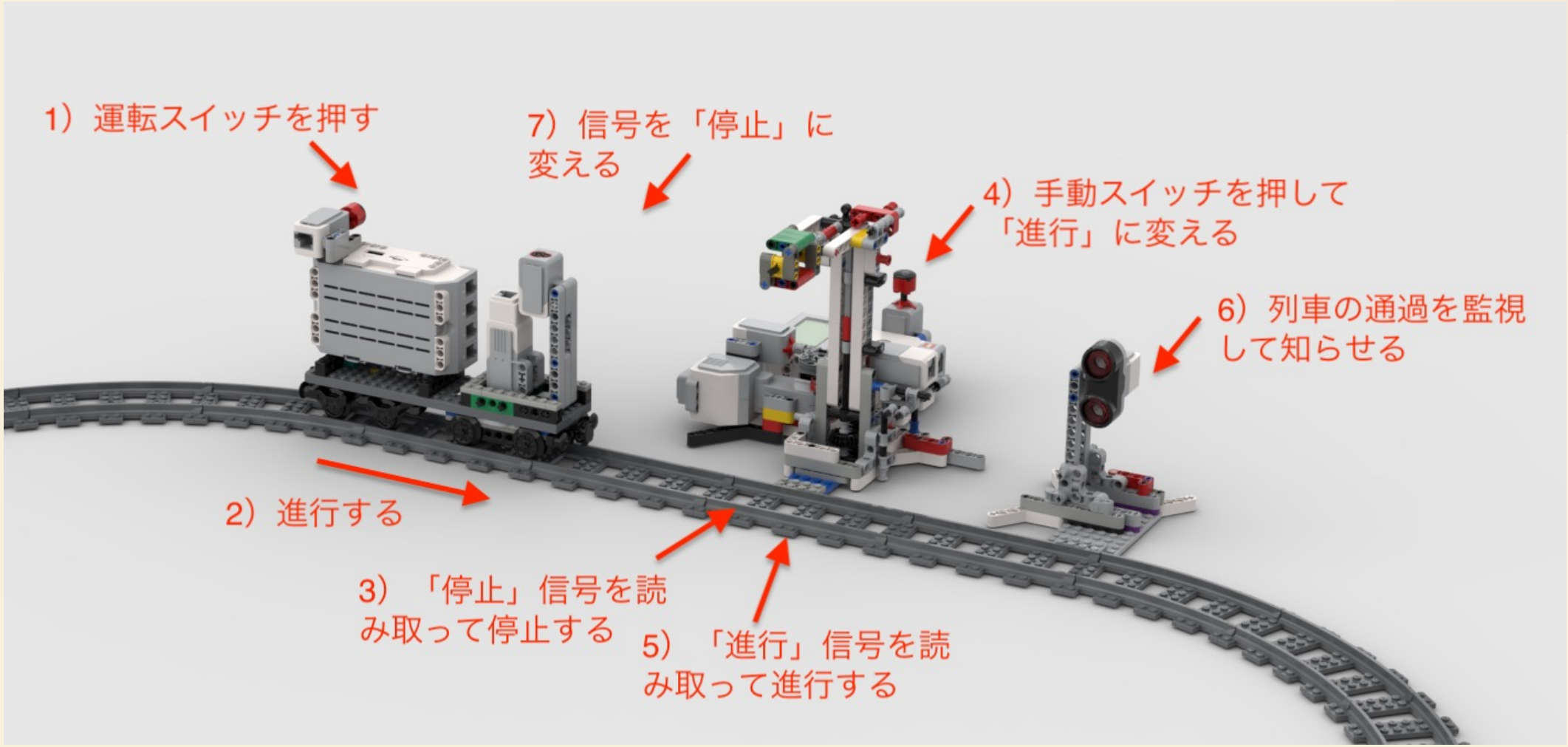
ROSベースの信号機のプログラム

ROSベースの列車のプログラム

サンプルシステムの動作シナリオ（1）

- エンドレスの線路に閉塞信号機がついている
  - 車上のカラーセンサーを使って、信号機が示すパレットの色を読み取る方式
- 列車は、エンドレスの線路上を周回する
- 信号が「停止」なら、列車は停止する
- 信号が「進行」に変わると、列車は走行する
- 信号機は、列車通過監視部で列車の通過を認識すると、信号を「停止」に変える
- 列車は、信号機の手前の「徐行」信号を通過すると、徐行する

サンプルシステムの動作シナリオ（2）

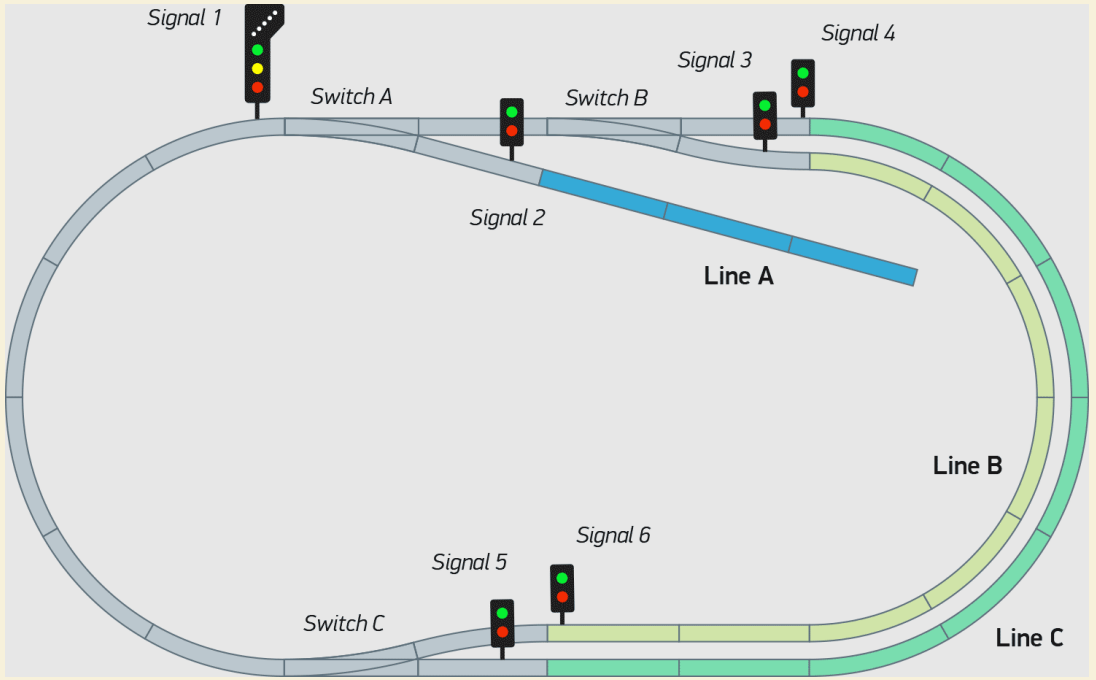


次は、信号や列車を増やしたいが…

- 実機では場所や機材の手配が大変
  - 信号機1機にEV3が1セット…
  - 列車1編成にEV3が1セット…
  - 線路を引く場所が…
- そうだ！

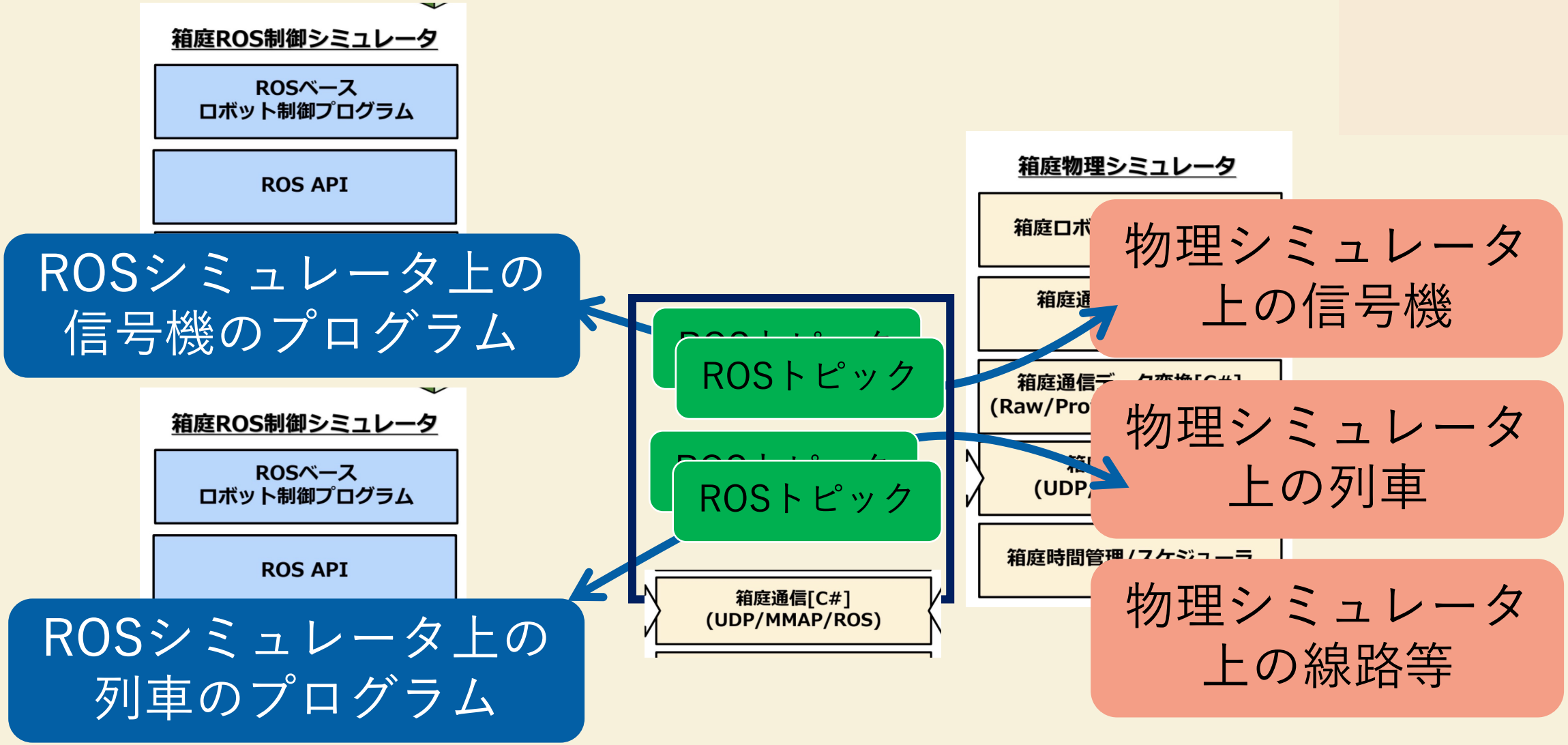
箱庭でやればいい！

  - まずは信号と列車1つずつ…



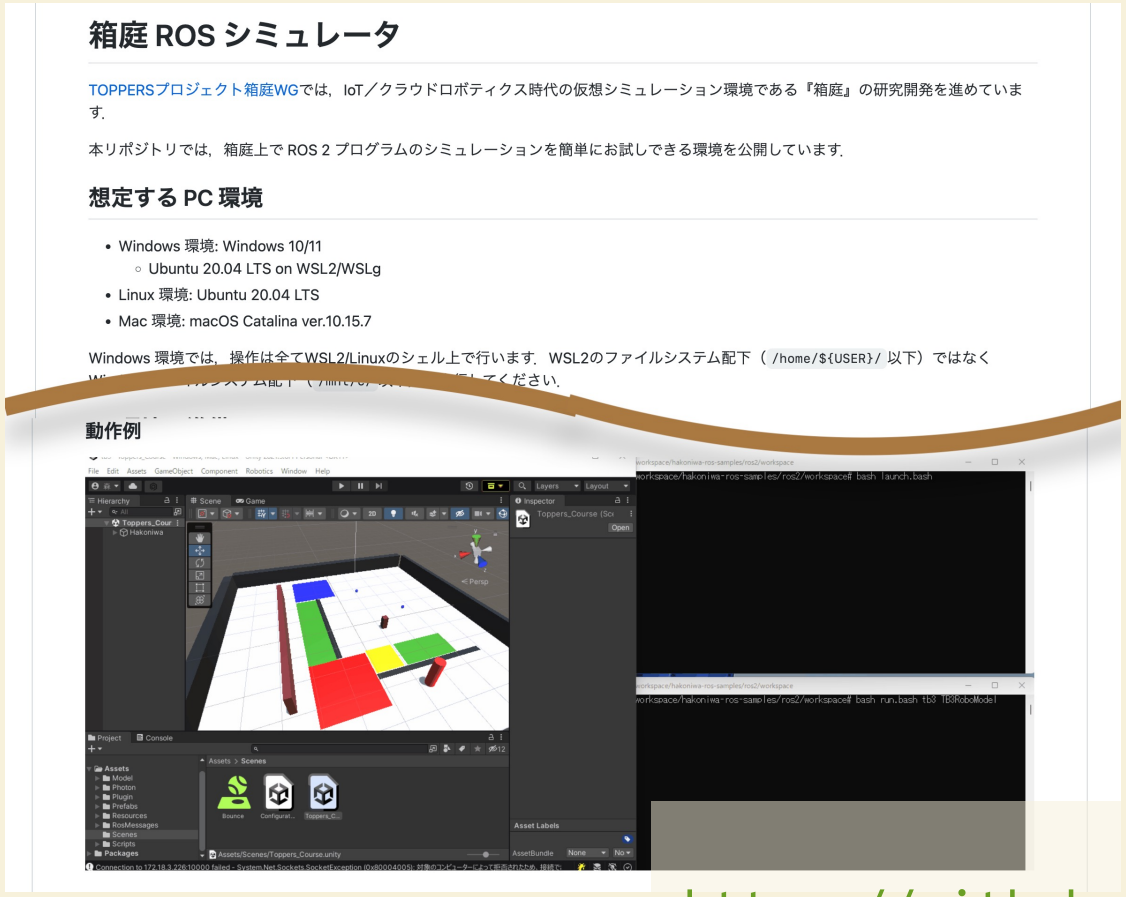
URB (Universal Railway Bus) systemより抜粋
https://arduinorailwaycontrol.com/examples.html#signal-system

箱庭における信号機と列車の構成



導入手順（1）

- ROSシミュレーションの環境を導入する



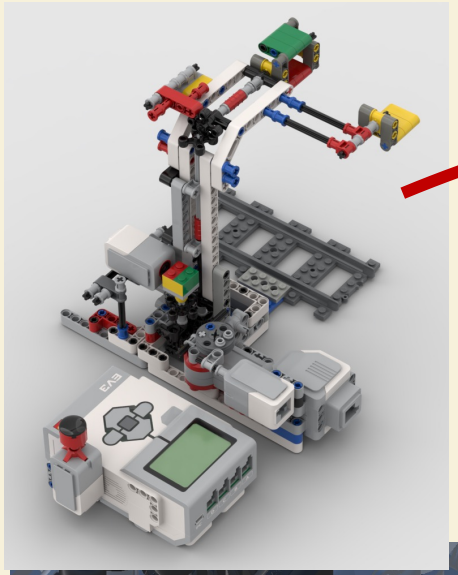
必要なもの

- Windows10～
- GitHub
- WSL2
- Docker
- Unity
- Blender

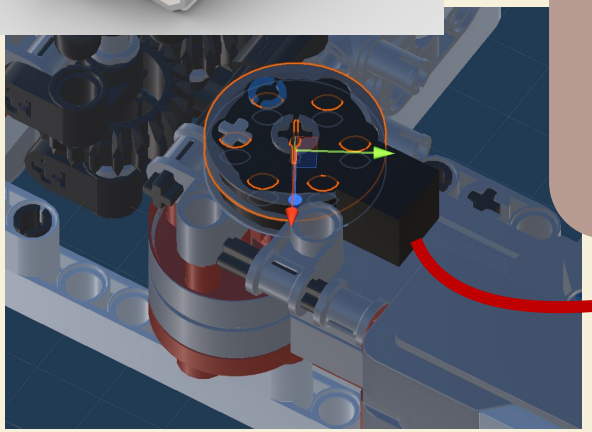
箱庭 ROS シミュレータ
https://github.com/toppers/hakoniwa-ros2sim/

導入手順（2）

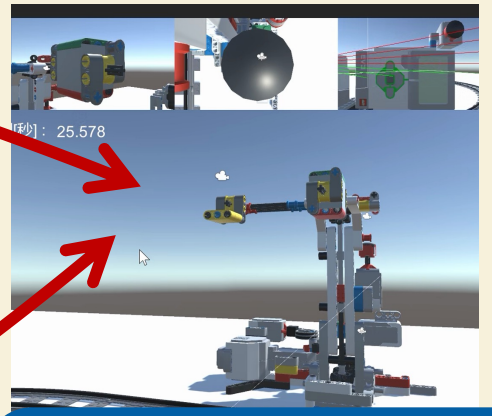
- 物理シミュレータ用の列車や信号機を作る



Unity用のデータ作成
（CAD/Blenderなど）



物理動作のための
オブジェクトの設定
（可動部、センサなど）



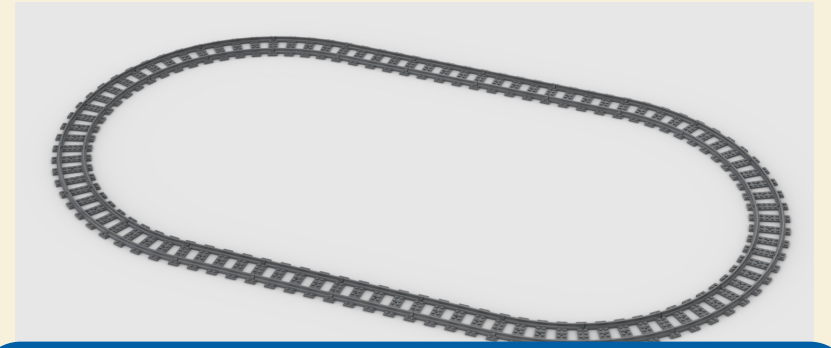
Unity上で動作する
仮想の信号機

あらかじめ作成したプログラムと環境は下記から入手できる

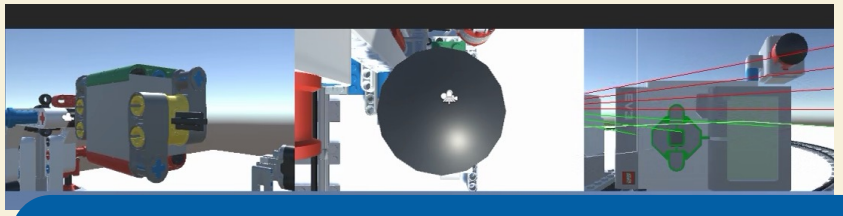
https://bit.ly/3QPzJm6

導入手順（3）

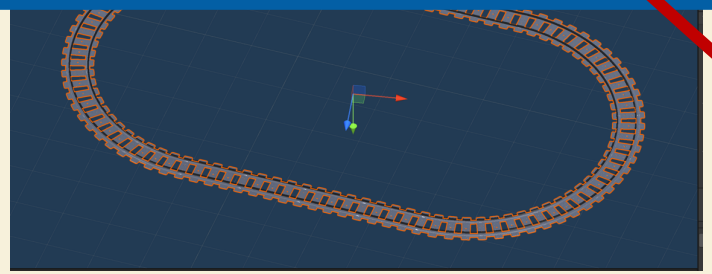
- 物理シミュレータ用の環境を作る

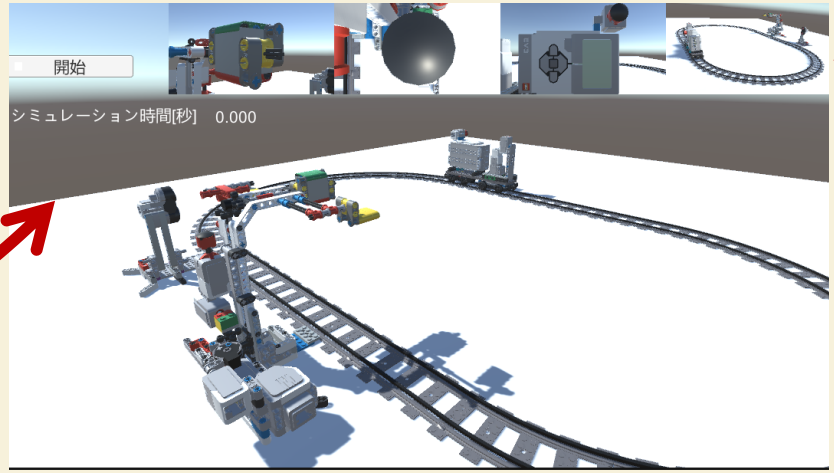


レールの形状、
レールが持つ物理制約



シミュレータ操作のためのUI
（ボタンや表示）、カメラなど

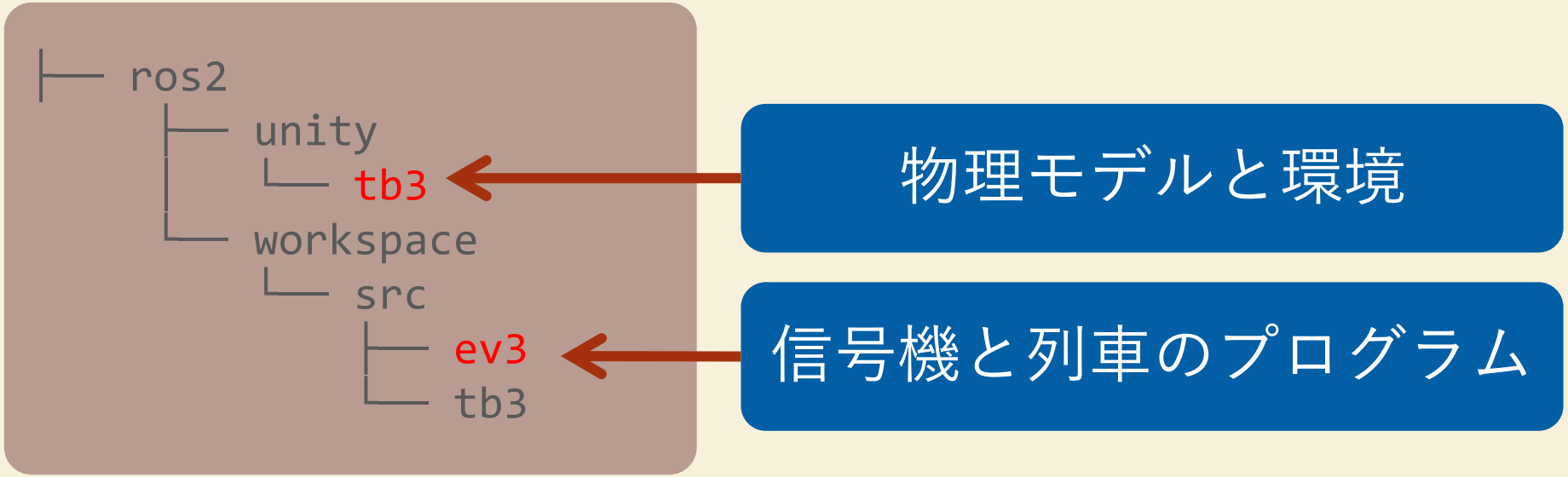




導入手順（4）

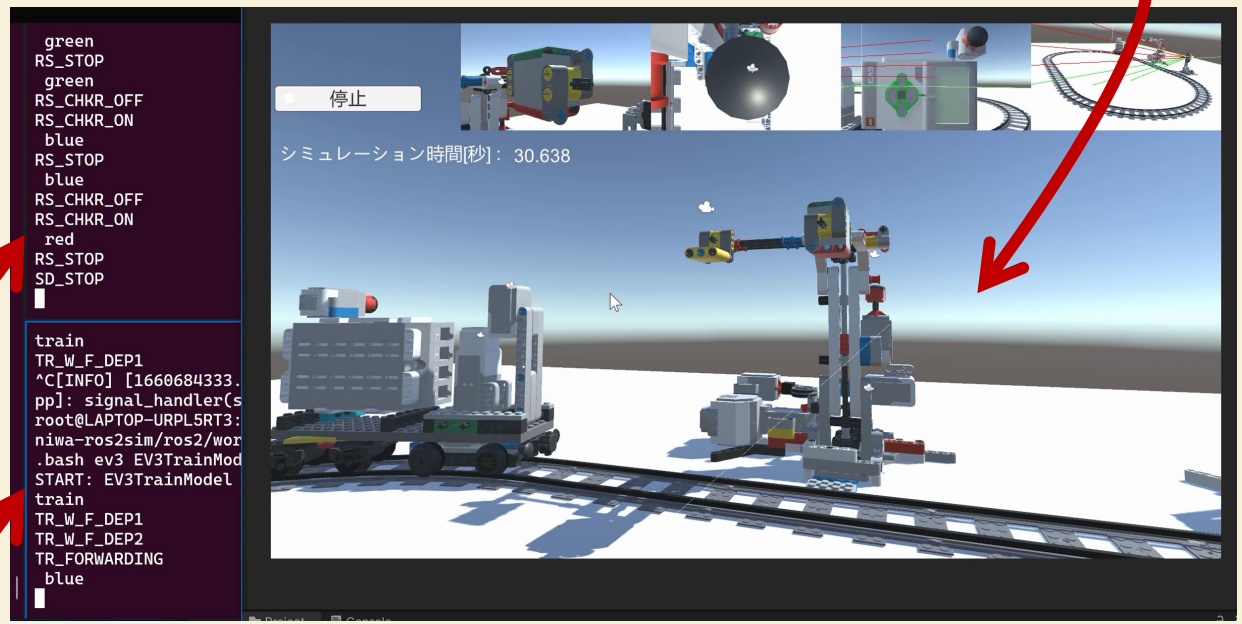
- ロボットと環境をアプリ化する
  - Windows用だと、.exeファイルとデータセットができる
  - アプリ化しないでUnityエディタで実行する方法もある
- プログラムとアプリを配置する

現状、アプリ化は条件付き
Unityエディタ内での実行が主



シミュレーションの実行

- 端末A
  - 開発用dockerコンテナを起動
  - 信号機と列車のプログラムをビルド
  - エンドポイントのプログラムを起動
- 端末B
  - 開発用dockerコンテナにアタッチ
  - 信号機のプログラムを起動
- 端末C
  - 開発用dockerコンテナにアタッチ
  - 列車のプログラムを起動
- Unityアプリorエディタ
  - 物理モデルのシミュレータを起動
  - シミュレーションを開始する



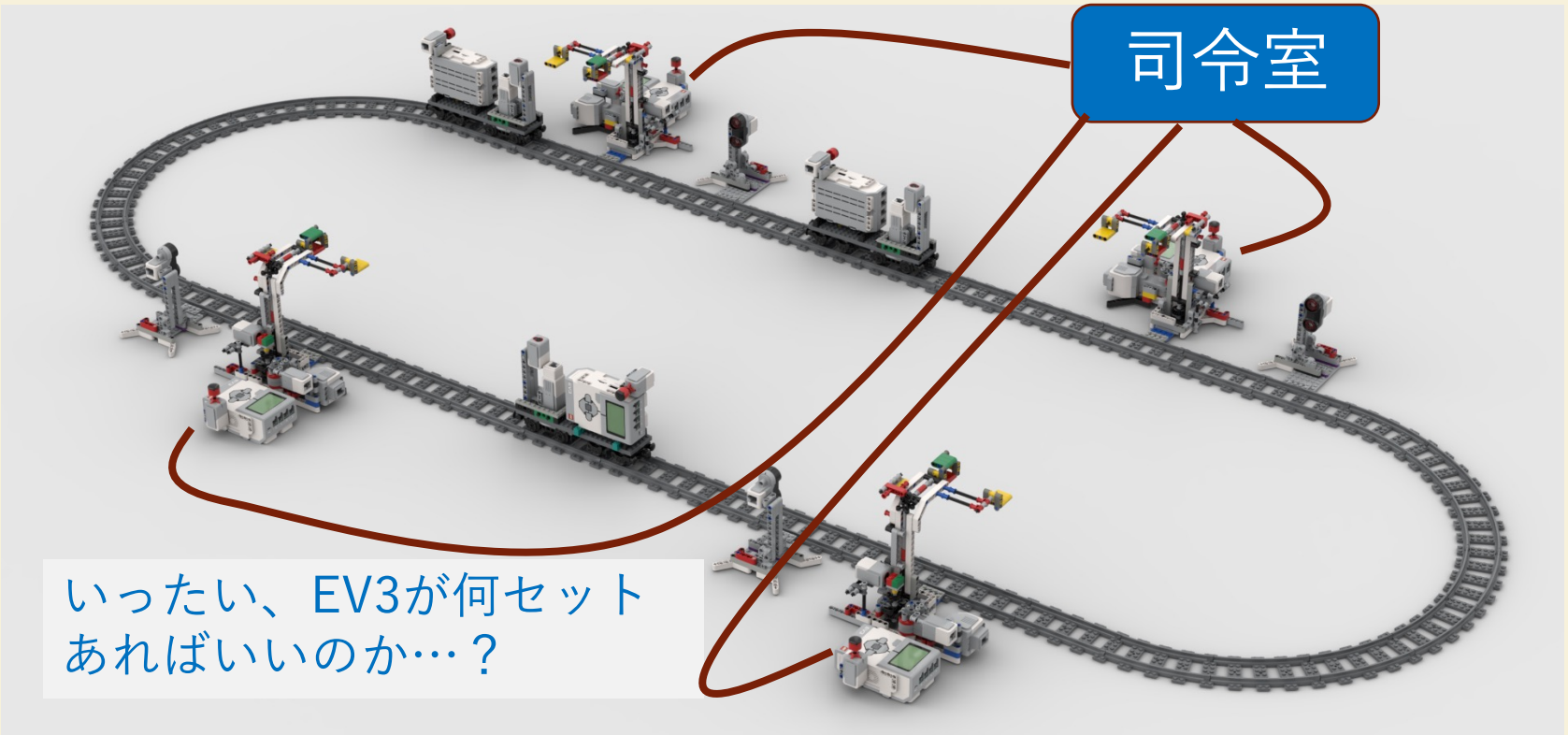
block_signal_demo03.mp4

箱庭によって解消される実機の課題

- 実機はキットの購入が必要
  - 演習するグループ数に応じたキット数が必要
  - 列車や信号機を追加するごとにキットが必要
  - EV3のキットは販売が終了していて、すでに購入が難しい
- 箱庭なら、もっと自由に開発できる
  - 実機や広い場所ながくても開発できる
  - 信号機を増やし、線路を長くできる
  - 電池がなくならない（実機は充電式バッテリーも購入が困難）
- とはいえ、実機が動くのも楽しいんですよね…

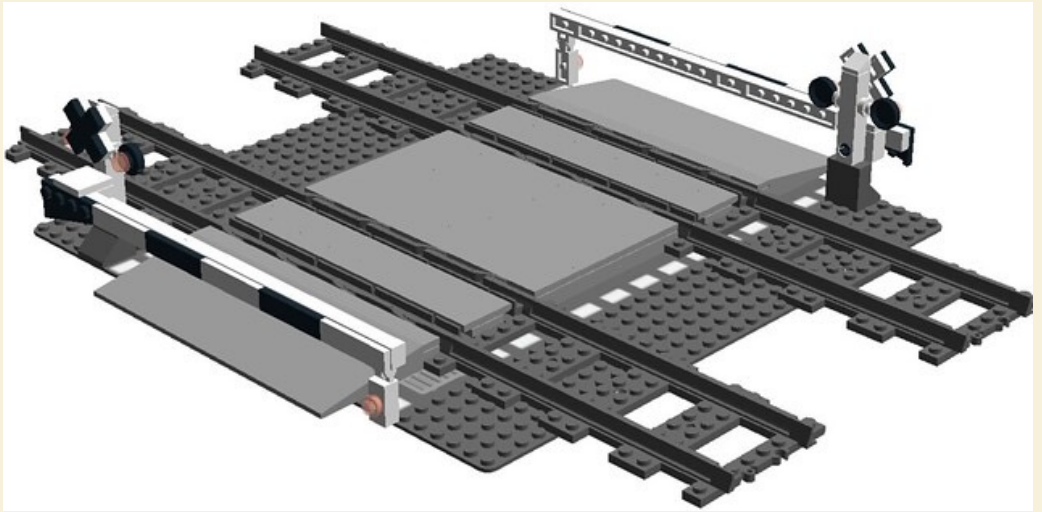
さらに発展させると…

- 複数の信号機で、複数の列車を走らせる
  - 閉塞区間が複数になる
  - 列車は信号に従うことでうまく走行できる
- 司令室の設置
  - どの閉塞区間に列車がいるか把握する
  - 通信で信号をまとめて切り替える



さらに発展させると…

- 待避線とポイントを追加する
  - 衝突しないよう複数の列車を双方向に走らせる
- 踏切を追加する
  - 列車の通過に合わせて、遮断器を動かす



踏切のモデル（アメリカの郊外のもの）
https://farm6.staticflickr.com/5523/14036859189_30285d3944_z.jpg

まとめ

- 箱庭を使ったシミュレーションを実行した
  - 自動搬送システム
    - マイコン制御シミュレータ＋物理シミュレータ
  - 信号機に従って走行する列車
    - 物理シミュレータ＋ROS制御シミュレータ
- 実機も欠かせないが、箱庭は開発を大いに助ける
- 実は…ソフト屋には物理モデルの作成がネック

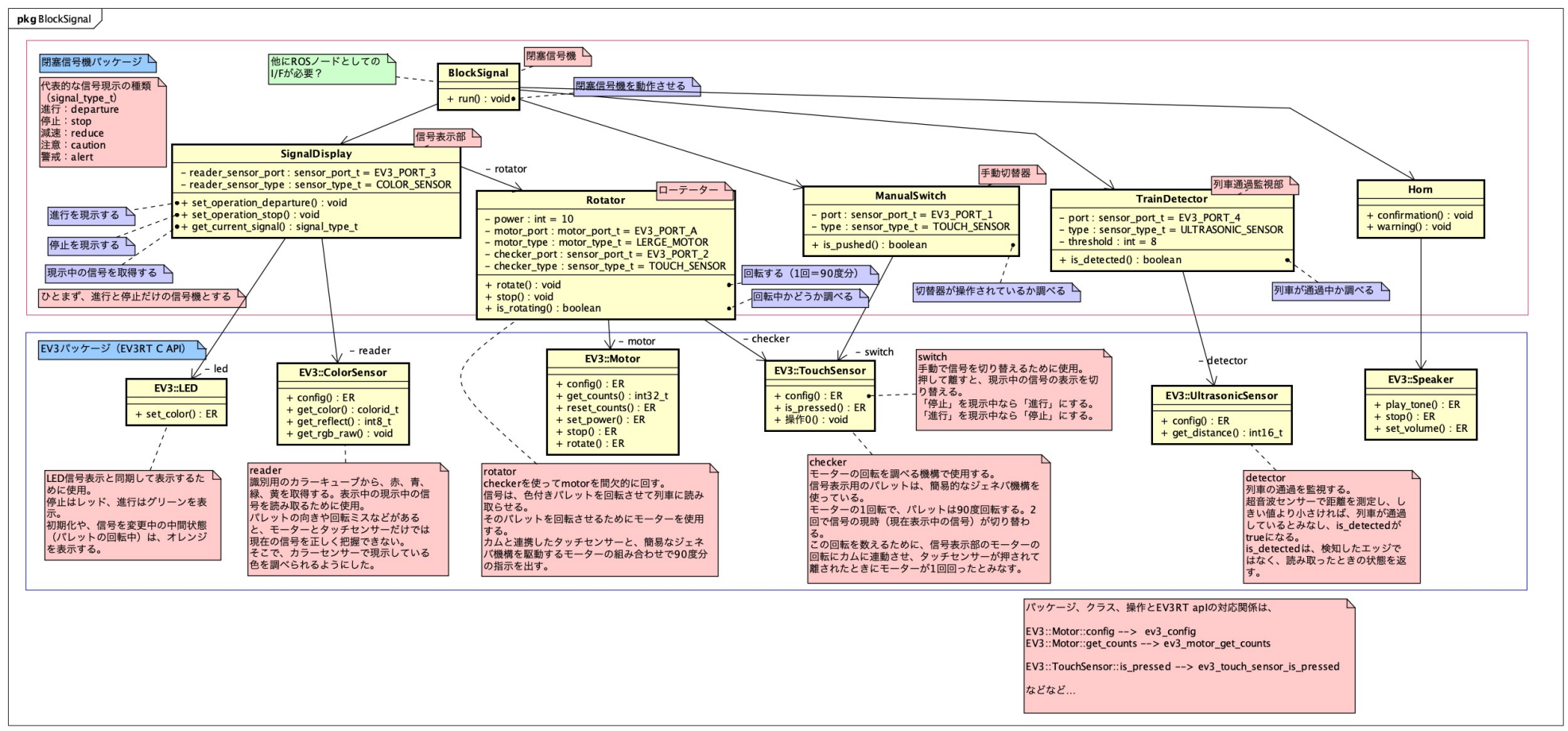
【おまけ】

作成に使ったモデル図、信号機や列車の詳細

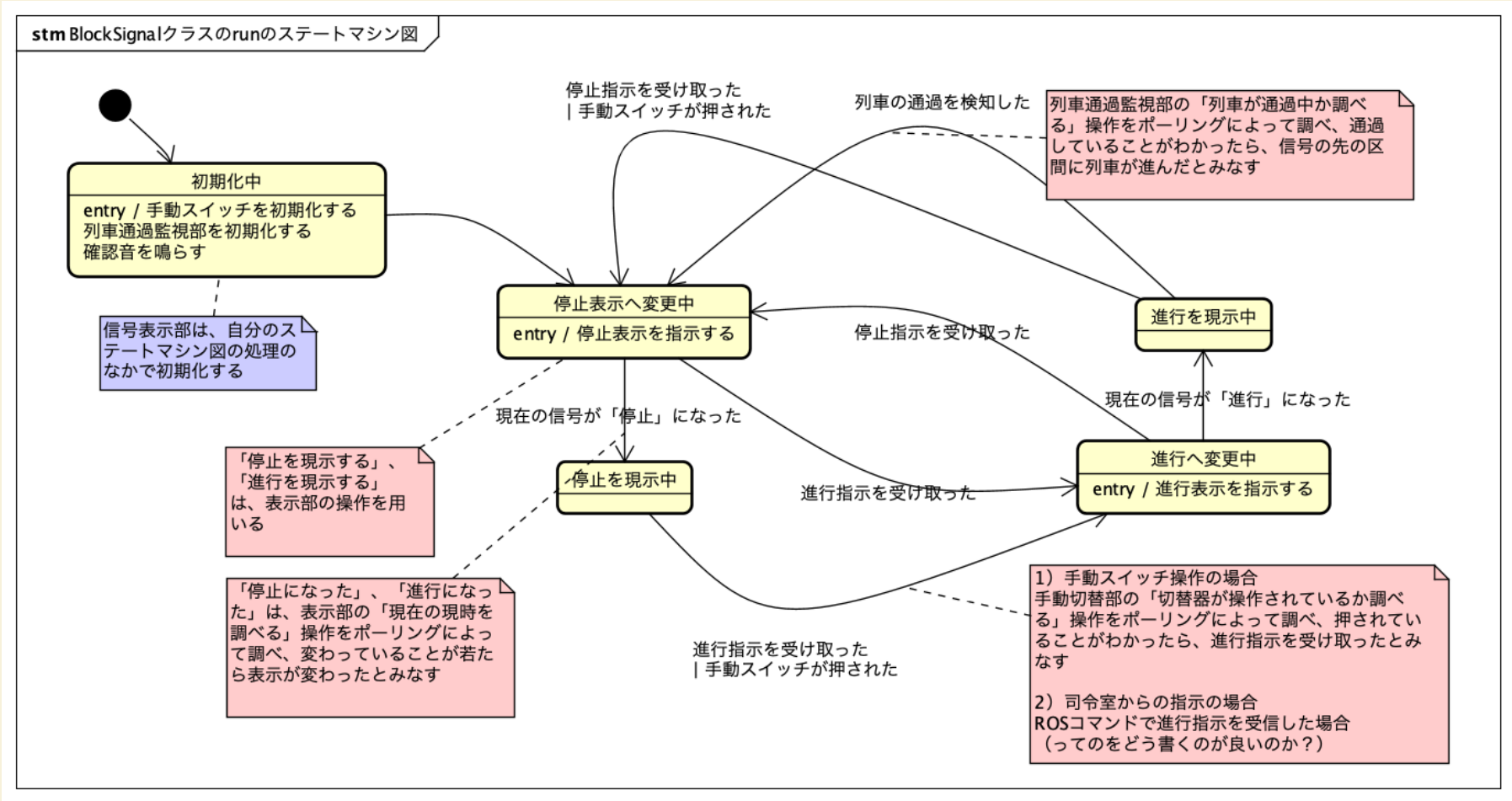
実機用のモデル図やCADデータは、GitHubから入手可能

https://github.com/kuboaki/ev3_train/

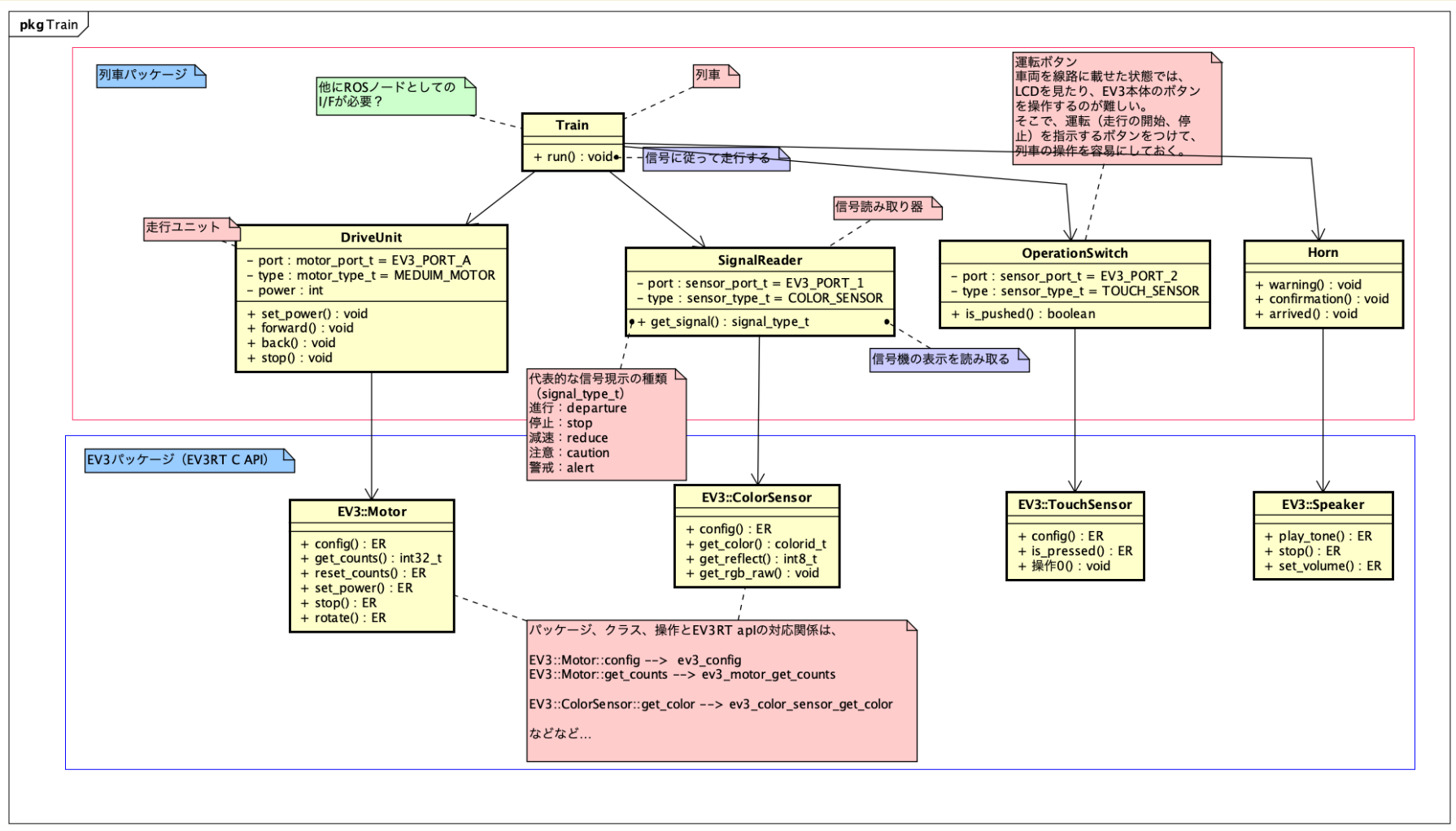
UMLで作成した信号機のクラス図



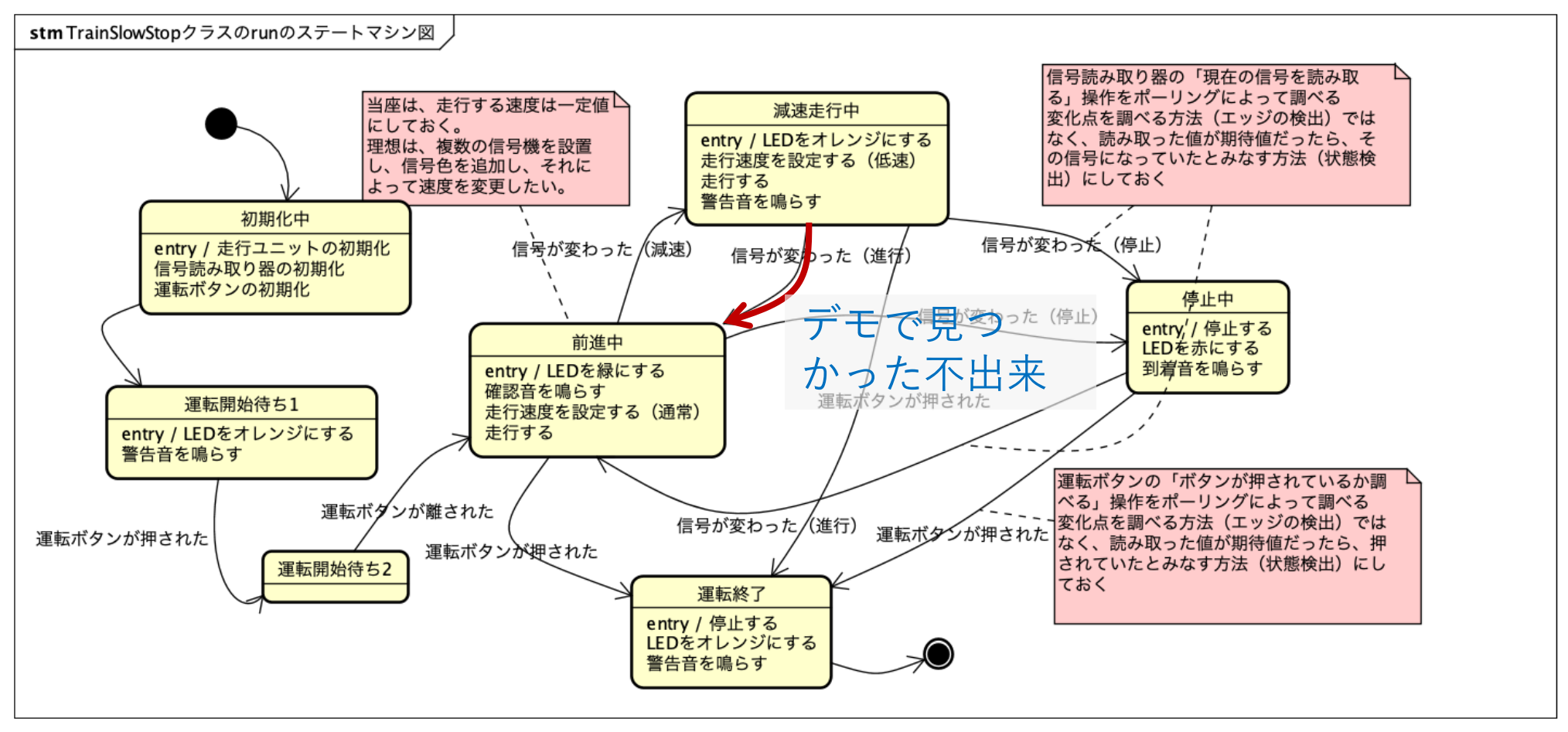
UMLで作成した信号機のステートマシン図



UMLで作成した列車のクラス図

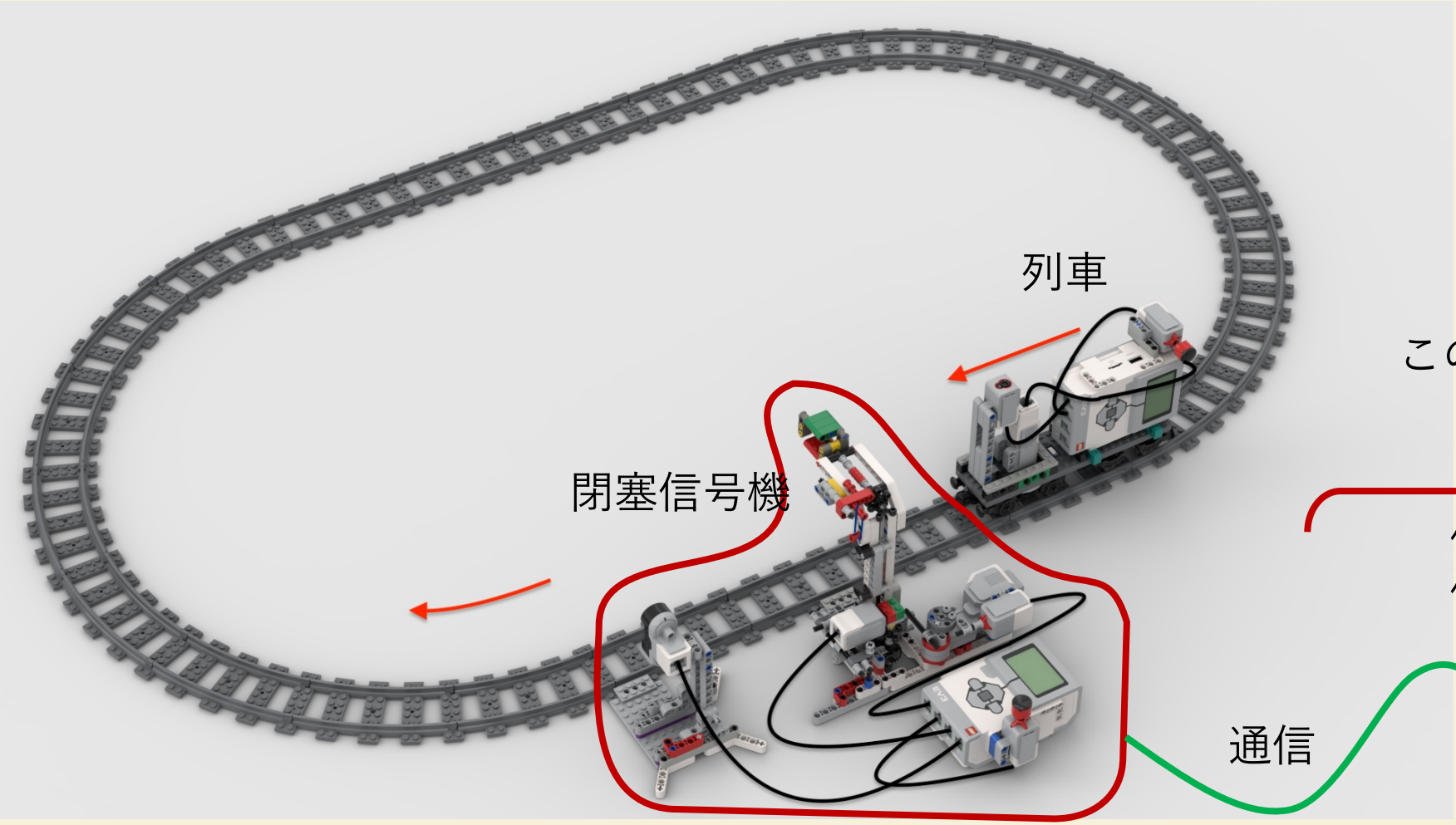


UMLで作成した列車のステートマシン図



システムの全体像

- エンドレスの線路、閉塞信号機、列車



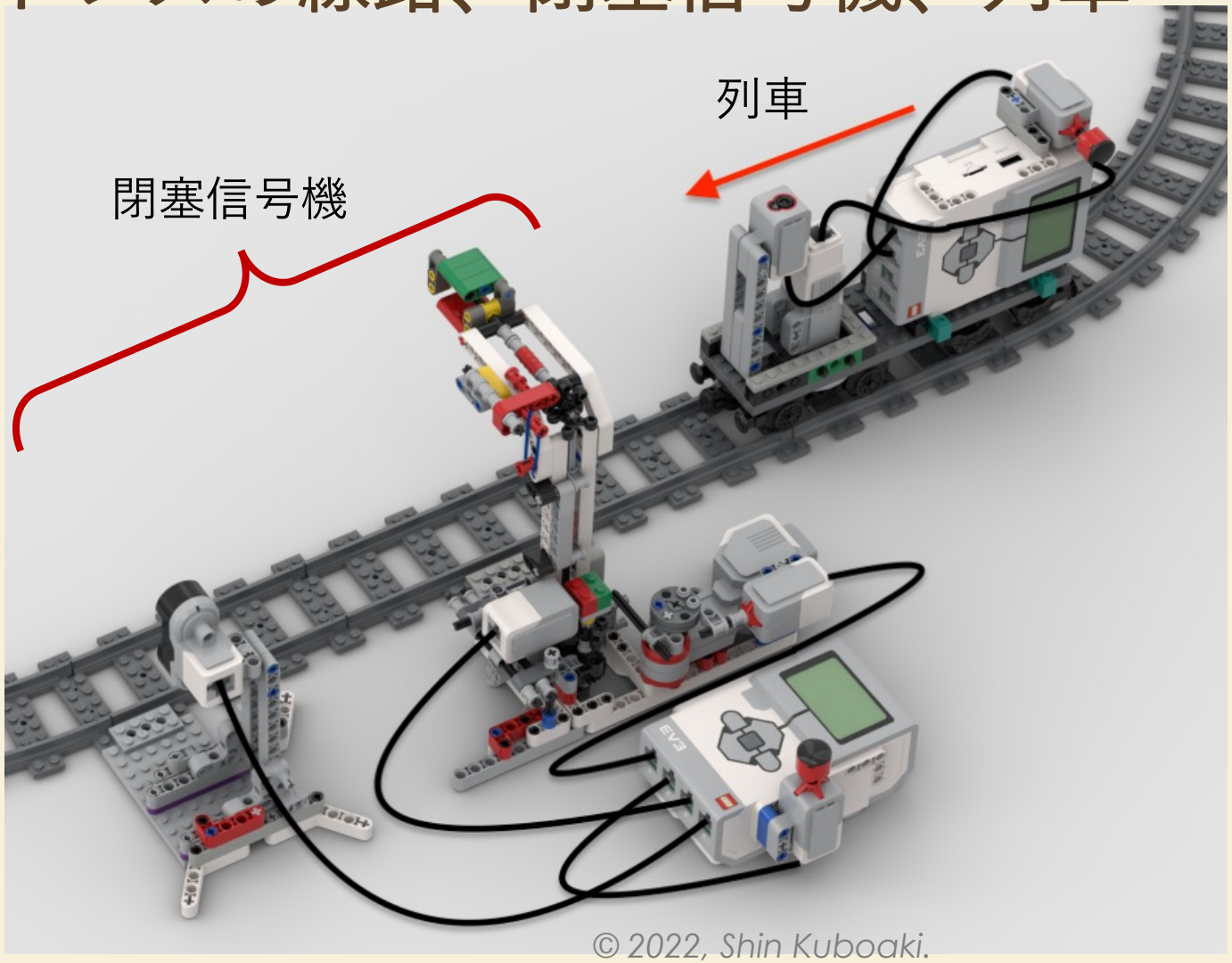
このような拡張も
考えられる

信号機への指示、
信号のモニター

司令室
（PC）

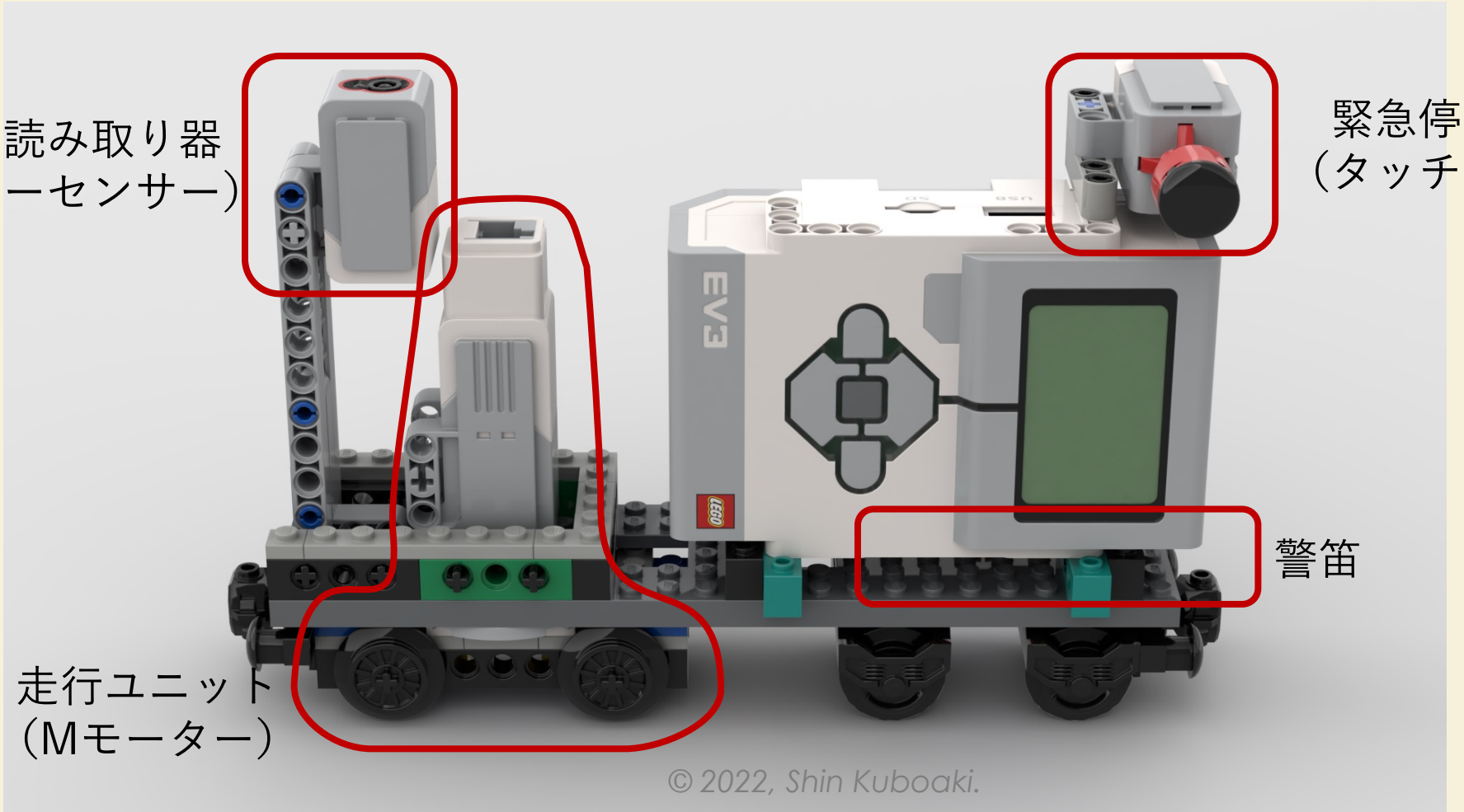
システムの全体像（拡大）

- エンドレスの線路、閉塞信号機、列車



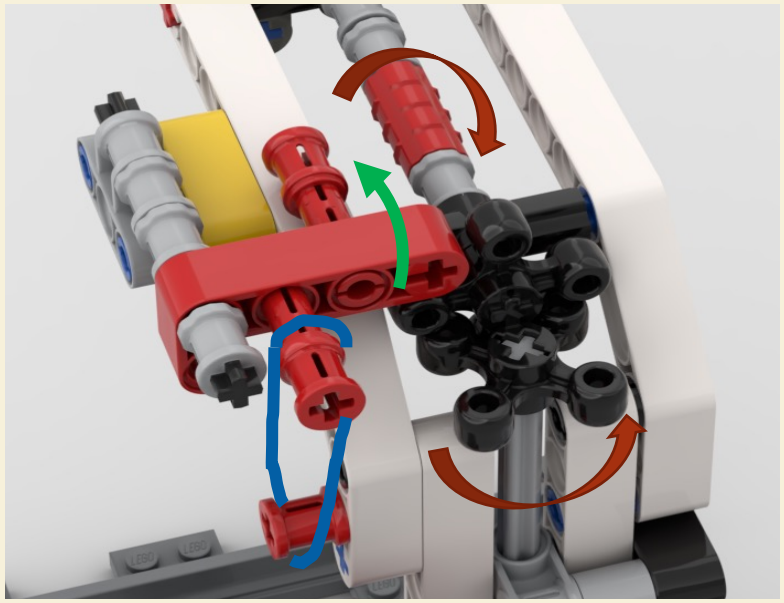
列車の構造

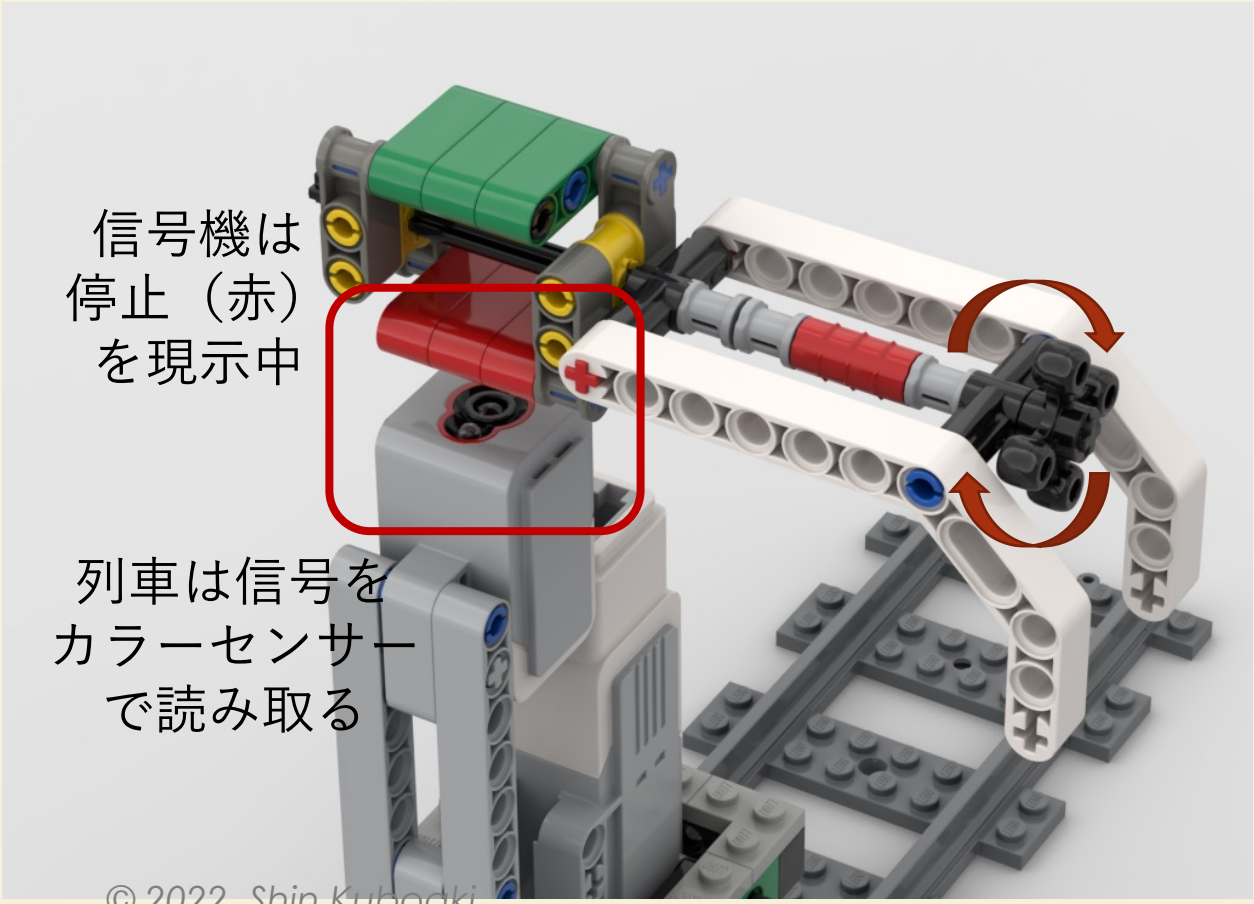
- 前進、後退する、信号読み取り、警笛を鳴らす



信号機の構造（1）

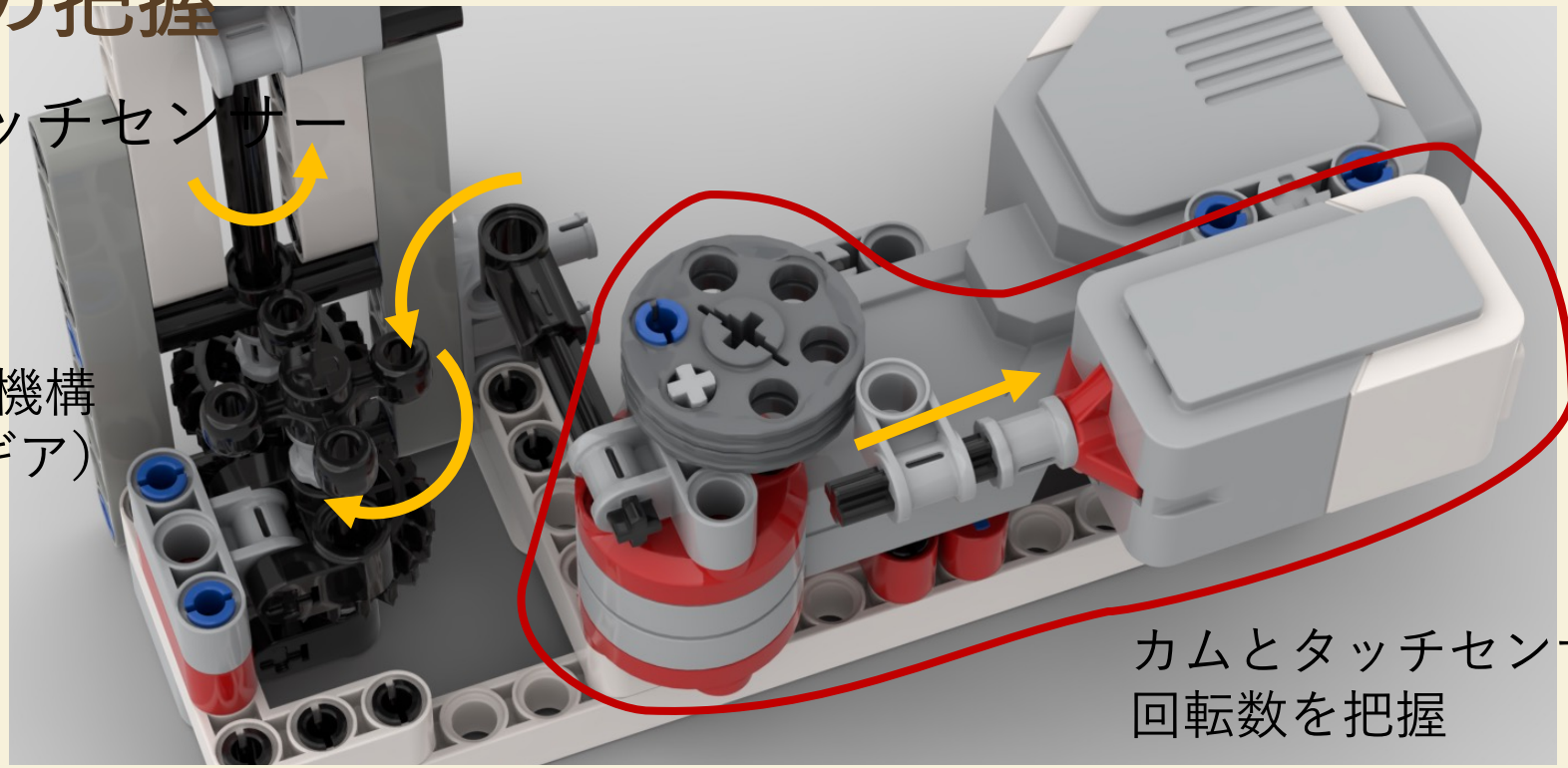
- 下向きパレットで進行・停止を示す2灯式
  - 列車は、カラーセンサーで信号を読み取る
- 間欠時の回転抑止
  - 十字ギアとラチェット





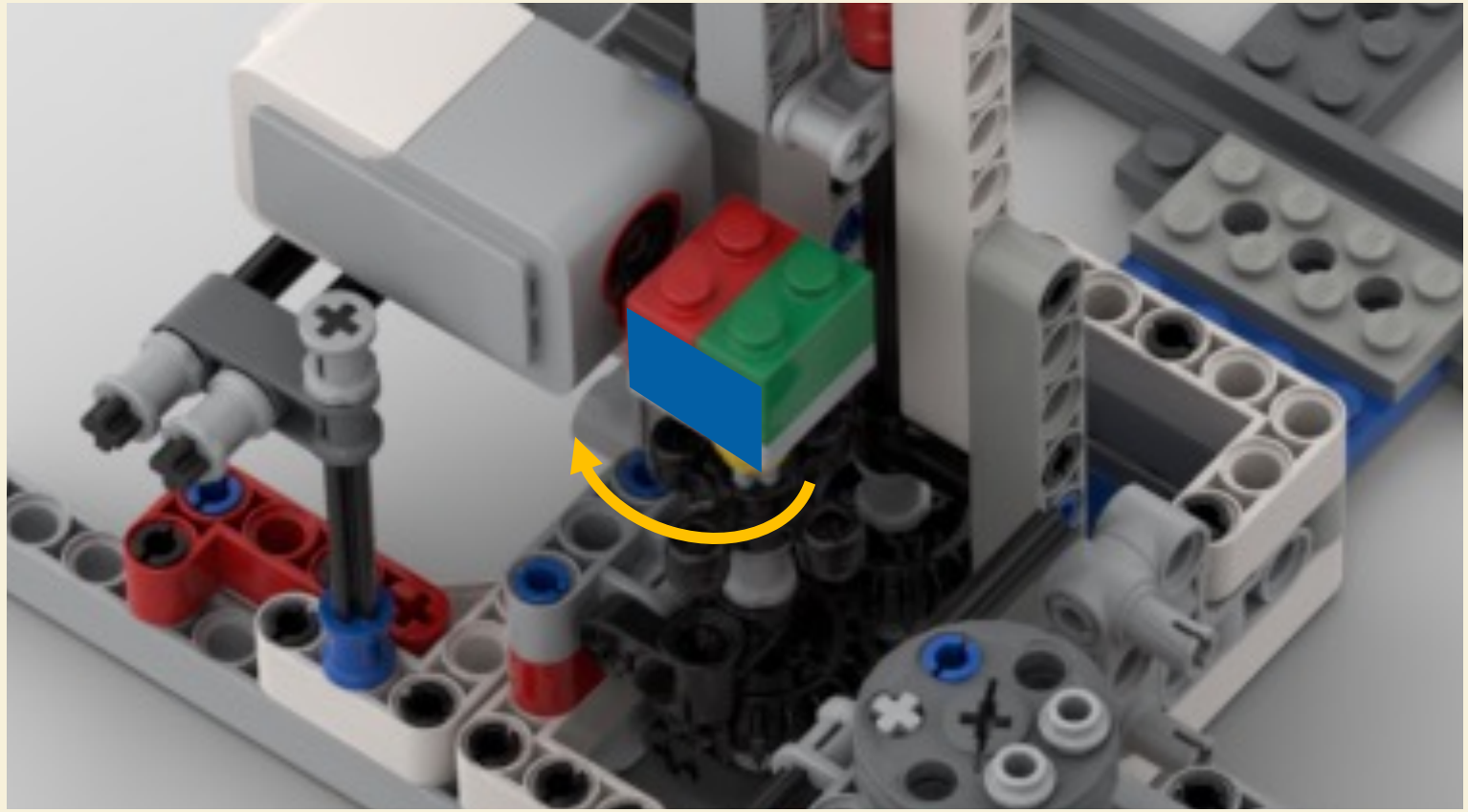
信号機の構造（2）

- パレットをモーターで間欠回転する機構
  - 簡素なジェネバ機構で間欠回転を実現
- 回転した数の把握
  - カム機構とタッチセンサー



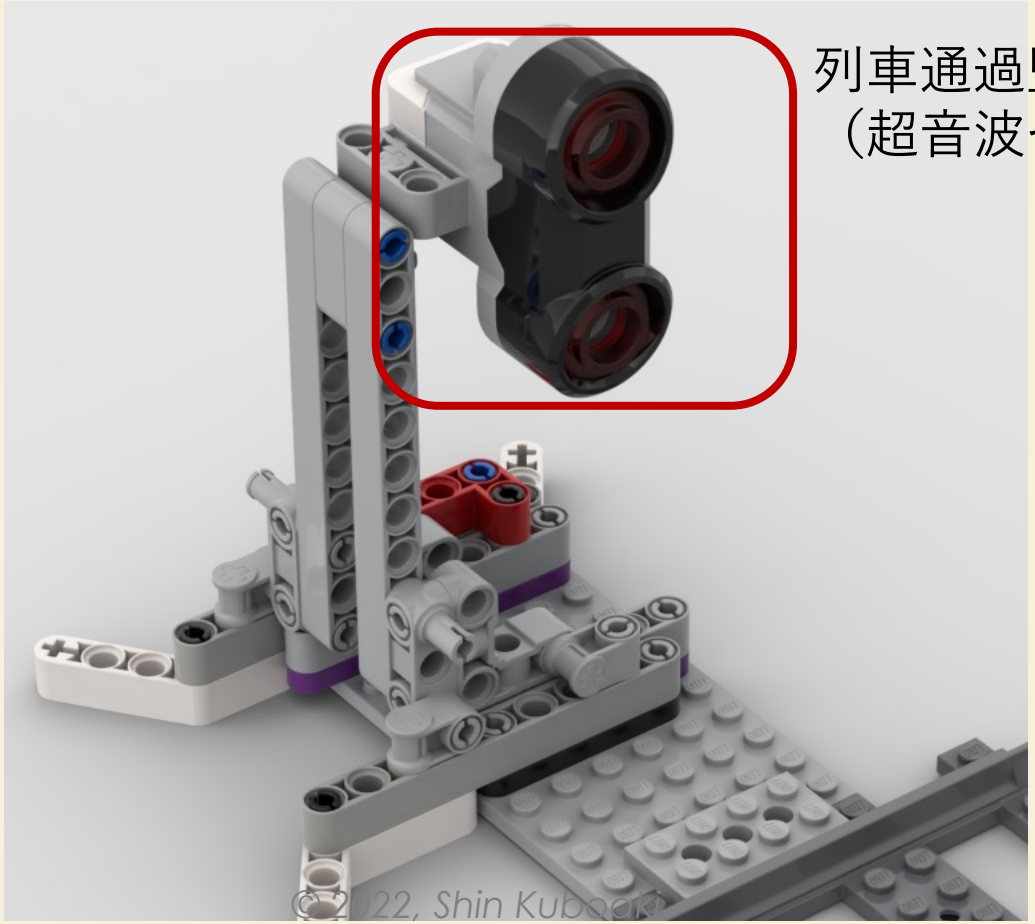
信号機の構造（3）

- 現示中の信号を調べる機構
  - 信号の回転に合わせてカラーセンサーで読み取る



列車通過監視部の構造

- 超音波センサーを使って通過を調べる



列車通過監視部
（超音波センサー）

9/2(金) 10:20～11:30 セッションs3b

ここから始める箱庭の活用

― おしまい ―



久保秋 真（チェンジビジョン）
細合 晋太郎（東京大学）
高瀬 英希（東京大学）