



Sweeping Robot Executor

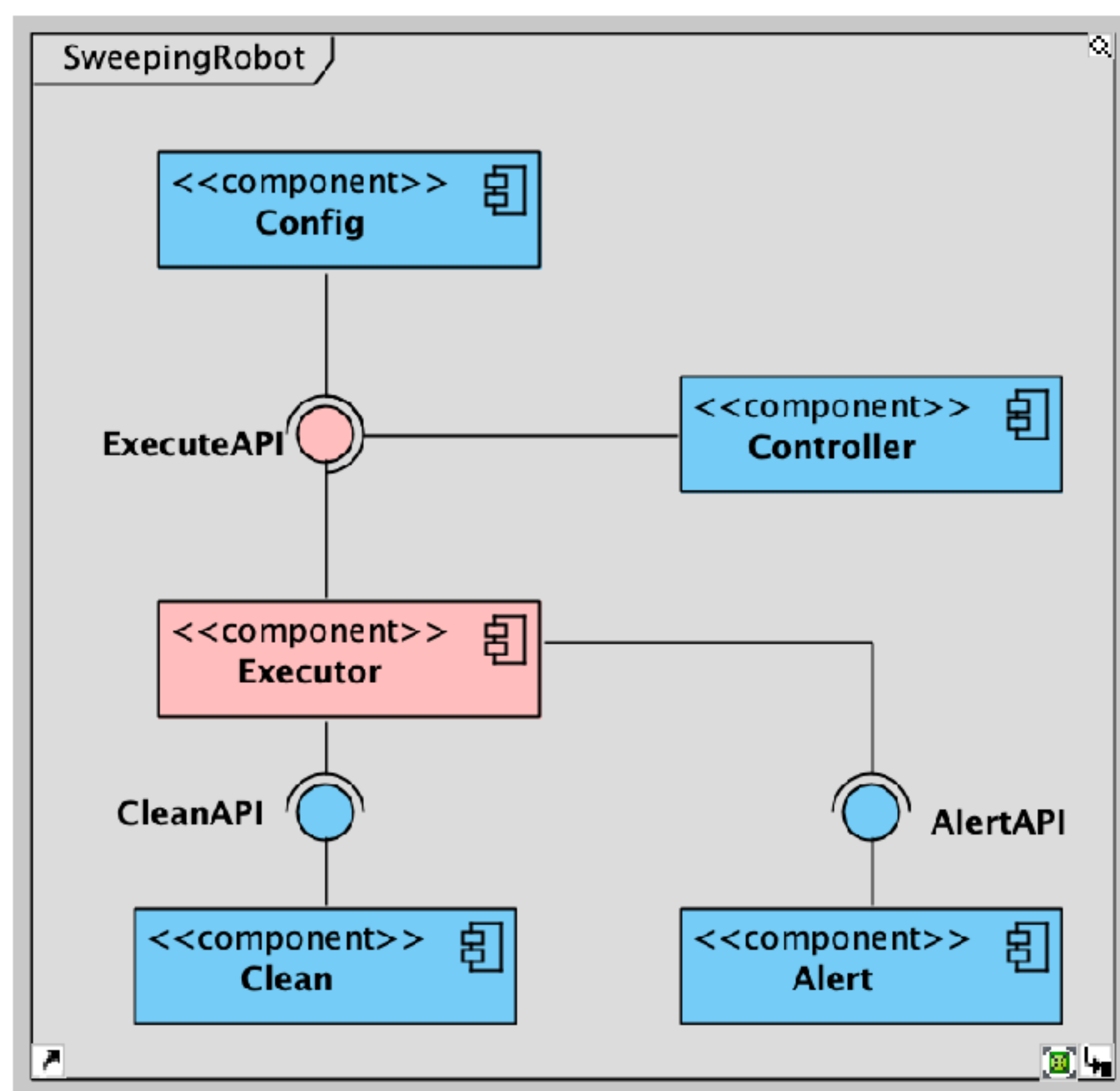


Design Bootcamp



<https://github.com/joycode-bootcamp>

Requirements



☐ 实现扫地机Executor组件：

- ☐ - 由Config组件进行初始化及配置;
- ☐ - 接收Controller组件的各种移动控制指令，维护扫地机器人的位置状态;
- ☐ - 执行命令过程中会使用Alert组件提供的告警上报API上报告警;
- ☐ - 执行命令过程中会使用Clean组件提供的API执行打扫动作;

Bootstrap Project

executor-bootstrap

The bootstrap c/c++ project of design bootcamp.

- setup

```
$ git clone git@github.com:joycode-bootcamp/executor-bootstrap.git  
$ cd executor-bootstrap  
$ chmod a+x ./ccup.sh  
$ ./ccup.sh -ubt
```

<https://github.com/joycode-bootcamp/executor-bootstrap>

Sprint 1

- Executor组件对Config组件提供初始化接口，负责将扫地机初始化在指定位置：
(*x, y, heading*)；其中*x, y*为整数类型，*heading*对应四个方向(东、南、西、北)；
- Executor组件可以执行Controller组件发来的如下动作指令：
 - * *turn right*: 右转90度，位置不变；
 - * *turn left*: 左转90度，位置不变；
- Controller组件可以从Executor组件获取扫地机当前的坐标位置和朝向；
如果Executor组件未被初始化位置，则接口默认返回0，0以及朝向北；
- 要求：自行设计Executor组件对外的接口，实现上述功能，需要有测试用例；



Sprint 2

- Executor组件可以执行Controller组件发来的如下动作指令：

- * *forward*: 沿当前朝向前进一步，方向不变
- * *backward*: 沿当前朝向后退一步，方向不变
- * *turn round*: 旋转180度，位置不变

注意：x轴移动的方向为东西方向，y轴移动的方向为南北方向。



Sprint 3

- Config组件可以给Executor组件设置一个危险的坐标点：(x1,y1)，Config组件保证危险点不会是机器人的初始化位置坐标点；
- 当扫地机移动到危险坐标点的时候，调用Alert组件的接口：“`void alert(int type, int x, int y)`”触发告警活动；其中type值填3，表示“经过危险坐标告警”；
- 反复在同一危险坐标点上执行命令，重复触发告警；



Sprint 4

- Config组件可以给Executor组件设置一个需要清扫的坐标点：(x1,y1)，Config组件保证清扫点不会是机器人的初始化位置坐标点；
- 当扫地机移动到需要清扫的坐标点的时候，调用Clean组件的接口：“`void clean(int x, int y)`”接口触发清扫活动；
- 反复经过清扫坐标点只调用一次clean接口；



Sprint 5

- Config组件可以反复调用接口配置/取消Executor组件告警坐标点和打扫坐标点，配置的告警坐标点和清扫坐标点不存在重复；
- 由于资源限制，Executor最多只能存在16个已配置的待告警或待清扫点（告警点和清扫点共享）；Config组件通过配置取消告警或者清扫坐标点后，Executor则回收对应的配置资源；
- 当Executor组件触发清扫接口后，可以自动回收一个坐标点的配置资源，用于Config组件继续配置新的告警坐标或者清扫坐标；



Sprint 6

- Executor组件可以执行Controller组件发来的如下动作指令：
 - * *forward n step*: 沿当前朝向前进n步，方向不变
 - * *backward n step*: 沿当前朝向后退n步，方向不变
 - * *repeat command n times*: 将指定的任一动作指令重复执行n次；



Sprint 7

- Executor组件可以执行Controller组件发来的如下动作指令：
 - * *{command1, command2, ...}*: 按照顺序执行一组任意指令；
- Executor组件提供的*repeat* 和 *command* 列表指令，能够支持任意嵌套，如：
 - * *repeat {command1, command2, ...} n*: 将*command* 列表重复*n*次；
 - * *{command1, repeat command n, ...} n*: *command* 列表中可以包含*repeat*指令；
 - * *{command1, {command1, command2, ...}, ...} n*: *command* 列表可以嵌套；



Sprint 8

- 由于手机APP端需要可视化扫地机的运动路径和状态，因此Executor组件需要将每次移动后的坐标位置，以及触发的行为，转换成json字符串，然后调用wifi组件的“`void send_json(const char* json)`”接口发送出去。
json字符串格式样例为： ``{"x" : 3, "y" : 5, "action" : "clean"}``
其中action可以为“none”，“alert”或者“clean”，分别对应坐标点上“没有发生活动”，“触发告警活动”或者“触发清扫活动”；
- 对于wifi组件的接口，需要建立git仓库，修改executor组件的编译时依赖，通过构建触发自动获取依赖的接口代码；



Sprint 9

- 为了节省wifi流量，Config组件可以选择配置Executor组件发送给wifi组件的消息为二进制格式，或者json格式（不配置默认依旧采用json格式）；

二进制发送调用接口：

```
"void send_msg(const struct WifiMsg* msg)"
```

```
"struct WifiMsg{int x, y, action;};"
```

其中action： 0表示没有触发活动，1表示触发告警，2表示触发清扫；



Sprint 10

*Executor*组件同时要支持低端扫地机，因此需要能够做如下的功能裁剪：

- 低端机不支持*Controller*组件执行批量命令：{*command1*, *command2*, ...}；
- 低端机不支持*Config*组件配置告警点以及触发告警活动的功能；
- 低端机仅支持*Config*组件配置8个清扫点；
- 低端机不支持配置对*Wifi*组件发送二进制消息；

*Executor*组件能够对低端和非低端机同时出版本；对低端机的版本中要能够把上述功能从配置能力、二进制以及源码上都能裁剪掉；





THANKS

Contact: e.wangbo@gmail.com

<https://github.com/joycode-bootcamp>