モデル駆動開発
～実践編～

目次

- はじめに（10分）
- LEDSliderを動かす（1時間）
  - LEDSliderとは（5分）
    - 要求
    - 状態を考える
  - モデル図の作成（15分）
  - m2tプラグインの設定変更（5分）
  - Raspberry piへ転送(5分)
  - 動作確認（20分）
- 開発は続く

3 はじめに

ライブラリ・テンプレートのダウンロード

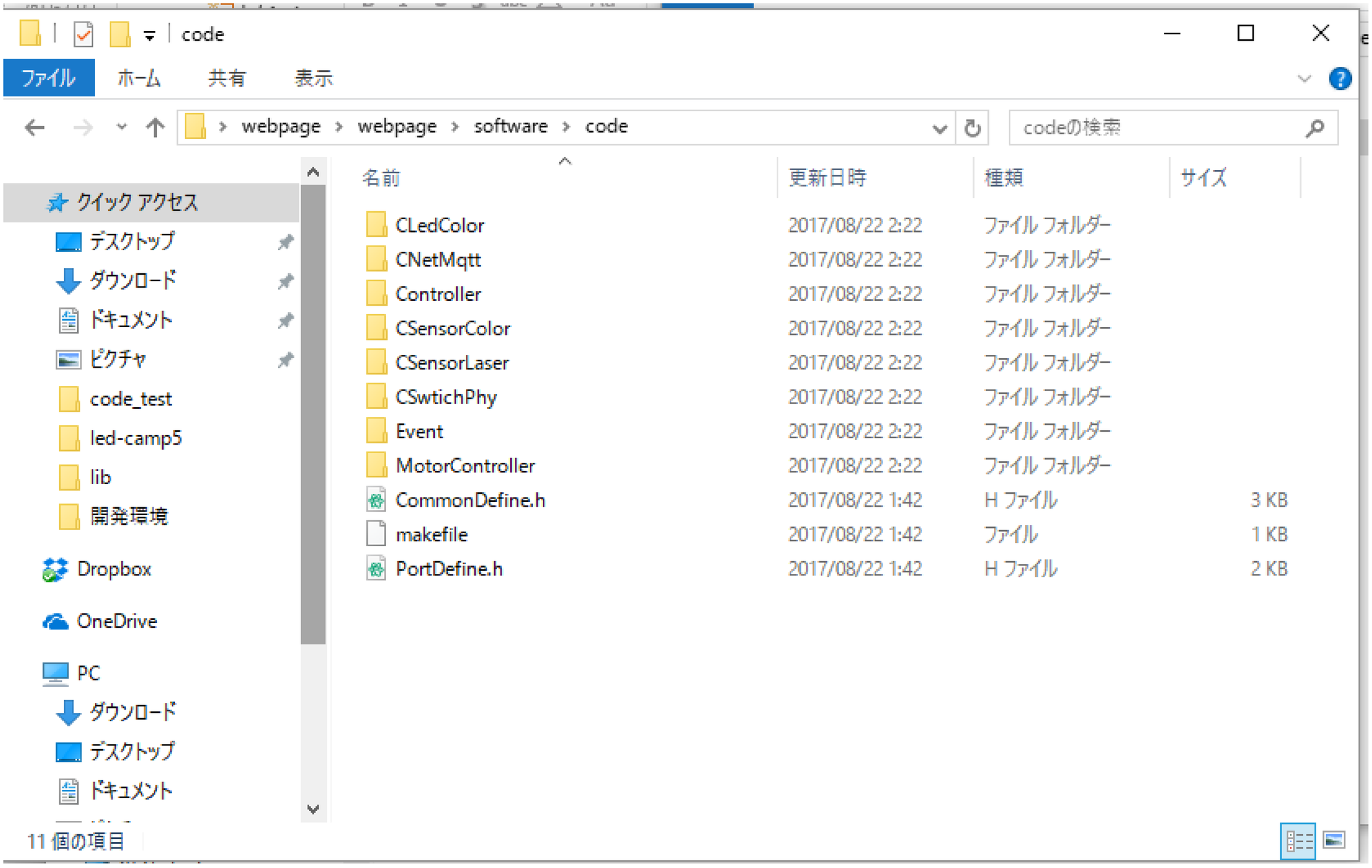
http://192.168.20.200へアクセスする

ソフトウェア類

- astah* Professionalプラグイン
  - モデル駆動開発プラグイン（.jarファイル）
  - コンソール出力プラグイン（.jarファイル）
- ライブラリ
- template

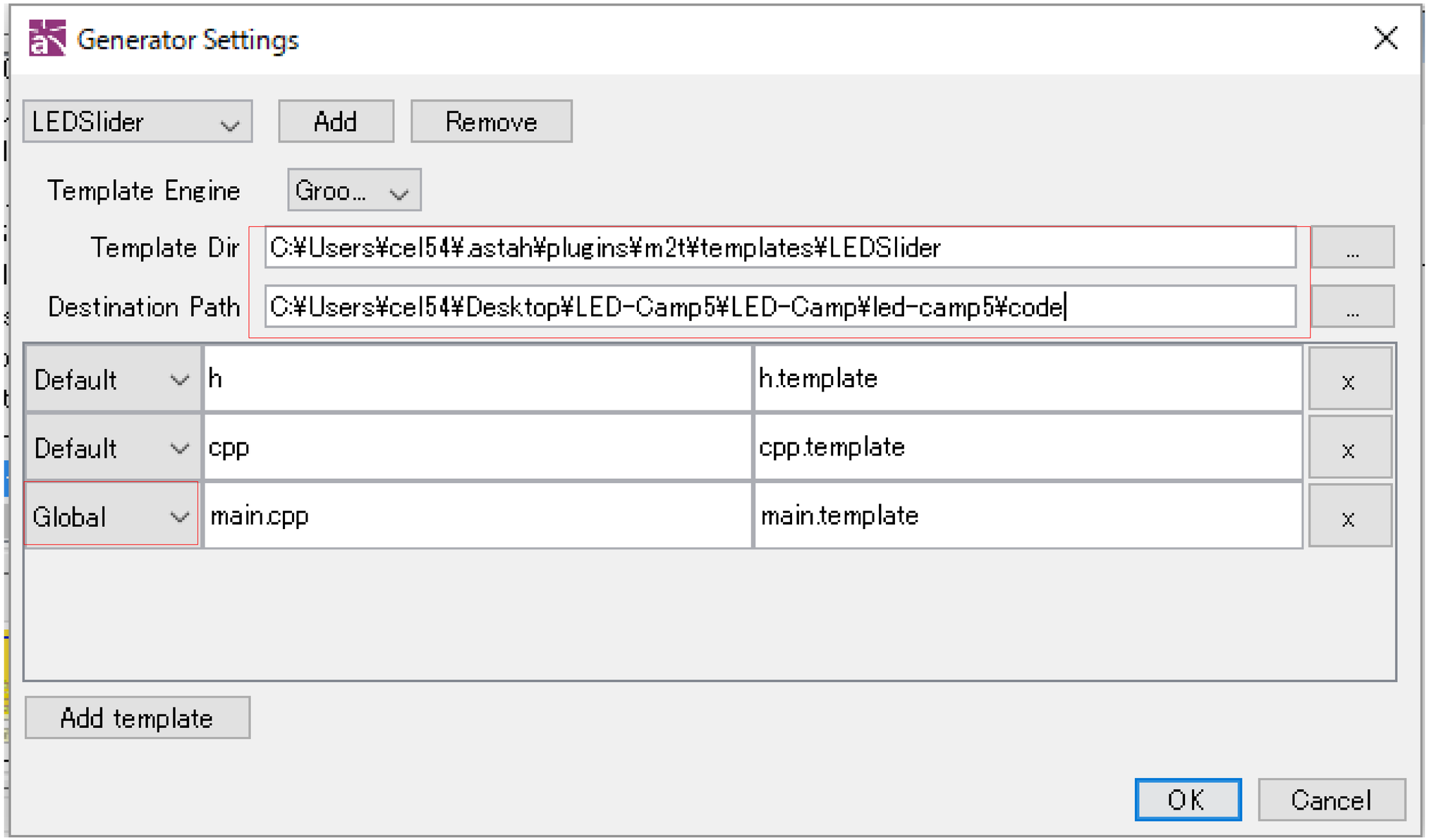
ライブラリの展開

「code.zip」を、任意のフォルダに展開する。



テンプレート設定

「template.zip」を展開して、
C:¥Users¥◯◯◯◯¥.astah¥plugins¥m2t¥templates
の中に「LEDSlider」フォルダごとコピーする。



7 LEDSLIDERを動かす

LEDSliderとは

事前実習では、PreLEDSliderで動作を確認してもらいました。

合宿も本番を迎え、いま目の前には、本物のLEDSliderがあります。

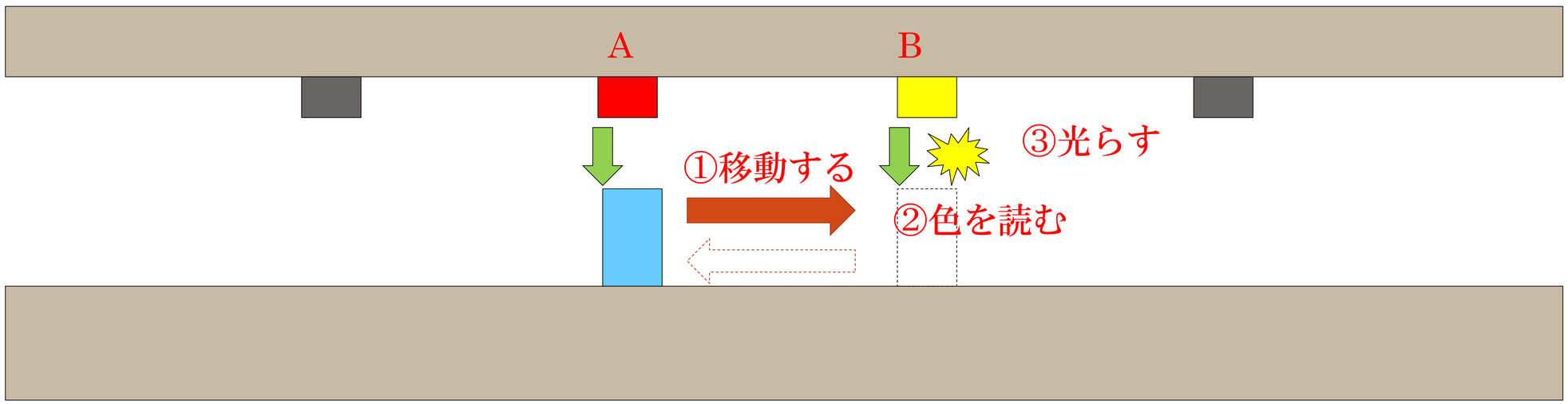
この「モデル駆動開発～実践編～」では、実物のLEDSliderを用いてモデル駆動開発を実践していきます。

要求

LEDSliderを使って、工場の検品作業をすることになりました。
A地点とB地点に交互に部品がやってきます。
今回、LEDSliderはこの交互にやってくる部品の色を読み、それと同じ色をLEDで点灯して、正しい部品が流れてきたかをチェックします。

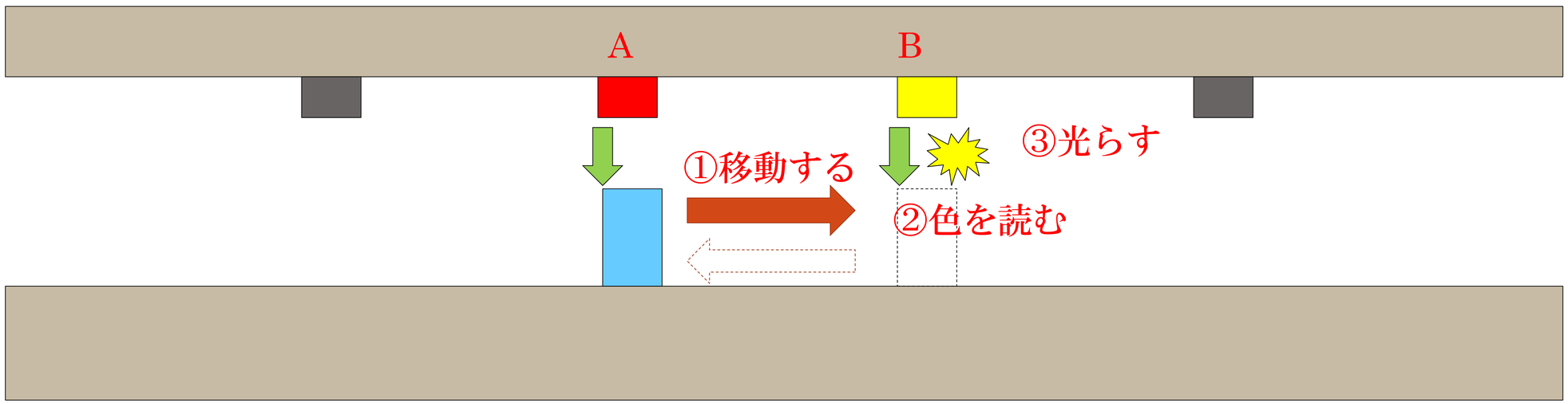
今回の作業をするにあたり、LEDSliderにはカラーセンサーと、LEDを取り付けてあります。
（実はその他にも、レーザーセンサーとかもついてたりしますが、また今度・・・）

今回の実習では、この要求を満たすようなLEDSliderのアプリケーションを開発しましょう。



状態を考える-1

モデルを作る前に、要求を分析してみましょう。
PreLEDSliderでは、クラス図と、ステートマシン図を用いてモデル化しました。
クラス図は構造を、ステートマシン図は振舞いをモデル化したものです。
今回、構造としてはPreLEDSliderと大きくは変わりません。しかし、振舞いは大きく変わります。

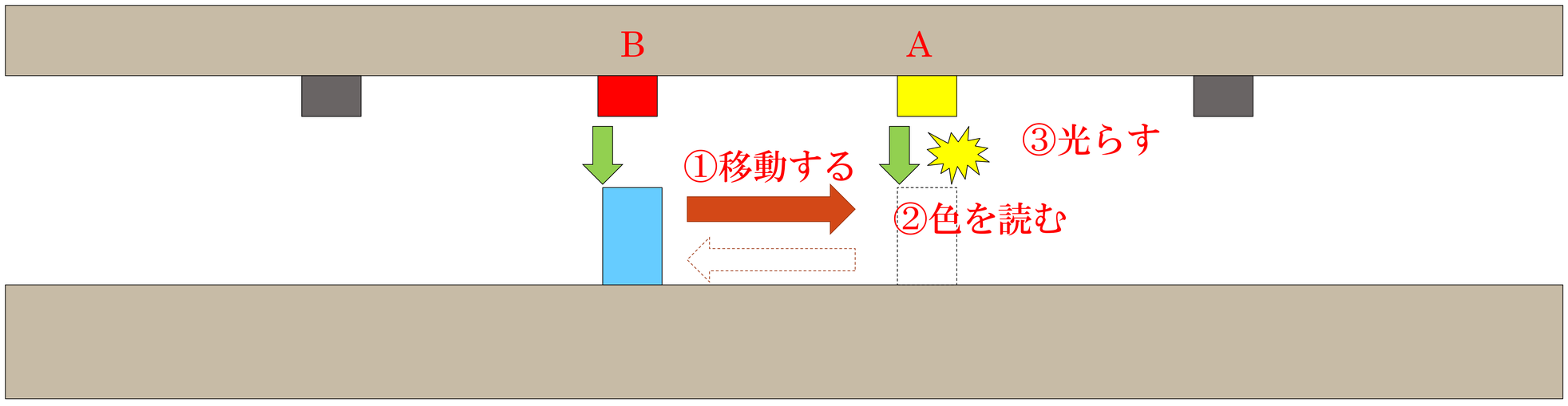


状態を考える-2

今回の要求から下記の状態があると考えました

1. A→Bへ移動する
2. 検品する
3. B→Aへ移動する

今回はこの3状態と+初期設定を行う状態の4状態を作りたいと思います。
※これは一例です。どのような状態が良いと考えるかは十人十色、いろいろな考え方があることでしょう。今後行う競技会のアプリケーション作成では、チーム内でどのような状態とすればもっともよいアプリケーションとなるか議論をしてください



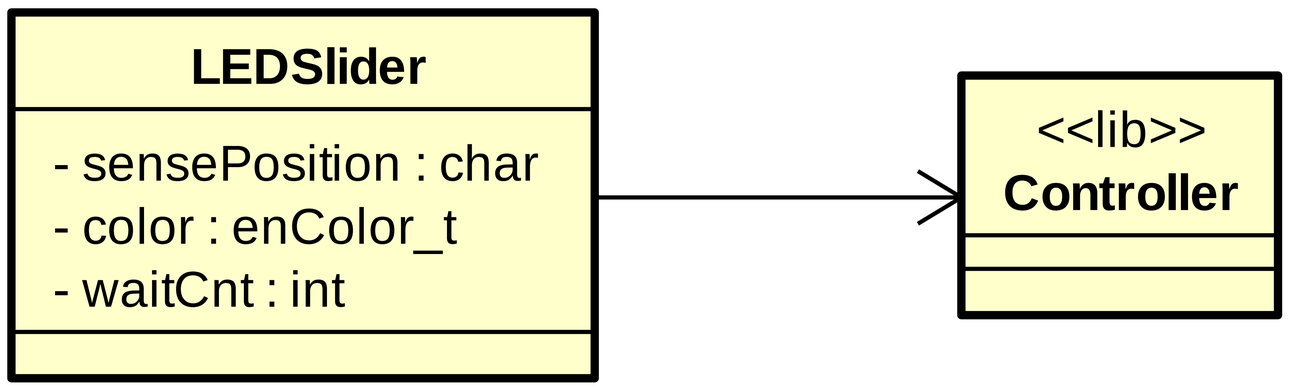
モデル図の作成

まずはクラス図を作成してみましょう。

事前実習では「PreLEDSlider」と「PreController」を使いましたが、今回はPreではないため、「LEDSlider」と「Controller」とします。

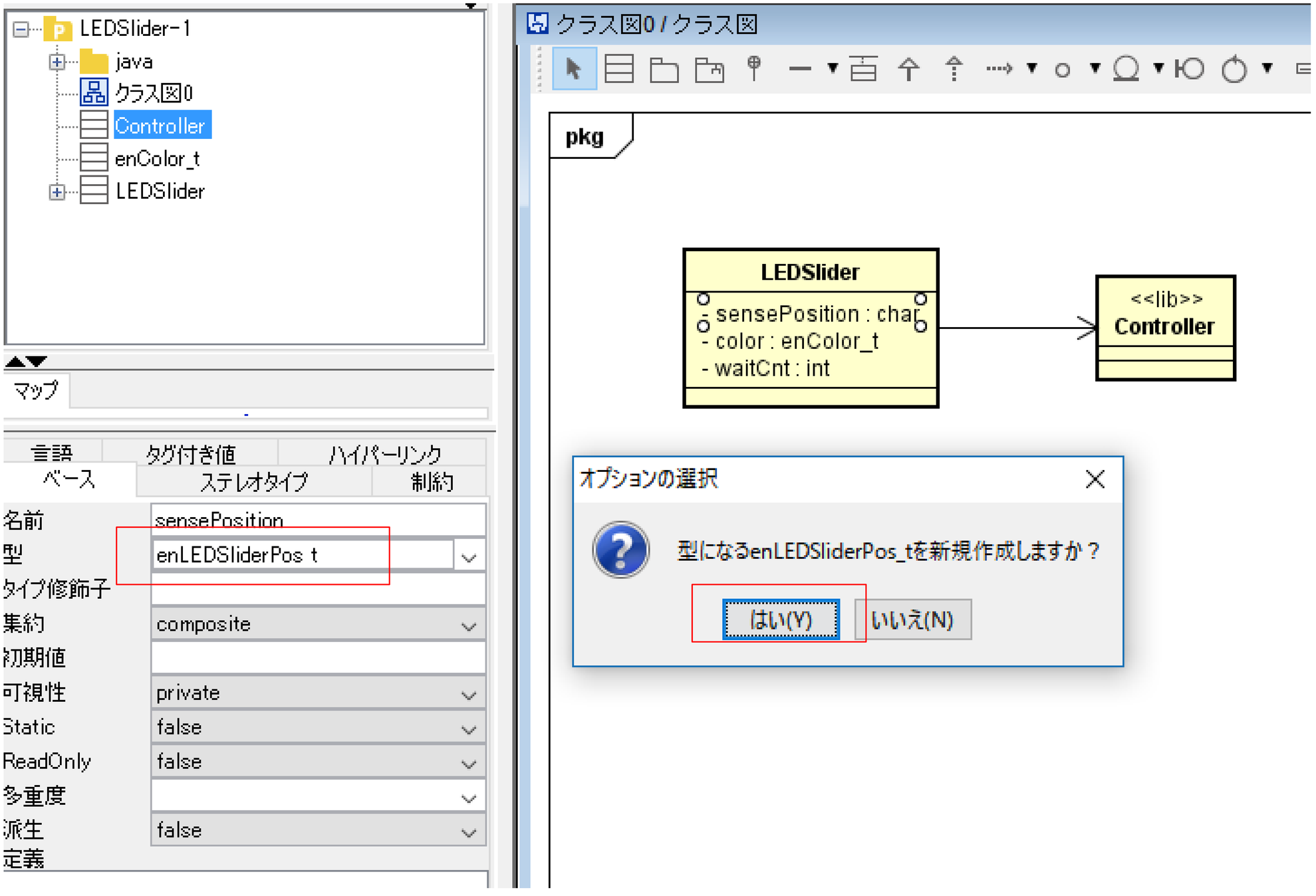
LEDSliderは下記2つの変数を持ちます。

- sensePosition：検品時にどの場所にいるかを文字で記憶します
- color：enColor_t型（enum型で定義）で、カラーセンサの読み取り値を記憶します。
- waitCnt：検品時に待つ時間をカウントします



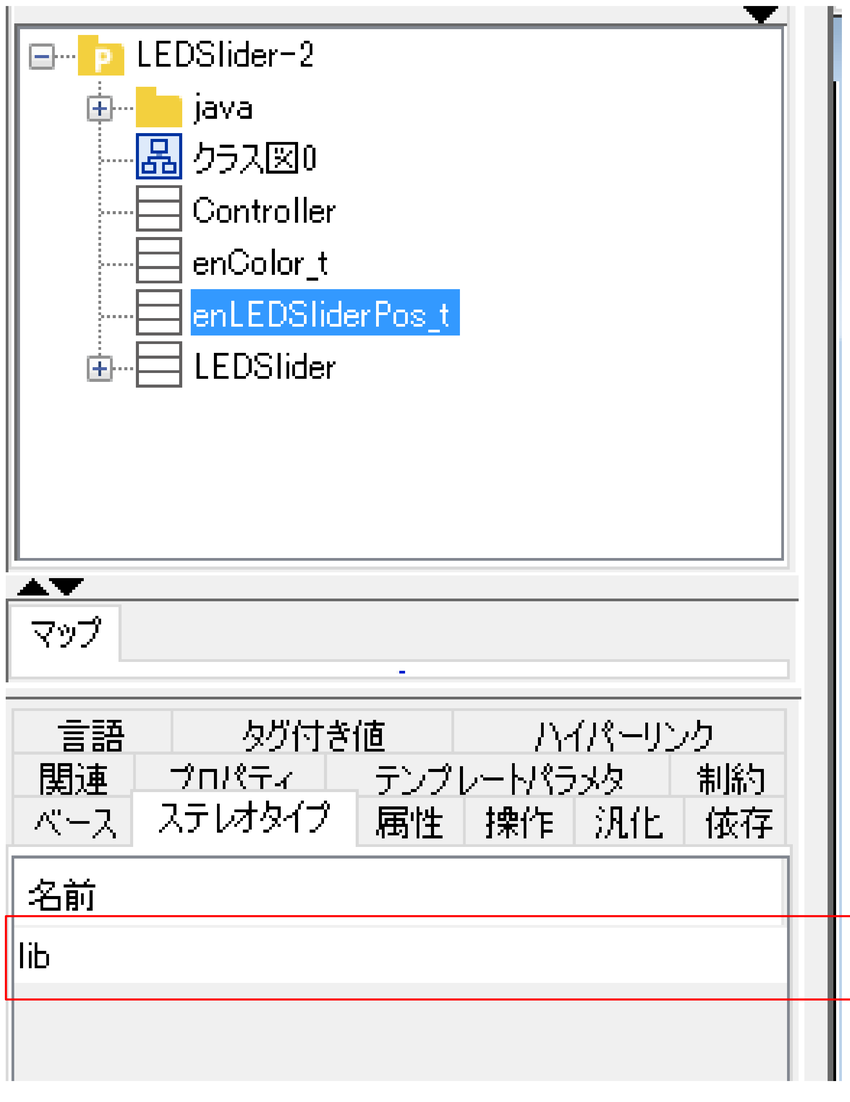
モデル図の作成

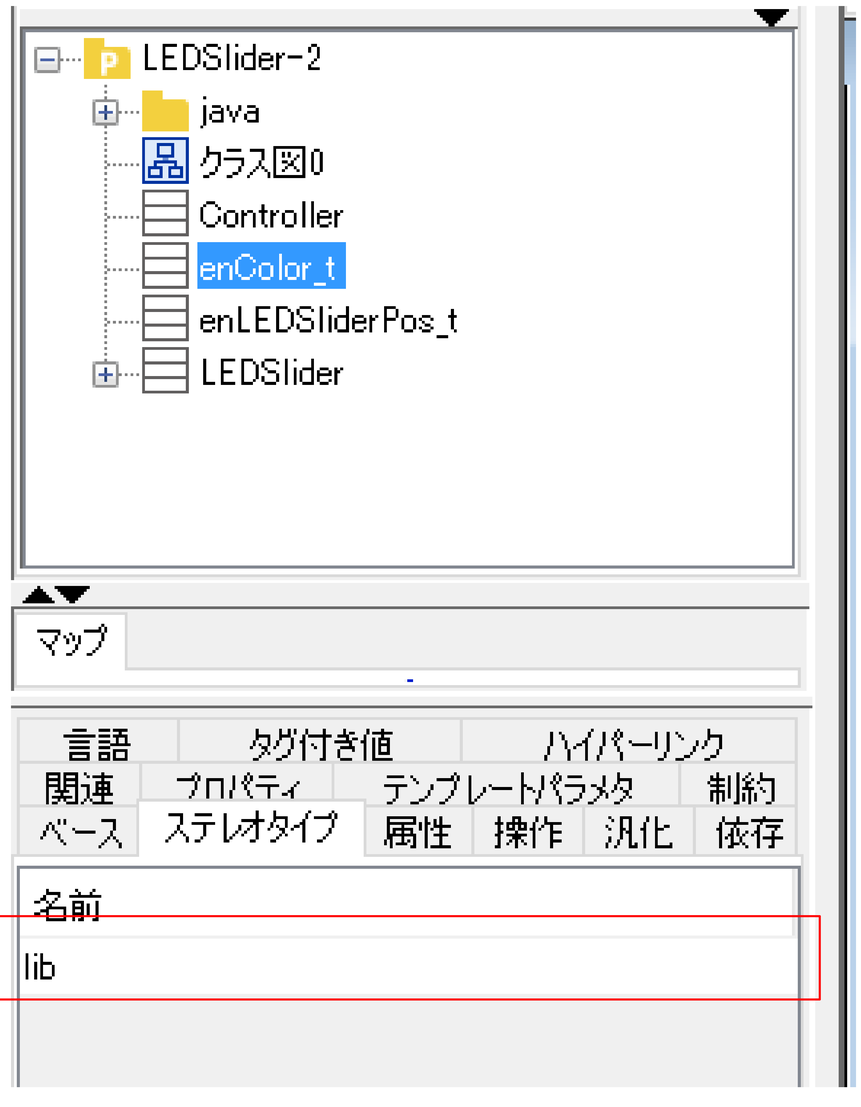
enColor_t型と、enLEDSliderPos_t型は今回作成したenum型です。
型名を打ち込むと、新規作成しますかと聞かれるので「はい」を選びます



モデル図の作成

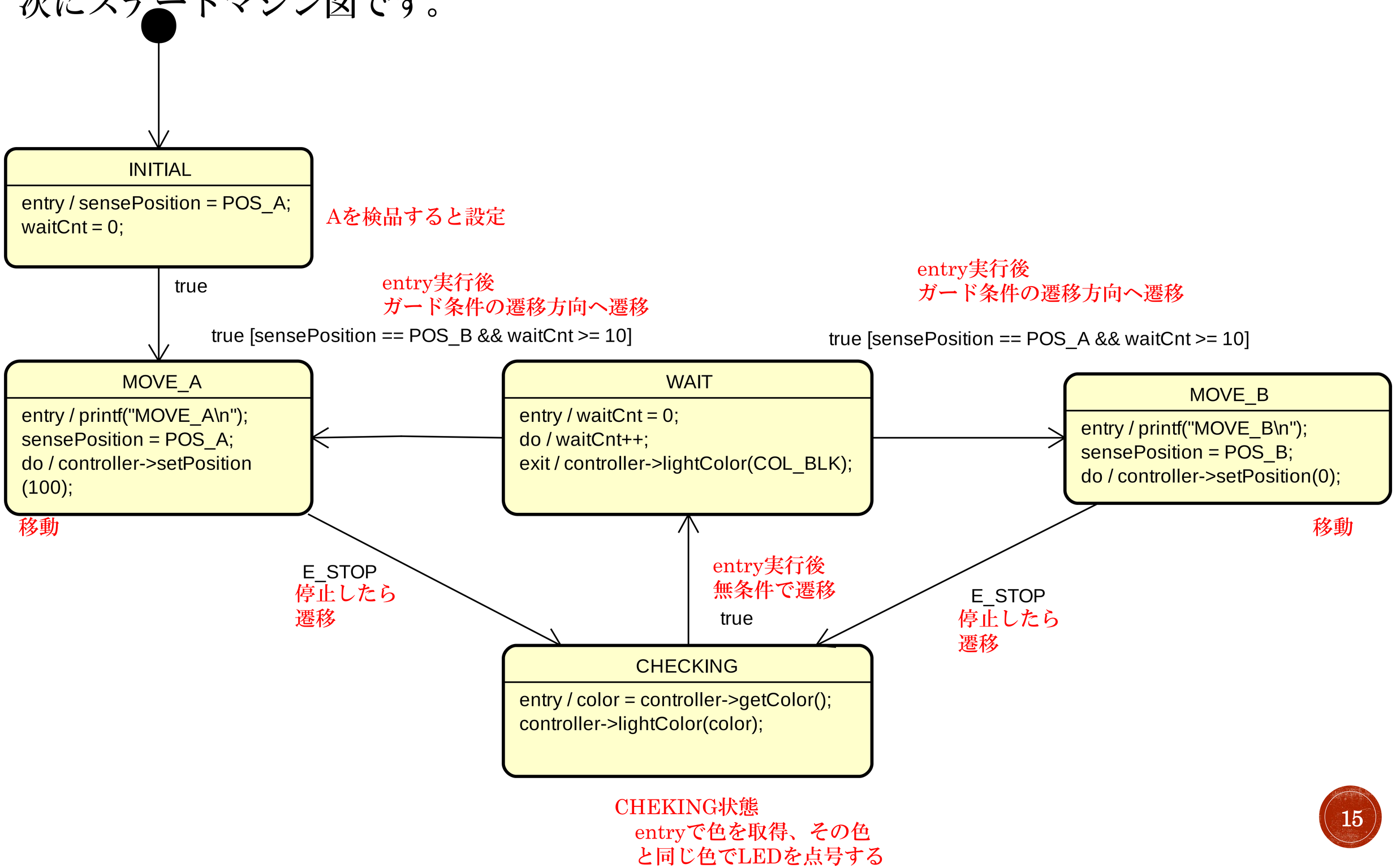
enColor_t型と、enLEDSliderPos_t型は、型だけでクラスではないため、
コード生成をしません。今回はennColor_t型とenLEDSliderPos_t型は
ステレオタイプを「lib」とします



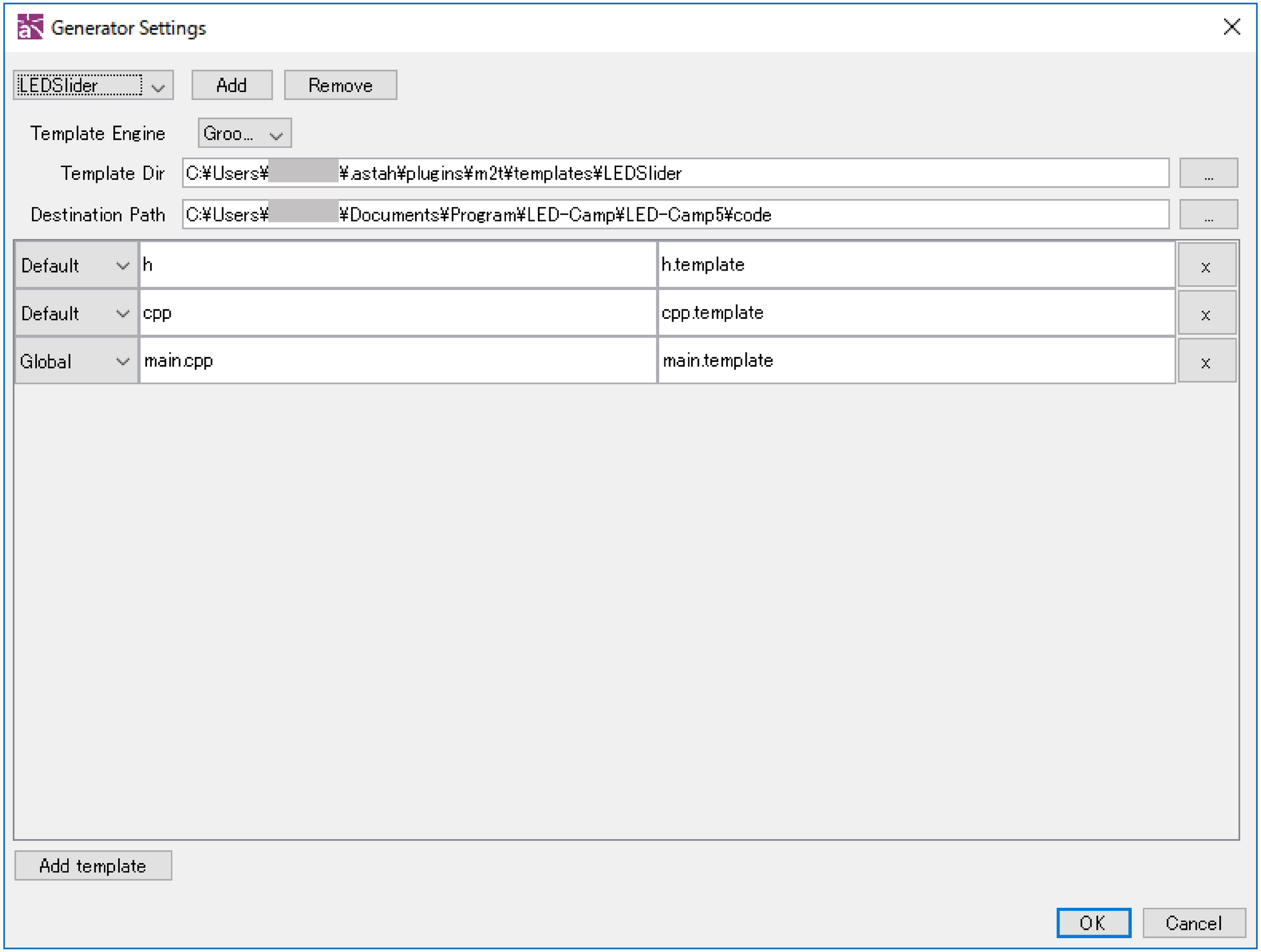


モデル図の作成

次にステートマシン図です。



m2tプラグインの設定変更



Raspberry piへの転送

scp –r –P 22 (転送したいファイル) (転送先アドレス):(転送先ディレクトリ)

例：
scp –r –P 22 code pi@192.168.20.21:~/

- ターミナルソフトを起動する（Teratermや、Poderossa、Cygwin等）
- scpコマンドで転送する
  - 接続するIPアドレスは、配布したネットワーク構成の資料を参考

動作確認

```
ssh (接続先アドレス)
cd code
make JUDGE=(接続先の番号) TEAM=(チーム番号)
sudo ./ledslider

例：
ssh pi@192.168.20.21
cd code
make JUDGE=A TEAM=C
sudo ./ledslider
```

- SSHクライアントを起動する（Teratermや、Poderossa、Cygwin等）
- SSHコマンドで接続する
  - 接続するIPアドレスは、配布したネットワーク構成の資料を参考
- 実行する

開発は続く

一応、それとなく動くものはできました。しかし、要求のすべてを満たしていたでしょうか？

実は、今のシステムは私たちが思い描くバラ色の動作を記述しただけです。もしも許可がでなかったら、もしも検品OKとならなかったら、システムは止まってしまわないでしょうか？

あぁ、また修正しなければ・・・と悲観をすることはありません、少し状態と遷移を追加すれば、ソースコードを書かないでも対応ができるでしょう。

それだけじゃありません、世の中は日進月歩、常に進歩していきます。ほら、工場のシステムのバージョンアップの話がまたすぐそこに・・・