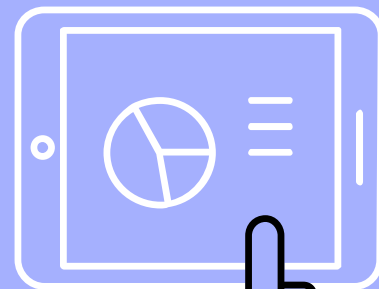
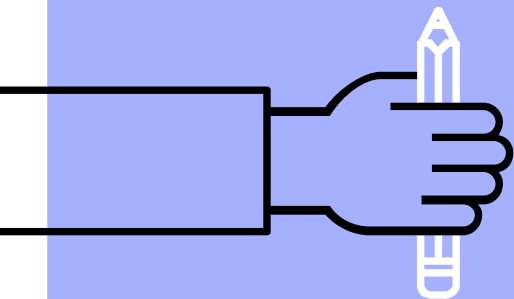
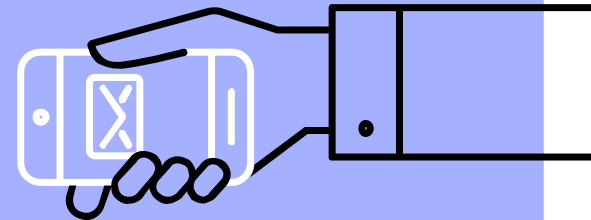
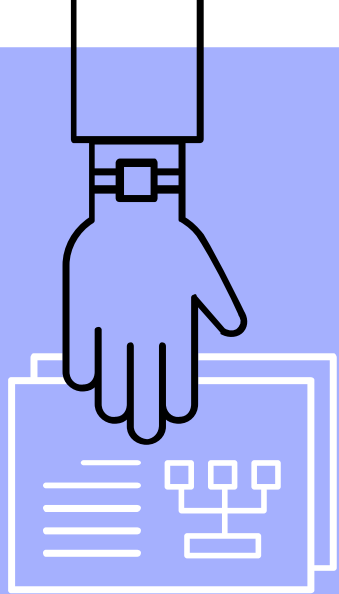
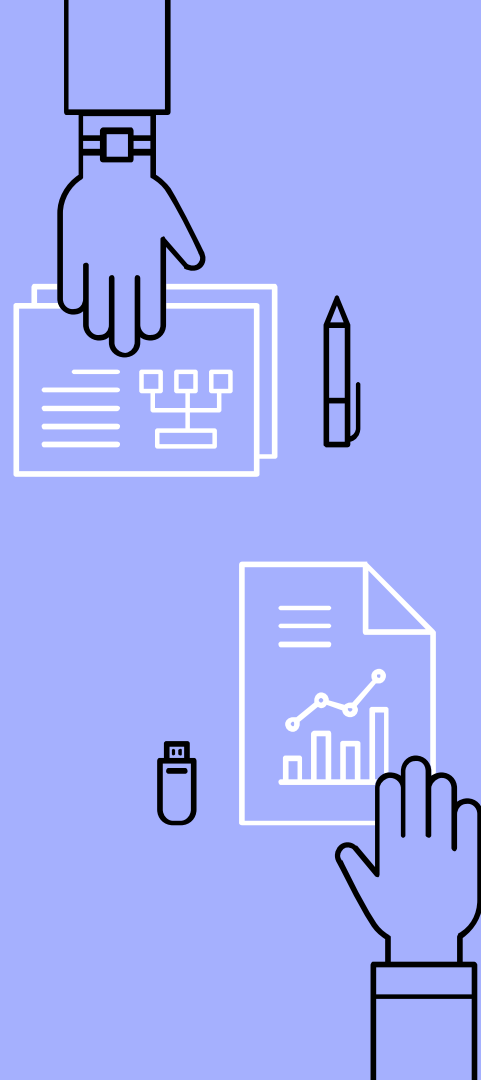


モデル駆動開発 ～実践編～

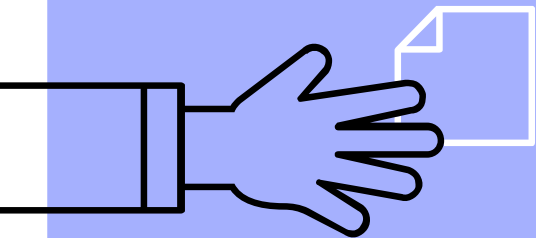
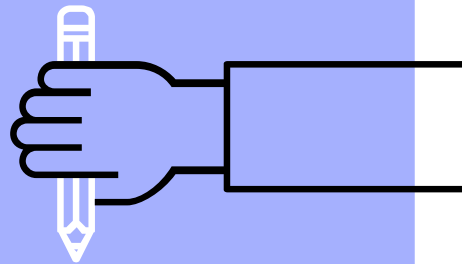


目次

- ▶ LED-Tankを動かす
 - LED-Tank
 - 本番用ファイルのダウンロード
 - Astahからコードの生成
 - Raspberry Piへ転送
 - 動作確認
- ▶ 仕様変更
 - 要求
 - 状態と遷移を考える
 - モデル図の変更
 - 動作確認



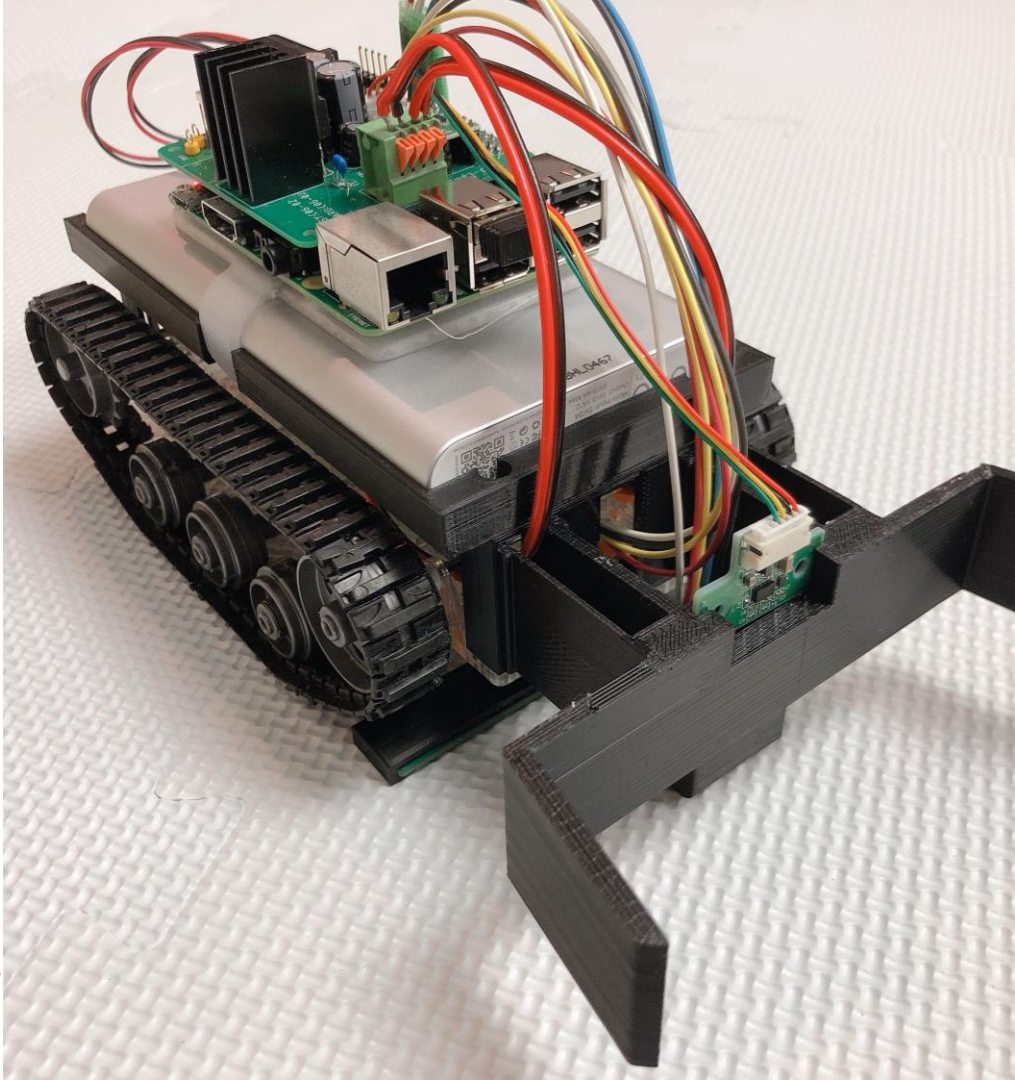
LED-Tankを動かす



LED-Tank

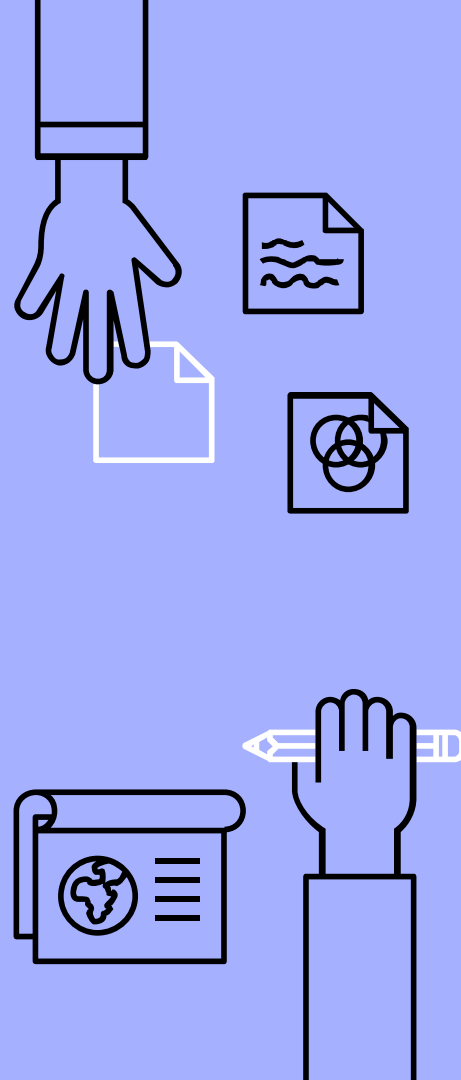
LED-TankとはLED-Campで使用する、キャタピラ型ロボットカーです。LED-Tankはどんな困難な課題でもあっさり解決する可能性を秘めた未来のスーパーロボットです。しかし今のLED-Tankは前進するだけしかできません。このLED-Tankに手を加え、ぜひスーパーロボットに近づけて欲しいと思います。

まずはチュートリアルを動かしてみましよう。



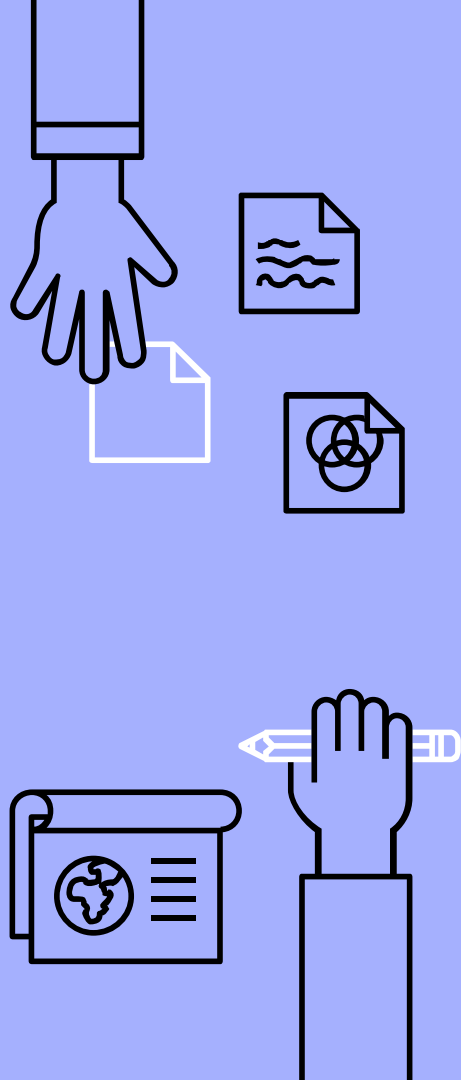
本番用ファイルのダウンロード

- ▶ bit.ly/LED-Camp7
- ▶ 「lecture.zip」をダウンロード



Astahからコードの生成

- ▶ astah-pro を起動
- ▶ File→open
 - LEDTank_lecture1.astaを開く
- ▶ Tools→m2t→settings
 - Template Dir
 - Destinatation path



m2tプラグインの設定変更

Pretank ▼ Add Remove

Template Engine Groovy ▼

Template Dir /home/shoki/LED-Camp7/template ...

Destination Path /home/shoki/LED-Camp7/code ...

Default ▼	cpp	cpp.template	x
Default ▼	h	h.template	x
Global	main.cpp	main.template	x

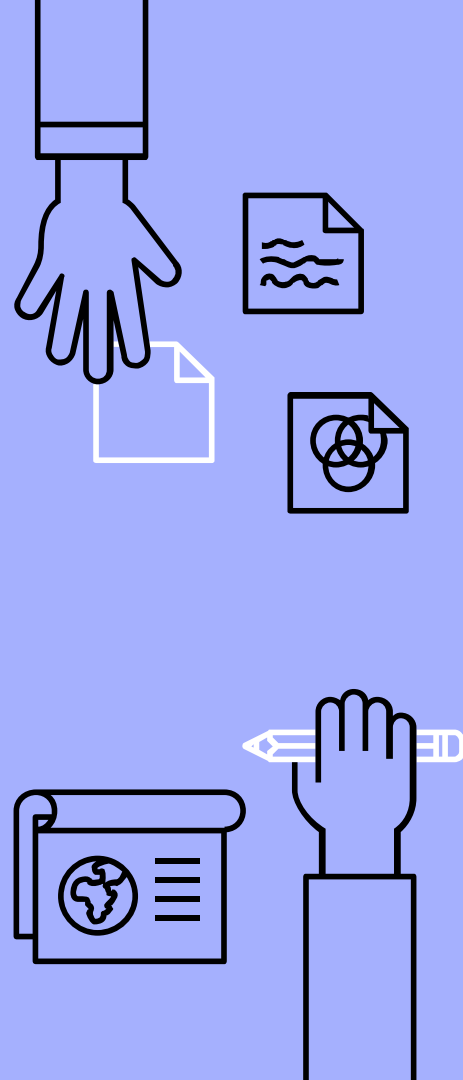
Add template Use 3Way Merge

OK Cancel

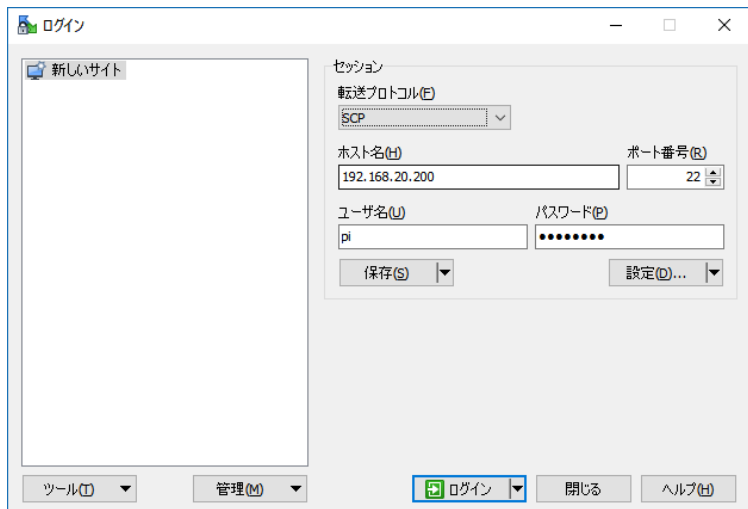
Global忘れずに！

Astahからコードの生成

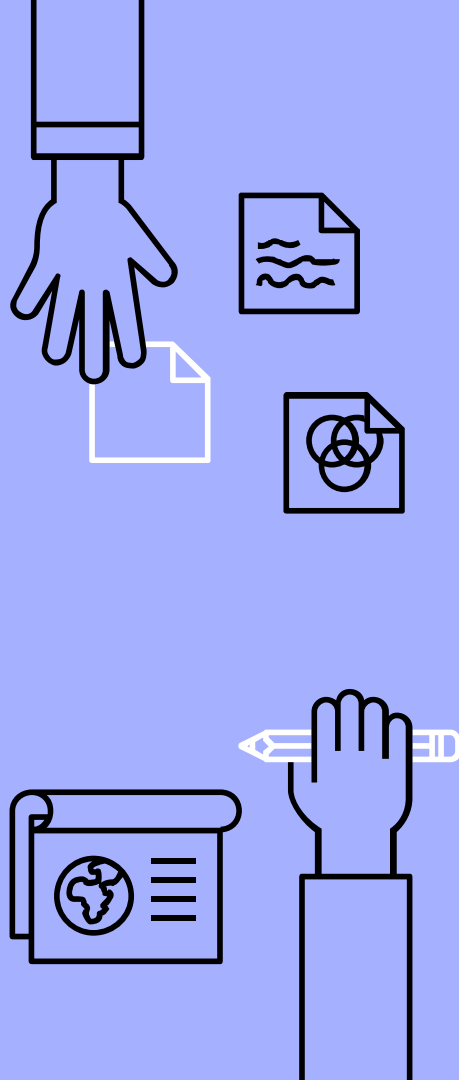
- ▶ Tools → m2t → Generate



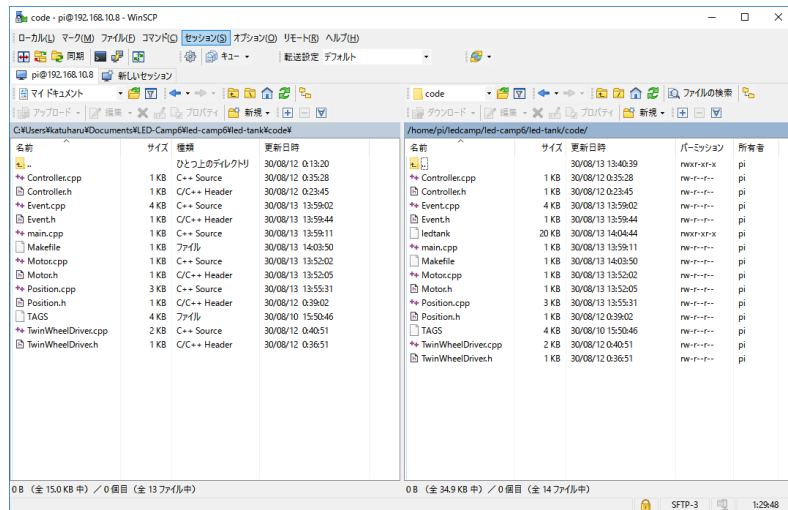
Raspberry Piへの転送 (WinSCP)



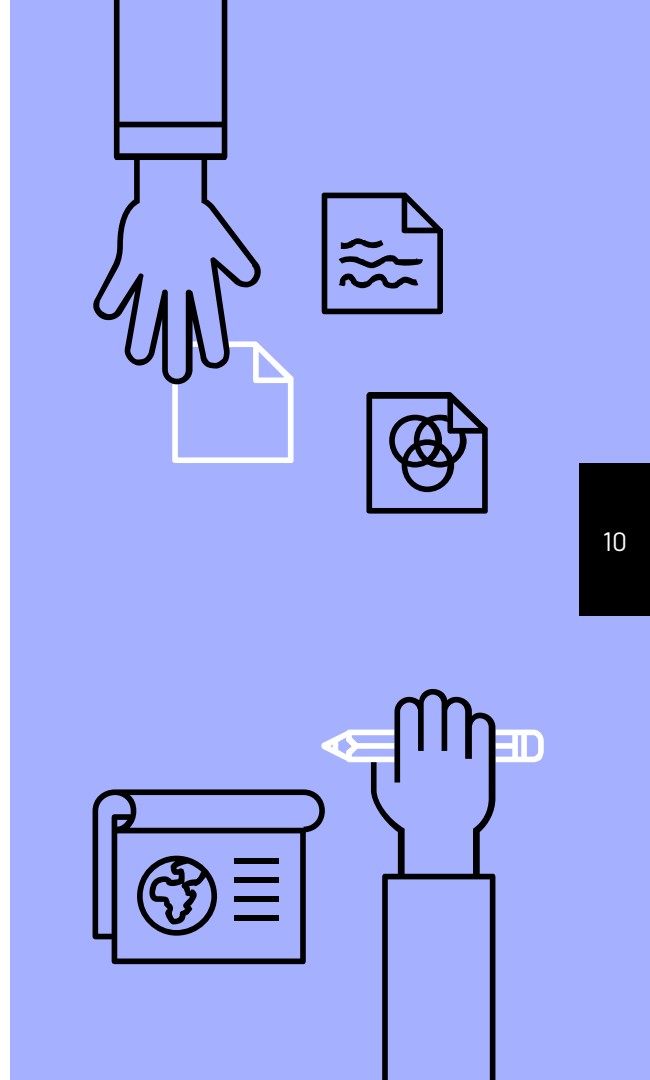
- ▶ WinSCPを起動
- ▶ 転送プロトコル:“SCP”, ホスト名(IPアドレス), ユーザー名:“pi”, パスワードを入力
- ▶ ログイン



Raspberry Piへの転送 (WinSCP)



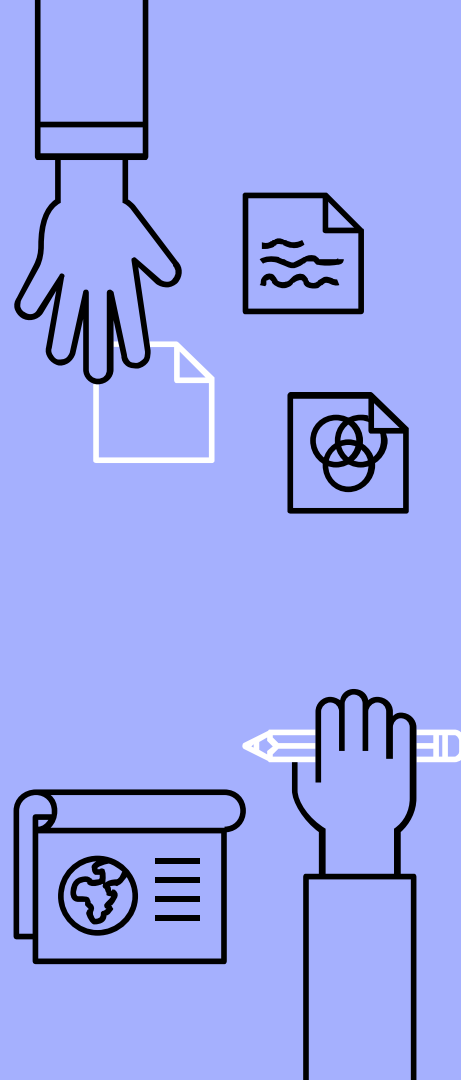
- ▶ 左ビュー→自分のPC, 右ビュー→ラズパイ
- ▶ 右のフォルダでLED-Camp7/codeに移動
- ▶ 転送するファイルを左から右へドラッグ&ドロップ



Raspberry Piへの転送 (コマンド)

```
scp -r -P 22 ~/LEDCamp7/code pi@192.168.20.200:~/LED-Camp7/code
```

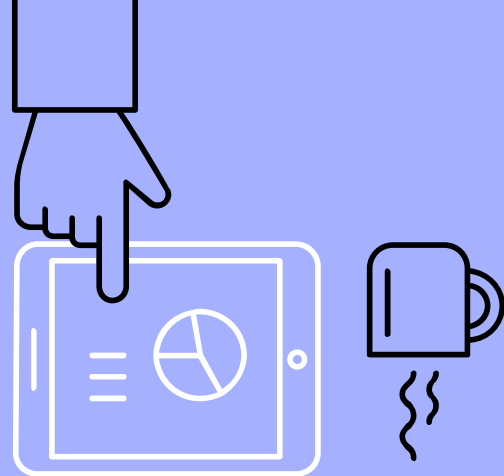
- ▶ ターミナルを起動(例; Teraterm, Cygwin, MINGW)
- ▶ scpコマンドで転送する
 - IPアドレスは配布資料を参考
 - -r : フォルダごとコピーするオプション
 - -P 22 : 22番ポートに接続するオプション



実行

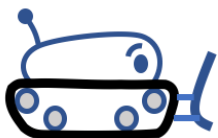
```
ssh pi@192.168.20.200
cd LED-Camp7/code
make -j4
(sudo) ./ledtank
```

- ▶ SSH接続
- ▶ 作業用フォルダに移動
- ▶ コンパイル
- ▶ 実行

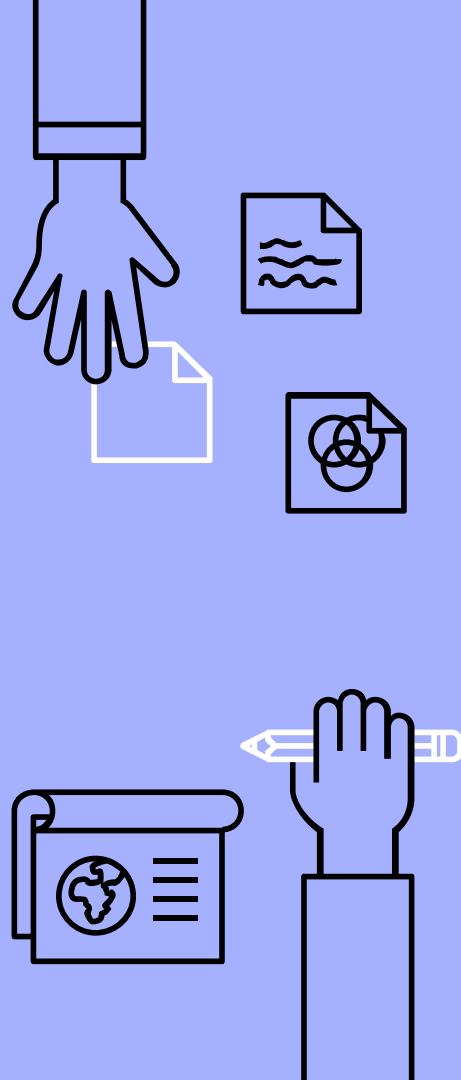


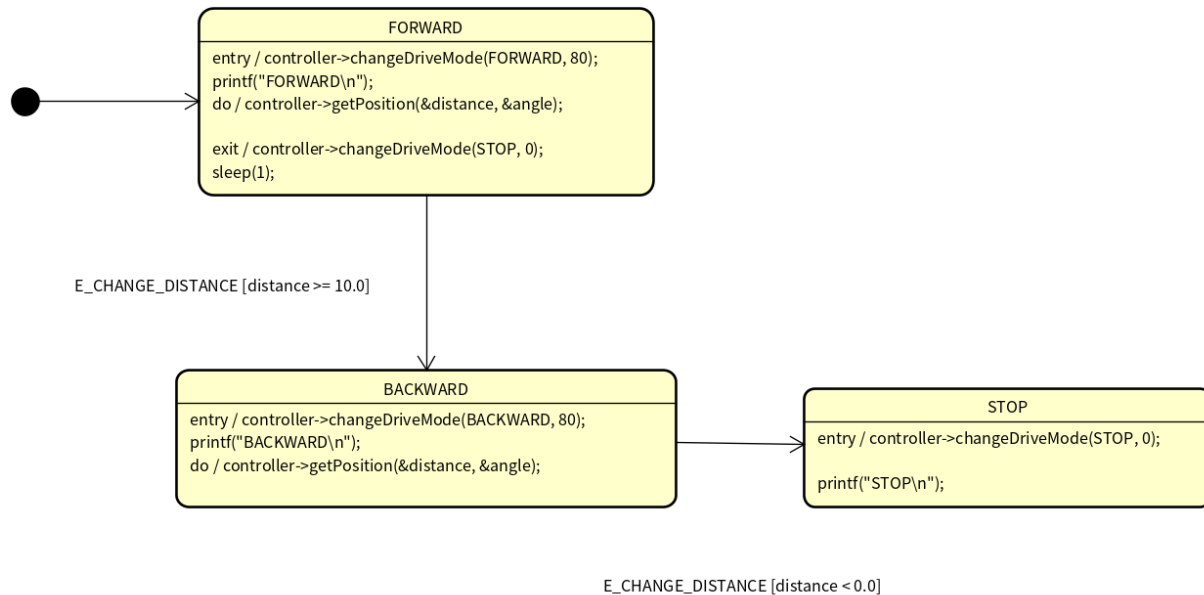
実行

- ▶ 10cm前進
- ▶ 1秒停止
- ▶ 10cm後退して停止

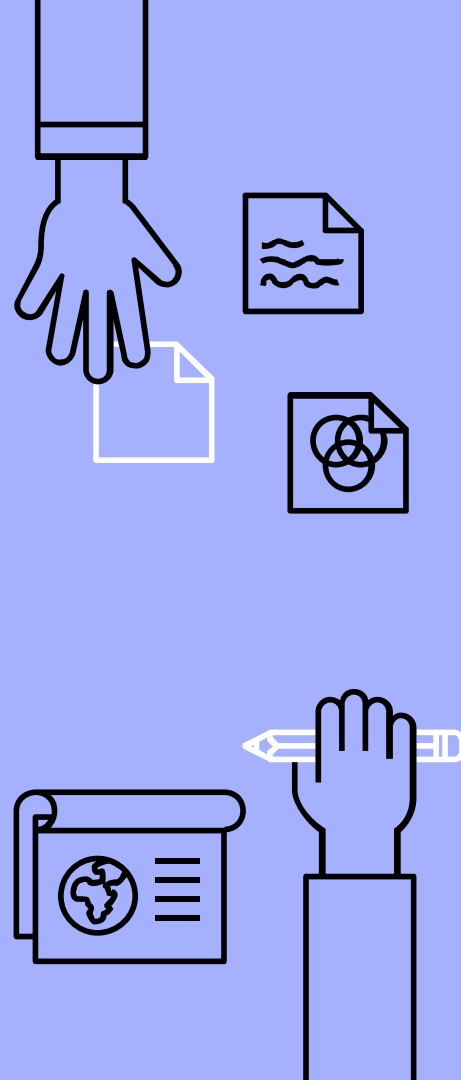


10cm





entry : 状態の最初に1回実行
 do : ループで実行
 exit : 状態を抜ける前に1回実行

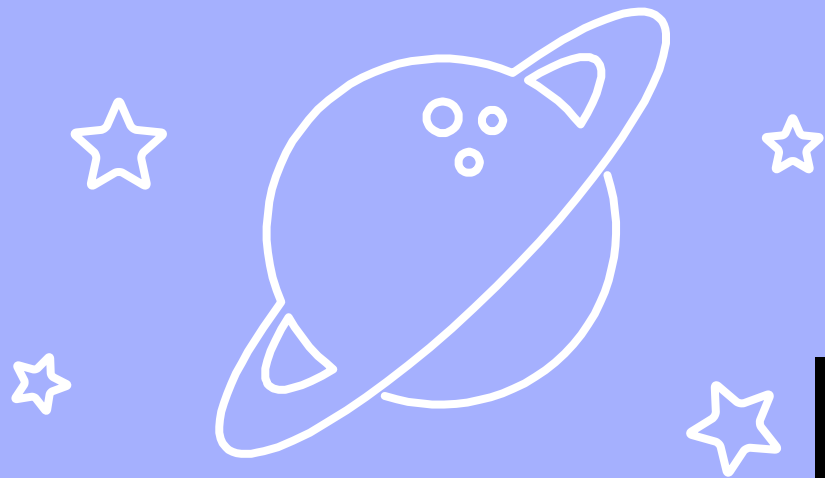


動いたでしょうか？

ここまでは開発に必要な環境の準備やコマンドの紹介を行いました。

この段階ではまだLED-Tankは前進と後退しかできません。

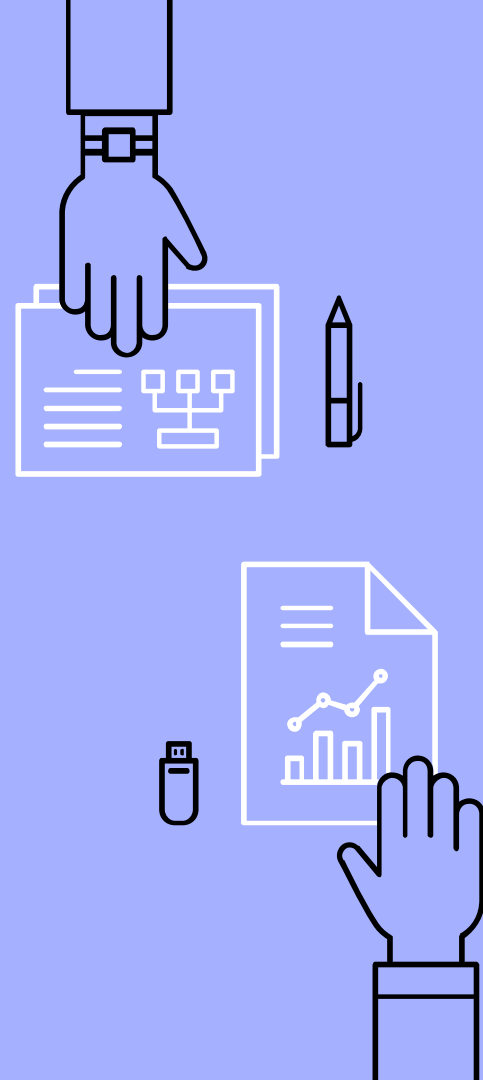
ここからは実際に手を動かしてLED-Tankを開発していきましょう。



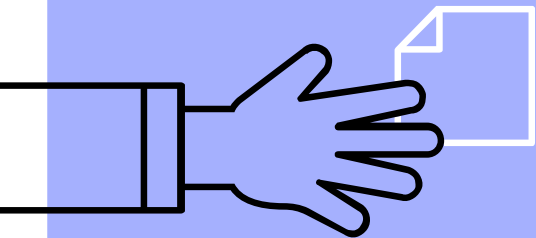
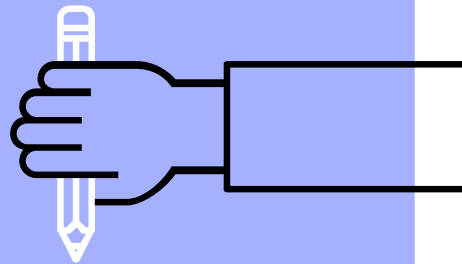
Next Step

目次

- ▶ LED-Tankを動かす
 - LED-Tank
 - 本番用ファイルのダウンロード
 - m2tプラグインの設定変更
 - 動作確認
 - Raspberry Piへ転送&実行
- ▶ 仕様変更
 - 要求
 - 状態と遷移を考える
 - モデル図の変更
 - 動作確認



仕様変更



仕様変更

今まではLED-Tankが前進後退するだけでした。

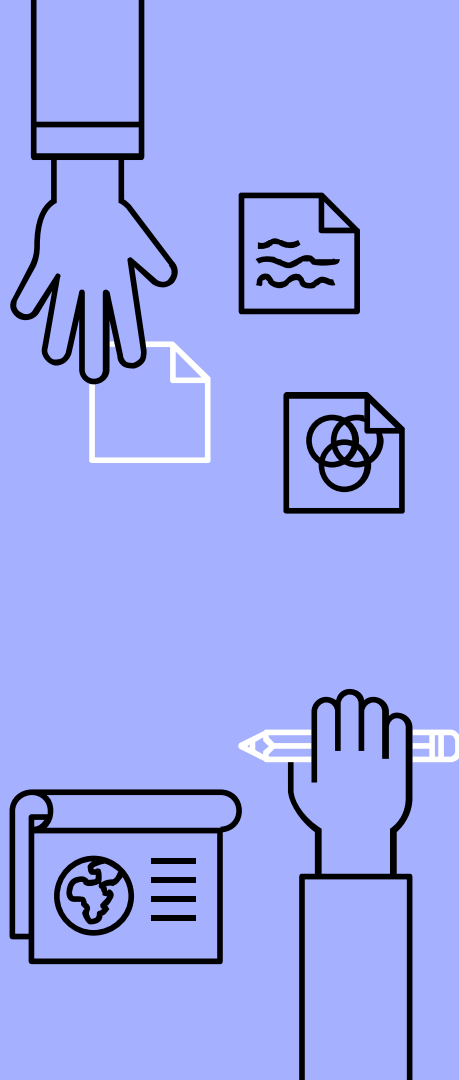
今回のコースには障害物となる「ブロック」が多数存在します。ブロックを1つエリア外に押し出して戻ってくるように変更しましょう。

エリア内外の判定には**ラインセンサ**を用います。

ラインセンサとは対象物が白か黒かを判別するセンサです。

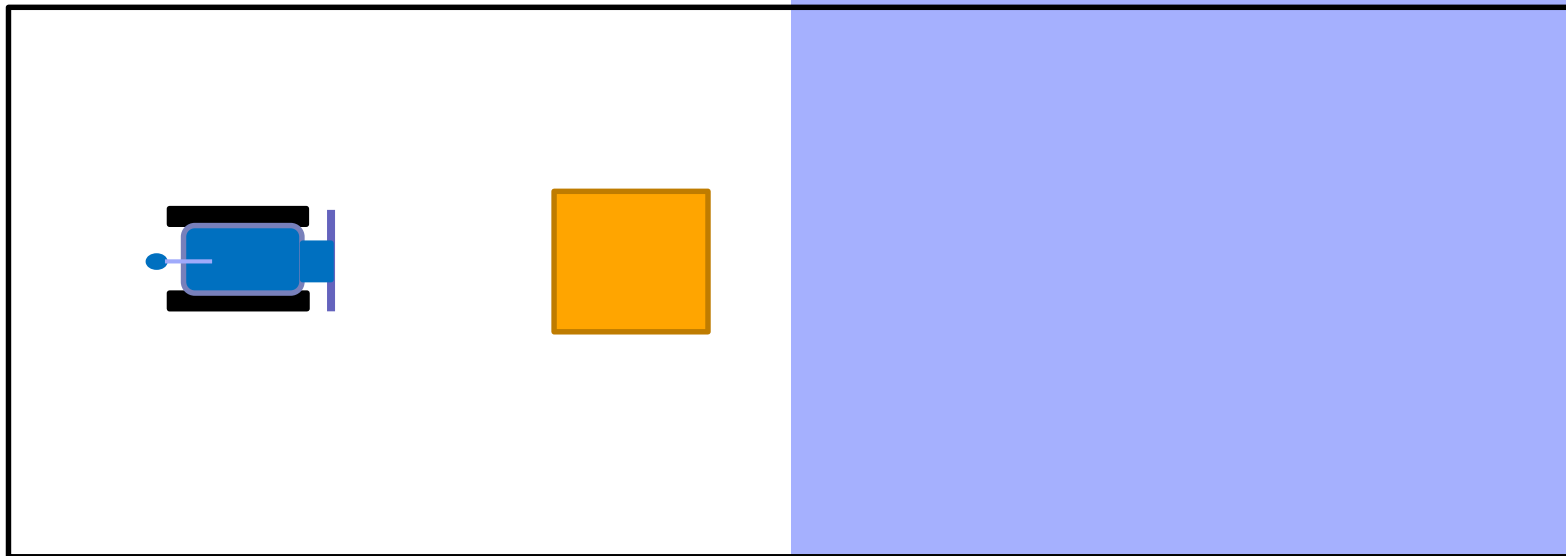
床の色が白か黒か判別してどちらのエリアか把握します。

これとさらに**エンコーダ**と**距離センサ**を組み合わせた動きを考えます。



要求

- ブロックが目のある前提でそのまま前進
- 距離センサが10cm以下検出すると一旦停止 (ブロックに接触)
- 前進し、ラインセンサが床の黒色を検知したら一旦停止
- エンコーダの値から進んだ分だけ後退



開発は続く

一応、それとなく動くものはできました。競技会では課題を通してより複雑な動きが要求されます。私たちが完璧な動作を記述したと想着いても、テストや試走で全く成果が出ないなんてことは日常茶飯事です。しかし、テストを重ね、状態と遷移をうまく追加すればいずれは理想の動きをさせることができるでしょう。

自分達が思い描く理想の動作を目指して3日間頑張っていきましょう。

