1.页面登陆:

Server接收：

topic: login 登陆验证 类型：string

data: {‘pad’:2017,”sn”:”00122”,”name”:”username”} 类型：json

server 回复：

topic: msg\_reply server端返回信息 类型：string

data.type:login\_reply 登陆返回信息 类型：string

data.content: 返回内容 类型：string

content：

1. login\_success 登陆成功 类型：string
2. login\_failure 登陆失败 类型：string
3. sn 序列 string

2.控制机器人流程：

A．请求机器人

Server接收：

topic: applyRobot 请求机器人 类型：string

data:

1.nickname 工位名 类型：string

1. sn 任务序列 string

Server返回：

topic:msg\_reply server端返回信息 string

data:

1.type: msg\_ApplyRobot 请求返回信息 string

2.id 请求到机器人id string

3.tablename 请求到工位的名字 string

4.sn 任务序列 string

1. reply
   1. 1 请求成功 init
   2. 0 请求失败 init

B. 锁定机器人

JSON.stringify({"device": "table","topic": "lock", "data": { "id": 小车id, "nickname": 工位名, "nk": 任务序列})

Server接收：

topic:lock 锁定机器人 string

data ：

1.id 机器人id string

2.nickname 工位名 string

3.sn 任务序列 string

Server返回：

topic:msg\_lock server端返回信息 string

id 锁定机器人id string

data：

1.type：msg\_lock 锁定机器人返回信息 string

2.reply： 锁定机器人返回值 init

1 锁定成功

0 锁定失败

3.tablename 工位名 string

4.sn 任务序列 string

C.移动机器人

JSON.stringify({"device": "table","topic": "target", "data": { "id": robotID, "nickname": tablename, "type": "", "sn": “00202”})

Server接收：

Topic：target 移动机器人 string

Data：

1.id 机器人id string

2.nickname 工位名 string

3.type 工位类型 string

4.sn 任务序列 string

Server返回：

topic:msg\_reply server端返回信息 string

id 机器id string

data：

1.type：msg\_RobotReached 机器人到达反馈 string

2.sn 任务序列 string

D.发送任务

JSON.stringify({"device": "table","topic": "mission", "data": { "id": robotID, "content": “mission1”, "sn": “00202”})

Server接收：

Topic：mission 工位任务 string

Data：

1.id 机器人id string

2.content 任务名 string

3.sn 任务序列 string

Server返回：

topic:msg\_reply server端返回信息 string

id 机器id string

data：

1.type：msg\_mission 任务反馈 string

1. content

workDone,任务名（misson1） 任务完成 string

workFailure,任务名(mission1) 任务失败 string

25，camera 货架id（25）string

workFailure,camera 扫描失败 string

3.sn 任务序列 string

扫描二维码：

1 姚工发送misison1

服务器返回：

成功：

1.camera,货架id

2.mission1,workDone

失败：

mission1,workFail

2 姚工发送mission2

服务器返回：

成功：

1.camera,货架id

2.mission2,workDone

失败：

mission2,workFail

F．停止机器人

Server接收：

Topic：cancel 停止机器人 string

Data:

1.id 机器人id string

2.sn 序列 string

Server返回：

topic:msg\_reply server端返回信息 string

id 机器id string

data：

1.type：msg\_RobotReached 机器人到达反馈 string

2.sn 任务序列 string

E. 解锁机器人

JSON.stringify({"device": "table","topic": "unlock", "data": { "id": robotID})

Server接收：

topic：unlock 解锁机器人 string

data：

1.id 机器人id string

2.sn 序列 string

Server返回：

topic:msg\_reply server端返回信息 string

id 机器人id string

data：

1.type：msg\_unlock 解锁机器人返回信息 string

2.locked： 解锁状态值 init

1 解锁失败

0 解锁成功

3.sn 序列 string

3.获取机器人位置

机器人位置是实时传递的

{device: "table", topic: "pose", id: "WT\_001", data: "{"y" :-0.8816405465350527, "x" : 1.9060866026786485 , "a" :-0.7048353026749968,"w":0.7093709862286769}" }

Server实时返回：

Topic：pose 机器人位置信息 string

id 机器人id string

data：

1.x 机器人x坐标 float

2.y 机器人y坐标 float

3.a 机器人a坐标 float

4.w 机器人w坐标 float

其中a和w代表一个角度，他们的换算公式为：

**th（角度）=2\*atan2(a,w);**

**a=sin(th/2),w=cos(th/2);**

**其中：-Pi<=th<=Pi; 0<=a<=1;0<=w<=1;**

4.机器人上线、下线通知

Server返回：

机器人上线：

topic：msg

data：

1.type：msg\_ robotbutton 机器人上线 string

2.uuid 机器人id string

机器人下线：

topic：msg

data：

1.type：msg\_robotdownline 机器人下线 string

2.uuid 机器人id string

1. 机器人返航（替代之前的task\_finish）

JSON.stringify({"device": "table","topic": "reversal", "data": { "id": robotID, "sn": “00202”})

Server接收：

Topic：reversal 返航 string

Data：

1.id 机器人id string

2.sn 任务序列 string

Server返回：

topic:msg\_reply server端返回信息 string

id 机器id string

data：

1.type：msg\_reversal 反馈 string

1. state：
   1. 0 已解锁
   2. 1 未解锁

3.sn 任务序列 string

6.返回报错信息

JSON.stringify({"device": "table","topic": "alarm","id":robotID,"tablename":"", "data": {"type":"msg\_exceptionInfo"  "sn":null,"content":"moving\_state,has\_obstacle"})

    "content":

        "battery\_level\_low,20"          string      电量20%

        "battery\_level\_badly\_low,15"    string      电量15%

        15%~~9% 为每少%1上报一次，直到9%

        "battery\_level\_badly\_low,9"     string      电量9%

        "moving\_state,has\_obstacle"      string     行走时有障碍

        "moving\_state,no\_obstacle"      string      行走时障碍移除

        "near\_target,has\_obstacle"      string     到达目标时有障碍

        "near\_target,no\_obstacle"       string     到达目标时障碍移除

**工作流程：**

1. 申请小车———》锁定小车————》移动小车。
2. 执行位置任务：

位置：在实际货架停放位置的前面，小车要到达的目标点

位置点：小车到这个位置会自动触发一条任务

业务控制方式，到一个点（Target），做一个任务（Mission）,再到一个点再做一个任务，任务是由若干动作组成，由木蚁软件编写完成，业务通过任务ID进行调用。

**顶起货架：**

**位置1（货架前）--->mission1（钻进去，扫ID上传）------->mission2（顶货架） ------>位置2（移出到道路上）----->mission3（随动摆正）**

1. 姚工发送移动命令控制小车移动到位置1，
2. 发送mission1任务（姚工在接收到小车到达信息后向服务器发送mission1关键字）
3. 小车执行mission1任务（旋转——》钻入货架——》扫码——》返回货架id号码），任务完成后，任务完成后，姚工收到 “workdone，mission1”
4. 姚工根据返回id判断是否需要顶升，需顶升向服务器发送命令mission2(顶升任务)

(5)小车会在顶升完成后向姚工返回顶升完成信息 “workdone,mission2”

姚工收到小车到达信息后发送target命令，把小车移出货架到位置2（移出位置）

（如果不顶升，上位直接发送target，把小车移出货架到位置2）

(6)小车到达位置2，启动随动任务mission3, (需要启动随动使小车和货架保持同向，姚工向服务器发送随动任务名，小车执行随动)。

(7)随动完成后小车向姚工发送完成信息，”workdone,mission3”

1. 姚工接收到任务完成信息，向小车发送target目标控制小车去位置3放货架。

**放货架：**

**位置3（放料道路点）----->mission4（随动） ----->**

**位置4（放料点）---->mission5（放下货架） ----->**

**位置5（道路点）----->返航并解锁机器人**

1. .小车到达位置3，发送任务4 mission4(需要启动随动功能，朝向位置4，需要姚工发送随动任务)

(2)随动完成后小车向姚工发送完成信息，”workdone,mission4”, 姚工向小车发送target进入位置点4，

(3)到达位置点4，下发下降任务mission5 （姚工控制小车放下货架）

(4)货架放下后收到，任务完成指令，”workDone,mission5”，需要小车移出货架，姚工发送target位置点5移出货架。

(5)移出货架后，姚工向服务器发送解锁命令，无任务时，再发送任务结束命令，让小车自动返航。

该方案主要是把一些小车动作封装成模块，我们可以通过专门的上位机软件，在设置点的同时，把这些动作添加到这个点上，当小车停到这个点时，姚工只需要发送启动哪个动作的命令就可以。

例如：需要去点A顶起货架，我们通过上位机把A点的所有可能要执行的动作都绑定上去，

mission1（左转顶起货架）/mission2（右转顶起货架）/mission3（左转放下货架）/mission4（右转放下货架），当小车到达点A时，姚工发送mission1，机器就会执行向左转顶起货架的动作。

这样做好处就是，我们只需要发送位置和任务名，小车就可以自动完成一系列顶货架或放货架动作。如果位置有变动，只需要通过专门的上位机软件重新取点和绑定一系列动作，姚工需要控制小车去该位置执行什么任务都可以自主来选择，不需要另外编写程序。