

# FC 配送机器人通信协议-socket

机器人本体中控与本体 app，socket 网络通信。

## format

协议数据链路层描述

使用 TCP 通信，下位机为固定 ip：192.168.1.3，  
指令通讯端口为：5656 负责传输所有指令响应，  
数据传输端口为：6868，传输地图信息及图片等流信息，防止主通讯端口阻塞。

下位机（机器人本体中控）为 server。

## request

参数名	长度	说明
frameHead	2 byte	0xFC 0xAA
type	1 byte	数据类型（可参考 3.2）
cmdFunction	2 byte	指令功能
msgId	4byte	消息id，标记数据的唯一性，便于后续追踪排查。示例：HHmmssSSS
encryption	1byte	加密压缩方式
statusCode	1 byte	状态码
dataLength	4 byte	数据总长度（帧头 + 数据长度）
GZipDataLength	4 byte	压缩前数据的长度
MD5	16 byte	data区MD5的校验值
Data	N byte	数据内容

response

参数名	长度	说明
frameHead	2 byte	0xEE 0xAA
type	1 byte	数据类型（可参考 3.2）
cmdFunction	2 byte	指令功能
msgId	4byte	消息id，标记数据的唯一性，便于后续追踪排查。示例：HHmmssSSS
encryption	1byte	加密压缩方式
statusCode	1byte	状态码
dataLength	4 byte	数据总长度（帧头 + 数据长度）
GZipDataLength	4 byte	压缩前数据的长度
MD5	16 byte	data区MD5的校验值
Data	N byte	数据内容

type 数据类型说明

类型取值	类型描述
0x00	数据流类：用于数据传输，比如图片，使用数据传输端口。
0x01	json指令类，用于控制下位机运行、设置参数，使用指令通讯端口。
0x02	protobuf 指令类，用于控制下位机运行、设置参数，使用指令通讯端口。（仅预留，暂不实现）
0x03	主动推送类指令

response 的 Data 默认包含

字段	数据类型	描述
cmdTime	long	从1970年1月1日开始，经历的毫秒数
resultCode	int	底盘返回状态码
resultMsg	String	底盘返回硬件信息

## 4.2 底盘通信错误码

代码	描述
1001	响应成功
2001	硬件协议错误
2002	硬件通信超时
2003	参数不合法
2004	消息过期

## 其他备注说明

### 关于 cmdFunction

现有使用：

0x0200~0x0300 是直接透传给上装的

主动上报(type = 0x03)的包有 0x0340

0x036x 往后的未使用

CommRequestHandle 对新增接口使用新的序列，  
回复获取接口从 0x0A01=2561 开始递增，  
主动上报接口从 0x0B01=2817 开始递增。

## 2023.5 新增补充

### 1. 获取机器人序列号 ~~0x0A01~~ 【2024.1 修改】

[FC 配送机器人 SN 管理](#)

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A01, // =2561
```

**request**

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
  "data": {  
    "robotSn": "100132001234567" // 15 位资产编码，后 7 位流水号，  
    根据供应商区分  
  }  
}
```

## 2. 根据站点 ID 获取站点信息 0x0A03

数据端口: 5656

```
{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "resultCode": 1001, // 1001： 响应成功
  "resultMsg": "下发成功" // 底盘通信状态信息
  "data": {
    "stationName": "hhh",
    "stationId": 0, // ?时间戳生成 ID
    "stationType": 0, // [0-导航 1-充电 2-电梯 3-闸机 4-自动门 5-消防门]
    "x": 0.5,
    "y": 0.6,
    "yaw": 0.2
  }
}
```

### 3. 发起一键回充 0x0A04

数据端口: 5656

```
{  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

## 4. 取消一键回充 0x0A05

描述：

- 任何状态下都会响应，停止运动，使回充状态进入“空闲”状态。
- 若在充电桩上，先执行退桩，再停止运动

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A06, // =
```

#### request

```
{  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
{  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
  "data": {  
    "rechargeStatus": 1, // 4：开始回充导航（正在去往充电桩的路上）  
    1：到达退桩点（已导航到充电桩前方） 2：到达充电点（已对桩完成） 3：开始充电  
    （已在桩上开启充电桩输出）0：未开始回充（其他：空闲、下桩）  
    "errorStatus": 0 // 0：成功 1：异常  
  }  
}
```

## 6. 主动上报回充过程和结果 0x0B01

描述：状态切换时会主动上报一次（可能存在重试进行自动切换状态，进而重复上报的情况）

- 异常，"errorStatus": 1（"rechargeStatus" 为前一个状态，可不关注）

异常状态后，本体不会再主动执行动作，只响应指令才开始动作（开始回充等）。

- 无异常，"errorStatus": 0， "rechargeStatus": 正常顺序：0 4 1 2 3

0：未开始回充（其他：空闲、下桩）

4：开始回充导航（正在去往充电桩的路上）

1：到达退桩点（已导航到充电桩前方）



数据端口: 5656

```
"type": 0x03,  
"cmdFunction": 0x0B01, //
```

### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": {  
    "rechargeStatus": 1, // 0: 未开始回充 4: 开始回充导航 1: 到达退桩点 2: 到达充电点 3: 开始充电  
    "errorStatus": 0 // 0: 成功 1: 异常  
  }  
}
```

### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

## 7. 一键上报地图点位信息

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0308, //
```

**request**

```
"type": 0x01,  
"cmdFunction": 0x0309, // =
```

### **request**

```
{  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

### **response**

```
"type": 0x03,  
"cmdFunction": 0x0342, // =
```

## request

```
{  
    "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
    "data": {  
        "upper_status": 14, // 10~99 机器人运行的状态量  
        "lower_status": 1401 // 子状态 当前可用于确定处于 14 状态 (在执行任务) 下的具体子状态  
        "broke_code": 123456 // 机器人出故障时的具体故障代码  
    }  
    // uint16 NO_LOWER_STATUS = 0      # 无子状态  
  
    // uint16 UNINITIAL      = 10      # 未初始化, 默认状态  
    // uint16 MAPPING        = 11      # 建图中  
    // uint16 IDLE           = 12      # 空闲中  
    // uint16 RECHARGE       = 13      # 回充中  
    // uint16 TASK           = 14      # 在执行任务  
  
    // uint16 PARKING        = 97      # 急停中  
    // uint16 RECOVER        = 98      # 已恢复  
    // uint16 BROKEN         = 99      # 故障  
  
    // uint16 MOVING         = 1400    # 运动中  
    // uint16 ARRIVED        = 1401    # 达到目标点  
    // uint16 ENTER_ELEV     = 1402    # 进电梯  
    // uint16 EXIT_ELEV      = 1403    # 出电梯  
}
```

## response

```
{  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

## 10. 获取任务执行历史记录

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x, // =
```

**request**

```
{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "taskId": 123,
}
```

## response

```
{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "resultCode": 1001, // 1001: 响应成功
  "resultMsg": "下发成功" // 底盘通信状态信息

  // 如果任务不存在需要返回相应信息
  "taskId": 123,
  "taskName": "任务名称",
  "startTime": 123, // 任务开始时间
  "finishTime": 123, // 任务结束时间
  "taskStatus": 0, // 任务执行状态 0: 执行中, 1: 正常结束, 2: 异常结束
  "stationStatus": [
    {
      "stationId": 122,
      "stationName": "站点 1",
      "arriveTime": 111 // 抵达站点时间
      "x": 1,
      "y": 2,
      "yaw": 3,
      "status": 0 // 站点执行状态 0: 执行中, 1: 正常结束, 2: 异常结束
    }
  ]
}
```

备注说明:

1. 如果任务有多个站点，前面站点执行异常导致后面站点无法执行，那么后面的站点都要标记为异常结束

## 11.遇障语音播报

数据端口: 5656

```
"type": 0x03,  
"cmdFunction": 0x0341, // =
```

### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "tasksId": 123,  
}
```

### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

## 12. 到达目标点通知上报

数据端口: 5656



```
"Data": {  
  "cmdTime": 1667823500544 // 时间戳，示例：1667739993  
  "resultCode": 1001, // 底盘通信状态码  
  "resultMsg": "获取导航状态成功" //  
}
```

## 13. 获取任务状态

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0353,
```

**request**

```
"type": 0x01,  
"cmdFunction": 0x0460, // =
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

**2023.7.17 17:30**

## 15.添加站点信息

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0302, // =
```

## request

```
{  
  "mapId" : 1671124918288,  
  "mapName" : "turtlebot3_world",  
  "stationName" : "gate1",  
  "stationType": 0, // 0, 导航站点; 1, 充电站点; 2, 电梯站点;  
                  // 3, 闸机站点; 4, 自动门站点; 5, 消防门站点;  
                  // 6, 待命点  
  "deviceId" : "FC1689322656488", // 导航点除外, 其他点位有该字段  
  "stationPos" : "outside" // 若为电梯点时, 有该字段.  
                           // ["outside": 电梯门外 ,  
                           //  "middle": 电梯门中间 ,  
                           //  "center": 轿厢中心,  
                           //  "inside": 轿厢里面,  
                           //  "wait": 等候点]  
}
```

## response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

**2023.10.13 14:37**

## 16.删除站点信息

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0304, // =
```

**request**

```
"type": 0x01,  
"cmdFunction": 0x0305, // =
```

#### **request**

```
{  
  "mapId" : 1671124918288,  
  "stationId" : 1671124918008,  
  "stationType" : 0,  
  "stationRename" : "gate2",  
}
```

#### **response**

```

{
  "mapId": 12345678, //请求加载地图的 id
  "mapName": "abc", //请求加载地图的名字
  "stationType": 0, //0,导航站点;1,充电站点;2,电梯站点;
                  //3,闸机站点;4,自动门站点;5,消防门站点,
                  //6,等待站点
  "cmdTime": 1667823500544 // 时间戳, 示例: 1667739993
}

```

## response

```

"Data": {
  "stations": [
    {
      "deviceId" : "1001330000000001",
      "stationId" : 1694140206787,
      "stationName" : "wait",
      "stationPos" : "wait",
      "mapId" : 1694140206787,
      "x" : "-0.81532",
      "y" : "-1.21390",
      "yaw" : "-0.07654"    },
    {
      "deviceId" : "1001330000000001",
      "stationId" : 1694140255073,
      "stationName" : "outside",
      "stationPos" : "outside",
      "x" : "-0.42947",
      "y" : "-0.25831",
      "yaw" : "-0.05305"    }
  ]
  "cmdTime": 1667823500544 // 时间戳, 示例: 1667739993
  "resultCode": 1001, // 底盘通信状态码
  "resultMsg": "获取点位列表成功" //
}

```

```
"type": 0x01,  
"cmdFunction": 0x0A07, // =
```

## request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": {  
    "iotDeviceType": "xxx" //“设备类型”取值范围：【elevator ,  
    autoDoor , gate , Charge , fireDoor】，该字段根据后续设备类型的增加而变  
    更。  
  }  
}
```

## response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": [  
    {  
      "address": "xxx",  
      "id": "001"  
      "comAddress": 0x14 //int  
    },  
    {  
      "address": "xxx",  
      "id": "002"  
      "comAddress": 0x14 //int  
    }  
    ...  
  ]  
}
```

**2023.08.01 新增<仅在测试（调试）APP 启用>**

## **20.设置目标地图**

数据端口: 5656



```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功 2002： 响应失败  
  "resultMsg": "设置目标地图成功" // 底盘通信状态信息 "设置目标地图失败"  
}
```

**2023.8.10 10:40**

## **21. 获取激光数据**

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0001,
```

**request**

```
"type": 0x01,  
"cmdFunction": 0x111,
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544,    // 时间戳, 示例: 1667739993  
  "mapId" : 1671124918288,    // 使用的目标地图 ID  
}
```

#### response

```
"Data": {  
  "image": "", // 名称  
  "resolution": 10.0, // 分辨率  
  "origin": {  
    "x": 1.0, // x 轴坐标, 单位: 米  
    "y": 1.0, // y 轴坐标, 单位: 米  
    "yaw": 1.0 // 角度, 单位: 弧度  
  },  
  "occupiedThresh": 10.0, //  
  "freeThresh": 10.0, //  
  "pngMapBtyes": "" // 源数据流, base64,  
  "cmdTime": 1669366407365, // 时间戳, 示例: 1667739993  
  "resultCode": 1001, // 底盘通信状态码  
  "resultMsg": "响应成功" // 底盘通信状态信息  
}
```

## 2023.8 ADP3.8 新增

```
"type": 0x01,  
"cmdFunction": 0x0A08, // =2568
```

### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": {  
    "update": "none" // none 读开关状态； on 设置开； off 设置关  
  }  
}
```

### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": {  
    "state": "on" // on 开; off 关  
  }  
}
```

## ~~24. 语音提醒的开关 0x0A09, // =2569~~

描述：

记录开关状态于机器人本体

数据端口: 5656

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": {  
    "state": "on" // on 开; off 关  
  }  
}
```

## 25. 切换业务环境 0x03A0, // =928

描述:

切换 release 环境（生产）和 alpha 环境（测试）

数据端口: 5656

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001： 响应成功  
  "resultMsg": "成功" // 状态信息  
}
```

## 26. 获取固件版本 0x0A0A, // =2570

### 描述：

（主控，开机运行时，通信获取各模块版本暂存于内存变量）填充 JSON 字段回复 socket。

三个模块的固件版本：

（上装）箱体固件版本

底盘固件版本

导航软件版本

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x03a1,
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "numFloors": 26, // 楼层的数量，注意标签的最大值才是总楼层数  
  "actFloors": ["-3", "-2", "-1", "A1", "1", ..., "22"], // 楼层的数量  
  "flgFloors": [1, 2, 3, 4, 5, ..., 26], // 楼层的数量  
  "resultMsg": "下发成功" // 底盘通信状态信息  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
}
```

## 2023.10 新增

### 28. 创建巡游/送物任务

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0350, // =
```

**request**



```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "taskName": "task_1", // 任务名称  
  "taskId": 1695733900004 , // 任务类型, 【1, 巡游; 2, 送物】  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
}
```

## 29. 执行任务

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0350, // =
```

**request**

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "taskName": "task_1", // 任务名称  
  "taskId": 1695733900004, // 任务类型, 【1, 巡游; 2, 送物】  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
}
```

## 2023.10 变更

### 30. 激光雷达点云显示

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0001,
```

### **request**

```
"Data": {  
  "cmdTime": 1667823500544,    // 时间戳，示例：1667739993  
  "mapId" : 1671124918288,    // 使用的目标地图 ID  
}
```

### **response**

## 2023.11.28 变更

### 31. 获取地图列表 0x0103, 获取地图信息 0x010c,

数据端口: 5656

```
"type":0x01,  
"cmdFunction": 0x0103,
```

Request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "msgId" : 90842323, // 指令 id  
}
```

Response

```

    "Data": {
        "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993
        "msgId" : 90842323, // 指令 id
        "resultCode": 1001, // 1001: 响应成功
        "resultMsg": "成功", // 状态信息
        "mapInfoList": [
            {
                "id": 1234567890,
                "name": "地图名称",
                "pngPath": "地图图片路径",
                "pgmPath": "",
                "yamlPath": "",
                "pbStreamPath": "",
                "configPath": "",
                "active": 1, // 是否是默认选择的地图
                "buildingId" : 1234567890, // long 2024.09.30 新增 楼栋
                id
                "floorId" : 1234567890, // long 2024.09.30 新增 楼层 id
                "tower": "A1 栋", // 楼栋 (不填默认 未知) - 新增
                "level": "B1", // 楼层 (不填默认 1) - 新增
                "mapVersion": 1234567890, // 地图版本号 - 新增
            }
        ]
    }
}

```

## 32. 保存地图 0x0109,

数据端口: 5656

```
"type":0x01,  
"cmdFunction": 0x0109,
```

## Request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "msgId" : 90842323, // 指令 id  
  "name": "地图名称",  
  "resolution": 2.10  
  "buildingId" : 1234567890, // long 2024.09.30 新增 楼栋 id  
  "floorId" : 1234567890, // long 2024.09.30 新增 楼层 id  
  "tower": "A1 栋", // 楼栋（不填默认 未知） - 新增  
  "level": "B1", // 楼层（不填默认 1 ） - 新增  
}
```

## Response

```
"type":0x01,  
"cmdFunction": 0x0320
```

## Request

```
"Data":{  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "msgId" : 90842323, // 指令 id  
  "mapList":[  
    {  
      "mapId":1234567890, //地图 id  
      "mapName":"地图名称",  
      "mapUrl":"https://cloud.tencent.com/xxx.zip" //地图存  
      储地址  
    }  
  ]  
}
```

## Response



```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "msgId" : 90842323, // 指令 id  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
}
```

## 38.上传地图

数据端口: 5656

```
"type":0x01,  
"cmdFunction": 0x0321
```

Request

```
"type":0x03,  
"cmdFunction": 0x0343
```

## Request

```
"Data":{  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "msgId" : 90842323, // 指令 id  
  "syncMapList":[  
    {  
      "mapName":"地图名称",  
      "mapId":1234567890 , //地图 id  
      "syncStatus": 1, //同步状态, 1: 上传, 0: 下载  
      "progress": 0, //同步进度 0-100, 间隔 5 上传一次数据:  
0,5,10,15,... 95,100  
      "result":0 //0 正在上传/等待上传, 1:成功, -1:失败  
    }  
  ]  
}
```

## Response

```
"Data": {  
  "cmdTime": 1667823500544 // 时间戳，示例：1667739993  
  "msgId" : 90842323, // 指令 id  
  "resultCode": 1001, // 底盘通信状态码  
  "resultMsg": "上报成功" //  
}
```

## 40. 获取正在同步的地图进度

数据端口：5656

```
"type": 0x01,  
"cmdFunction": 0x0322
```

Request

```
"type": 0x01,  
"cmdFunction": 0x0A0C, // =2572
```

## request

```
"Data": {  
    "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

## response

```
"Data": {  
    "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
    "resultCode": 1001, // 1001： 响应成功  
    "resultMsg": "成功" // 状态信息  
    "data": [  
        {  
            "code": 320001, // 异常码  
            "triggerTimeStampMs": 1701247598 , // 触发时间点时间戳  
            "description": "xxx", // 异常信息描述  
            "guidance": "xxx", // 解决指引  
            "name": "recharge_failed_after_retry", // 异常名称  
            "level": "FATAL", //异常等级。FATAL:致命故障，ERROR:普通  
异常，WARNING:警告  
            "source": "controller", // 异常来源  
            "cause": "xxx", // 可能原因  
        },  
        // ...  
    ]  
}
```

检查机器人本体各功能模块是否进入正常工作状态。

开机后会并行进行各项目检查，默认状态为 `status_unknown`，默认 2min 超时，最终状态为 `status_normal` 或 `status_abnormal`。

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A0D, // =2573
```

#### **request**

```
"Data": {  
    "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### **response**

```

>Data": {
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993
  "resultCode": 1001, // 1001: 响应成功
  "resultMsg": "成功" // 状态信息

  "data": [
    {
      "itemName": "node", // 自检项目可选值:
      // node: 节点状态检测
      // localization: 定位状态检测
      // controller: 主控状态检测
      // navigation: 导航状态检测
      // boot_at_station: 在充电桩启动

      "itemStatus": "xxx", // 自检项目状态可选值:
      // status_unknown: 未知
      // status_normal : 正常
      // status_abnormal: 异常
      "itemInfo": "xxx", // 详情可能为空, 异常时候有效
    },
    // ...
  ]
}

```

## 43. 获取机器人格口门配置 0x0A02, // =2562

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A02, // =2562
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

2023.11 更新修改

【企微文档】FC 配送机器人格口配置 [FC 配送机器人格口配置](#)

**格口物理位置** "cellLocation" 可选值:

左上角: leftTop

左下角: leftBottom

右上角: rightTop

右下角: rightBottom

上部: top

下部: bottom

左侧: left

右侧: right

全部: all



```

{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "resultCode": 1001, // 1001: 响应成功
  "resultMsg": "下发成功" // 底盘通信状态信息
  "data": {
    "cellNums": 4, // 格口数量
    "cellDoorList": [
      {
        "cellId": 1, // 格口编号
        "cellType": 1, // 1: 小格口 2: 中格口 3: 大格口
        "cellLockId": 1, // 箱体锁编号
        "cellDoorId": [1, 2] // 格口对应的门
      },
      {
        "cellId": 2,
        "cellType": 1, // 1: 小格口 2: 中格口 3: 大格口
        "cellLockId": 2,
        "cellDoorId": [3, 4]
      }
      ...
    ],
  }
}

```

## 44. 设置机器人格口配置 0x0A0B, // =2571

描述:

2023.11 更新修改

【企微文档】FC 配送机器人格口配置 [FC 配送机器人格口配置](#)

列举配置写在底盘本地的配置文件，

APP 接口设置当前生效配置

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A0B, // =2571
```

## request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "data": {  
    "activeCellConfigTypeIndex": 1, //1~n 更新  
    activeCellConfigTypeSeq 值  
  }  
}
```

## response

```
{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "resultCode": 1001, // 1001: 响应成功
  "resultMsg": "成功" // 底盘通信状态信息
}
```

分格形式:

类型序号 "cellConfigTypeIndex"	分格形式		本体存储的格口配置形式
1	leftTop	rightTop	{ "cellConfigTypeIndex": 1, "cellNums": 4, "cellDoorList": [ { "cellLocation": "leftTop", "cellId": 1, "cellType": 1, "cellDoorId": [ 1 ] }, { "cellLocation": "rightTop", "cellId": 2, "cellType": 1, "cellDoorId": [ 2 ] }, { "cellLocation": "leftBottom", "cellId": 3, "cellType": 1, "cellDoorId": [ 3 ] }, { "cellLocation": "rightBottom", "cellId": 4, "cellType": 1, "cellDoorId": [ 4 ] } ] }
	leftBottom	rightBottom	

	<div>bottom</div>		<pre>{   "cellLocation": "bottom",   "cellId": 2,   "cellType": 2,   "cellDoorId": [     3,     4   ] }</pre>
3	<div>left</div>	<div>right</div>	<pre>{   "cellConfigTypeIndex": 3,   "cellNums": 2,   "cellDoorList": [     {       "cellLocation": "left",       "cellId": 1,       "cellType": 2,       "cellDoorId": [         1,         3       ]     },     {       "cellLocation": "right",       "cellId": 2,       "cellType": 2,       "cellDoorId": [         2,         4       ]     }   ] }</pre>
4	<div>all</div>		<pre>{   "cellConfigTypeIndex": 4, // 类型序号   "cellNums": 1, // 格口数量   "cellDoorList": [     {       "cellLocation": "all", // 格口物理位置       "cellId": 1, // 格口丝印编号       "cellType": 3, // 1: 小格口 2: 中格口 3: 大格口       "cellDoorId": [         1,</pre>

			<div>2, 3, 4 ]// 格口对应的门电机 } ] }</div>

(旧版本)

APP 下发格口门配置，覆盖写到本体配置文件中。

设计共有 4 种格口类型。

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0A0B, // =2571
```

## request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": { // 可选值见下表  
    "cellNums": 1, // 格口数量  
    "cellDoorList": [  
      {  
        "cellId": 1, // 格口编号  
        "cellType": 3, // 1: 小格口 2: 中格口 3: 大格口  
        "cellDoorId": [  
          1,  
          2,  
          3,  
          4  
        ] // 格口对应的门  
      }  
    ]  
  }  
}
```

## response

```
{
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993
  "resultCode": 1001, // 1001: 响应成功
  "resultMsg": "成功" // 底盘通信状态信息
}
```

分格配置：

4	1 分格	<pre>{   "cellNums": 1, // 格口数量   "cellDoorList": [     {       "cellId": 1, // 格口编号       "cellType": 3, // 1: 小格口 2: 中格口 3: 大格口       "cellDoorId": [         1,         2,         3,         4       ] // 格口对应的门     }   ] }</pre>
2	2 分格(左右)	<pre>{   "cellNums": 2,   "cellDoorList": [     {       "cellId": 1,       "cellType": 2,       "cellDoorId": [         1,         2       ]     }   ] }</pre>

		<pre>"cellDoorList": [   {     "cellId": 1,     "cellType": 2,     "cellDoorId": [       1,       3     ]   },   {     "cellId": 2,     "cellType": 2,     "cellDoorId": [       2,       4     ]   } ]</pre>
1	4 分格	<pre>{   "cellNums": 4,   "cellDoorList": [     {       "cellId": 1,       "cellType": 1,       "cellDoorId": [         1       ]     },     {       "cellId": 2,       "cellType": 1,       "cellDoorId": [         2       ]     }   ] }</pre>



		<pre>}, {   "cellId": 3,   "cellType": 1,   "cellDoorId": [     3   ] }, {   "cellId": 4,   "cellType": 1,   "cellDoorId": [     4   ] } ]</pre>
--	--	--

**2024.01.02 10:22**

**45. 设置机器人待命时的位置**

描述：

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0311, // =
```

## request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": [  
    {  
      "mapId": 1667823500544, // 点位所在的地图 id  
      "stationId": 1667823500544, // 点位的 id  
      "stationType": 6 // 点位的类型：可选【1-充电点 6-待命点】  
    }  
  ] // 使用数组，以防后续会拓展为设置多个待命点的情况。  
}
```

## response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
}
```

## 46. 获取机器人待命时的位置

描述:

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0312, // =
```

**request**

```
"type": 0x01,  
"cmdFunction": 0x0313, // =
```

## request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": [  
    {  
      "mapId": 1667823500544, // 点位所在的地图 id  
      "stationId": 1667823500544, // 点位的  
      id  
    }  
  ]  
}
```

## response

```
"type": 0x01,  
"cmdFunction": 0x0314, // =
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": [  
    {  
      "mapId": 1667823500544, // 点位所在的地图 id  
      "stationId": 1667823500544, // 点位的  
id      }  
    ]  
  }  
}
```

**2024.01.29 16:08**

## 49. 上报机器人任务进度

描述:

数据端口: 5656

```
"type": 0x03,  
"cmdFunction": 0x0345, // =
```

#### request

```
"Data": {  
  "process": 1  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

备注：“process”的取值说明

- 0. 非特殊进度（一般进度）
- 1. 过闸机
- 2. 过自动门
- 3. 过消防门
- 4. 乘坐电梯中
- 5. 呼叫电梯中
- 6. 进入电梯中
- 7. 走出电梯中

#### response

```
"type": 0x01,  
"cmdFunction": 0x0A01, // =2561
```

#### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

#### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "下发成功" // 底盘通信状态信息  
  "data": {  
    "robotSn": "100132001234567" // 15 位资产编码，后 7 位流水号，  
    根据供应商区分  
    "robotModel": "Indoor" //  
  }  
}
```

## 51. 查询上装接口 0x0204

#### 描述：

原接口，未修改。控制格口前会先查询上装。

信息同步如下：

1. 上位机根据下位机返回状态来控制格口门；
2. 状态中保留使用了两个字段，来源于上装寄存器 0x207，代表推杆电机和箱锁电机（二代机

```
"type": 0x01 ,  
"cmdFunction": 0x204, // = 516
```

### **request**

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  
}
```

### **response**



```
"Data": {

    "boxPushStatus" : 7, //推杆状态,7 收回到原始位置,4: 推出正转到位
    "selectCellId":0 ,   //被释放的箱体编号,0 表示复位,没有释放任何箱体
    "cmdTime" : 1706498076699,
    "doorLockInfoList" :
    [
        {
            "lockId" : 1,      //门锁 id
            "lockStatus" : 7   //门锁状态, 4:打开到位, 7: 关闭到位
        },
    ],

    "resultCode" : 1001,
    "resultMsg" : "获取上装状态成功",
    "selectCellId" : 0

}
```

## 52.获取机器人所在站点

描述: 查询当前机器人所在站点

数据端口:5656

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳, 示例: 1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": {  
    "mapId": 123456666, // 地图 id  
    "stationId": 3333333, // 站点 id  
    "stationName": "21 楼-导航点 3", // 站点名称  
    "stationType": 0 // 见接口 18 《获取站点列表》, 同名称字段  
  }  
}
```

**注:** 当前机器人不再任何一个站点, 获取 **data** 数据为空

## 53. 主动上报机器人抵达站点

数据端口: 5656

```
"Data": {  
  "cmdTime": 1667823500544 // 时间戳，示例：1667739993  
  "msgId" : 90842323, // 指令 id  
  "resultCode": 1001, // 底盘通信状态码  
  "resultMsg": "成功" //  
}
```

## 54. 上传竞争资源到调度系统

描述:

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x0333, // =
```

### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
}
```

### response

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "resultCode": 1001, // 1001: 响应成功  
  "resultMsg": "成功" // 状态信息  
  "data": {  
    "condision": "true" // true-条件满足 false-条件不满足  
  }  
}
```

## 56. template

描述:

数据端口: 5656

```
"type": 0x01,  
"cmdFunction": 0x, // =
```

### request

```
"Data": {  
  "cmdTime": 1667823500544, // 时间戳，示例：1667739993  
  "data": {  
    "xxx": "xxx" //  
  }  
}
```

### response