# 自动叉车接口文档（AGV）

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改描述 | 作者 |
| 2017-10-16 | V1.0 | 文档撰写 |  |
| 2018-07-16 | V1.1 | 文档撰写 |  |

目录

[自动叉车接口文档（AGV） 1](#_Toc522115953)

[修订记录 1](#_Toc522115954)

[目录 2](#_Toc522115955)

[**1.** 3](#_Toc522115956)

[**1.1.** **编写目的** 3](#_Toc522115957)

[**1.2.** **术语定义** 3](#_Toc522115958)

[**2.** **接口概要** 3](#_Toc522115959)

[**2.1.** **通信方式** 3](#_Toc522115960)

[**2.2.** **报文格式** 3](#_Toc522115961)

[**2.3.** **报文序号** 3](#_Toc522115962)

[**2.4.** **ACK机制** 3](#_Toc522115963)

[**3.** **指令类型** 4](#_Toc522115964)

[**3.1.** **模式设置（WMS->AGV）** 4](#_Toc522115965)

[**3.2.** **获取货位坐标（WMS->AGV）** 5](#_Toc522115966)

[**3.3.** **获取站点坐标（WMS->AGV）** 5](#_Toc522115967)

[**3.4.** **移位指令1（LL，WMS->AGV）** 6](#_Toc522115968)

[**3.5.** **移位指令2（LL，WMS->AGV） 不用** 7](#_Toc522115969)

[**3.6.** **拣选指令（LS，WMS->AGV）** 8](#_Toc522115970)

[**3.7.** **回库指令（SL，WMS->AGV）** 9](#_Toc522115971)

[**3.8.** **移动指令（SS，WMS->AGV）** 10](#_Toc522115972)

[**3.9.** **释放指令（release，WMS->AGV）** 11](#_Toc522115973)

[**3.10.** **获取机器当前执行任务指令（WMS->AGV）** 11](#_Toc522115974)

[**3.11.** **解绑机器任务指令（WMS->AGV）** 12](#_Toc522115975)

[**3.12.** **转移机器任务到其他指定机器（WMS->AGV）** 13](#_Toc522115976)

[**3.13.** **Agv任务删除事件（WMS->AGV）** 13](#_Toc522115977)

[**3.14.** **获取异常指令（WMS->AGV）** 14](#_Toc522115978)

[**3.15.** **取消指令（cancel，WMS->AGV） 禁用** 14](#_Toc522115979)

[**3.16.** **强制取消指令（force\_cancel\_mission，WMS->AGV）禁用** 15](#_Toc522115980)

[**3.17.** **删除机器所有任务指令（delete\_robot\_mission，WMS->AGV）** 15](#_Toc522115981)

[**3.18.** **指令任务状态（status，AGV->WMS）** 15](#_Toc522115982)

[**3.19.** **任务异常状态（exception，AGV->WMS）** 16](#_Toc522115983)

[**3.20.** **负载(取空)异常状态（loadexception，AGV->WMS）** 16](#_Toc522115984)

[**3.21.** **满入异常状态（onloadexception，AGV->WMS）** 16](#_Toc522115985)

[**3.22.** **启用机器（WMS->AGV）** 17](#_Toc522115986)

[**3.23.** **禁用机器（WMS->AGV）** 17](#_Toc522115987)

[**3.24.** **删除机器（WMS->AGV）** 17](#_Toc522115988)

[**3.25.** **开始启动所有机器（WMS->AGV）** 17](#_Toc522115989)

[**3.26.** **暂停所有机器（WMS->AGV）** 18](#_Toc522115990)

[**3.27.** **获取地图数据（WMS->AGV）（内部使用）** 18](#_Toc522115991)

[**3.28.** **机器实时数据状态（AGV->WMS）** 19](#_Toc522115992)

[**3.29.** **其他错误（AGV->WMS）** 20](#_Toc522115993)

[**4.** **业务举例** 21](#_Toc522115994)

[**4.1.** **移位LL指令** 21](#_Toc522115995)

[**4.2.** **LS,SS,SL配套使用** 23](#_Toc522115996)

[**4.3.** **LS,SL配套使用** 27](#_Toc522115997)

1. 1. **编写目的**

本文档描述第三方WMS系统与自动叉车AGV系统之间的接口。

* 1. **术语定义**

|  |  |
| --- | --- |
| **名称** | **描述** |
| WMS | 仓库管理系统（Warehouse Management System） |
| AGV | 自动叉车（Automated Guided Vehicle）调度系统 |

1. **接口概要**
   1. **通信方式**

Websocket通讯方式，AGV为服务端，WMS作为客户端主动连接AGV。端口使用33333，URL地址"ws://"+ip+":33333/ws/connect"，AGV服务端只支持一个连接，除了机器实时数据以外，所有报文指令都是使用这个URL地址，机器实时数据使用的URL地址是：

"ws://"+ip+":33333/ws/robotinfo"

* 1. **报文格式**

接口内容采用json字符串格式，每条报文都含有“cmdcode”和“cmdid”，其中“cmdcode”表示报文(指令)类型，值类型是字符串，“cmdid”表示报文序号，值类型是整型

{"cmdcode":”ack”, "cmdid": 1 ｝

* 1. **报文序号**

WMS和AGV各自单独维护自己的报文序号。每次发送报文给对方时，报文序号在前面报文序号基础上加1，报文序号范围1~999999，当序号达到999999时，报文序号重新从1开始。

{"cmdcode":”ack”, "cmdid": 1~999999 ｝

* 1. **ACK机制**

接收方在成功接收一条报文后，立即反馈一条ACK消息给发送方。发送方发送一条报文后，必须收到该报文的ACK后才可以发送下一条报文。发送方发送报文后，3秒（可调整）后还未收到ACK，需要重新发送该报文，重复多次还未收到ACK的情况下，发送方应该断开Socket连接,尝试重新连接。

{"cmdcode":”ack”, "cmdid":原报文序号｝

1. **指令类型**
   1. **模式设置（WMS->AGV）**

AGV服务有两个模式，一个是测试模式，一个是应用模式；测试模式是AGV服

务为了方便小车在实际场地能不依赖其他WMS系统也可以直接跑动起来，来验证场地是否满足实际应用需求； 应用模式是与WMS系统对接

1.获取AGV服务模式命令：

{

"cmdcode":"getmode",

"cmdid": 1,//WMS的报文序号

}

返回数据

{

"cmdcode":" getmode ",

"cmdid": 1,//agv的报文序号

"mode": "0" //0表示测试模式,1表示应用模式

}

2.设置AGV服务模式命令：

{

"cmdcode":"setmode",

"cmdid": 1,//WMS的报文序号

"mode": "0" //0表示测试模式,1表示应用模式

}

返回数据

{

"cmdcode":" setmode ",

"cmdid": 1,//agv的报文序号

"mode": "0" //0表示测试模式,1表示应用模式

}

* 1. **获取货位坐标（WMS->AGV）**

主要是为了实现WMS系统的库位绑定货位坐标，通过这个命令可以获取货位坐标，返回当前地图的货架坐标x,y和货架层数z,然后通过x，y，z组成LL，LS，SL指令中Location的货位坐标，具体命令如下：

{

"cmdcode":" getlocations", //字符串，

"cmdid":1, //WMS的报文序号

｝

返回数据

{

"cmdcode":" getlocations ", //字符串，

"cmdid":1, //AGV的报文序号

"locations":[{"x":17,"y":5,"z":1},{"x":17,"y":5,"z":2}…] 数组 z表示货架层数，默认最大是4层货架

｝

* 1. **获取站点坐标（WMS->AGV）**

通过这个命令可以获取Station站点的任务坐标，返回当前地图的Station站点坐标x,y，然后通过x，y 组成LS，SL，SS指令中Station站点的坐标，这些Station站点坐标需要根据实际场景进行配置，具体命令如下：

{

"cmdcode":" getstations", //字符串，

"cmdid":1, //WMS的报文序号

｝

返回数据

{

"cmdcode":"getstations",

"cmdid":1,

"stations":[{"x":45,"y":14,"station\_no":"B101"},{"x":45,"y":15,"station\_no":"B102"}...]数组

}

station\_no ：站点编号，在设计地图时候确定下来,后期添加和修改，需要我们配合去更改

* 1. **移位指令1（LL，WMS->AGV）**

Location to Locaton，货位到货位，小车从一个货位拿货到另外一个货位放货，WMS发送指令任务给AGV, AGV自动反馈任务执行状态给WMS，具体请看**任务状态指令**

LL指令格式：

{

"cmdcode":"LL",

"cmdid": 1,//WMS的报文序号

"missiongroups" :[

{"startx":，整型，起点X

"starty":，整型，起点Y

"startz": 整型 起点Z

"endx":, 整型，终点X

"endy":, 整型，终点Y

"endz":, 整型，终点Z，

"robotid":, 整型，机器人编号

"missiongroupid":"" ，字符串 任务组编号 自定义，不能重复

}

]

}

**备注**：

"cmdcode"：字符串，指令类型，

"cmdid": 整型，报文序号，

"missiongroups"：任务组数组，可以批量发送任务，

"startx":整型，起点X，

"starty":，整型，起点Y，

"startz": 整型 起点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"endx":, 整型，终点X，

"endy":, 整型，终点Y，

"endz":, 整型，终点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"missiongroupid":"" ，字符串 任务组编号 自定义，不能重复

"robotid":, 整型，机器人编号，默认为0，忽略 也可以，如果需要指定机器去执行任务，则为对应机器编号

* 1. **移位指令2（LL，WMS->AGV） 不用**

//这个指令跟上面的功能是一样的，个别项目用到

{

"1":{

"cmdcode":"LL",

"cmdid": 7,

"startx": 5,

"starty":16,

"startz":2,

"endx":5,

"endy":36,

"endz":2,

"missiongroupid":"3",

"robotid":2001,

},

"2":{

"cmdcode":"LL",

"cmdid": 8,

"startx": 5,

"starty":19,

"startz":2,

"endx":5,

"endy":38,

"endz":2,

"missiongroupid":"2"

},

"3":{

"cmdcode":"LL",

"cmdid": 9,

"startx": 5,

"starty":23,

"startz":2,

"endx":5,

"endy":40,

"endz":2,

"missiongroupid":"3"

}

}

**备注**：

"cmdcode"：字符串，指令类型，

"cmdid": 整型，报文序号，

"missiongroups"：任务组数组，可以批量发送任务，

"startx":整型，起点X，

"starty":，整型，起点Y，

"startz": 整型 起点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"endx":, 整型，终点X，

"endy":, 整型，终点Y，

"endz":, 整型，终点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"missiongroupid":"" ，字符串 任务组编号 自定义，不能重复

"robotid":, 整型，机器人编号，默认为0，如果需要指定机器去执行任务，则为对应机器编号

* 1. **拣选指令（LS，WMS->AGV）**

Locaton to Station，货位到站点，小车从货位拿货到站点等待，人工拣货完成后需要发送指令让小车把货物放到货位上，所以跟SS,SL指令配套使用，不能单独使用，他们的missiongroupid 必须是一样的。WMS发送指令任务给AGV, AGV自动反馈任务执行状态给WMS，具体请看**任务状态指令**

LS指令格式：

{

"cmdcode":"LS",

"cmdid": 1,//WMS的报文序号

"missiongroups" :[

{"startx":，整型，起点X

"starty":，整型，起点Y

"startz": 整型 起点Z

"endx":, 整型，终点X

"endy":, 整型，终点Y

"robotid":, 整型，机器人编号

"missiongroupid":"" ，字符串 任务组编号

}

]

}

**备注**：

"cmdcode"：字符串，指令类型，

"cmdid": 整型，报文序号，

"missiongroups"：任务组数组，可以批量发送任务，

"startx":整型，起点X，

"starty":，整型，起点Y，

"startz": 整型 起点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"endx":, 整型，终点X，

"endy":, 整型，终点Y，

"missiongroupid":"" ，字符串 任务组编号 自定义，跟SL指令一样

"robotid":, 整型，机器人编号，默认为0，如果需要指定机器去执行任务，则为对应机器编号

* 1. **回库指令（SL，WMS->AGV）**

SL，Station to Location，站点到货位，LS拣选或者SS指令任务完成后，需要发送指令让小车把货物放到货位上，不能单独使用这个指令。WMS发送指令任务给AGV, AGV自动反馈任务执行状态给WMS，具体请看**任务状态指令**

SL指令格式：

{

"cmdcode":"SL",

"cmdid": 1,//WMS的报文序号

"missiongroups" :[

{"startx":，整型，起点X

"starty":，整型，起点Y

"endx":, 整型，终点X

"endy":, 整型，终点Y

"endz": 整型 终点Z

"robotid":, 整型，机器人编号

"missiongroupid":"" ，字符串 任务组编号 自定义

}

]

}

**备注**：

"cmdcode"：字符串，指令类型，

"cmdid": 整型，报文序号，

"missiongroups"：任务组数组，可以批量发送任务，

"startx":整型，起点X，

"starty":，整型，起点Y，

"endx":, 整型，终点X，

"endy":, 整型，终点Y，

"endz":, 整型，终点Z，值范围在1~4之间时候表示货架层数，大于30，表示叉车升降杆与地面的高度，单位毫米，其他值表示错误值

"missiongroupid":"" ，字符串 任务组编号 自定义，跟LS指令一样

"robotid": 整型，机器人编号，跟LS或者SS配套使用，LS或者SS已经分配了机器，需要指定机器去执行任务

* 1. **移动指令（SS，WMS->AGV）**

SS，Station to Station，即站点到站点，小车从一个站点移动到另外一个站点等待，如果叉杠有托盘货物，则跟LS，SL指令配套使用，missiongroupid一样

SS指令格式：

{

"cmdcode":"SS",

"cmdid": 1,//WMS的报文序号

"missiongroups" :[

{"startx":，整型，起点X

"starty":，整型，起点Y

"endx":, 整型，终点X

"endy":, 整型，终点Y

"robotid":, 整型，机器人编号

"missiongroupid":"" ，字符串 任务组编号 自定义

}

]

}

**备注**：

"cmdcode"：字符串，指令类型，

"cmdid": 整型，报文序号，

"missiongroups"：任务组数组，可以批量发送任务，

"startx":整型，起点X，

"starty":，整型，起点Y，

"endx":, 整型，终点X，

"endy":, 整型，终点Y，

"missiongroupid":"" ，字符串 任务组编号 自定义，跟LS指令一样

"robotid":, 整型，机器人编号，单独使用默认为0，如果是跟LS配套使用，LS已经分配了机器，需要指定机器去执行任务，则为对应机器编号

* 1. **释放指令（release，WMS->AGV）**

释放那些需要等待的指令任务，比如LS指令和SS指令

{

"cmdcode":"release ", //字符串，

"cmdid":1，//WMS的报文序号

"missiongroupid":"1", 任务组编号

｝

备注：比如LS指令，小车把货物拿到站点位置后，会一直在等待，等待处理托盘的货物，处理完成之后需要告诉小车已经处理完成了，如果人工直接把托盘都拿走的话，就需要释放小车，让它可以接收其他任务，如果还需要把货物放回货位的话，就需要SL指令了

* 1. **获取机器当前执行任务指令（WMS->AGV）**

{

"cmdcode":"get\_robot\_curmission ", //字符串，

"cmdid":1，//WMS的报文序号

"robotid":, 整型，机器人编号

｝

返回 数据

{

"cmdcode":"get\_robot\_curmission ", //字符串，

"cmdid":1，//AGV的报文序号

"robotid":, 整型，机器人编号

"missiongroupid":1，//字符串 任务组编号

"mtype":1，//任务类型 0，移动 1，叉货 2，放货 其他表示充电或者休息

"x":1，//整型 任务 x坐标

"y":1，//整型任务y坐标

"z":1，//整型 任务z坐标

"batterypower":，//整型 电量

"posx":，//整型x坐标

"posy":，//整型 y坐标

" poscode":，//整型 二维码

" errorcode":，//整型//机器 错误码

" enable":，//整型 //机器 启用码 0,1 是禁用 2,启用

" warncode":，//整型 //机器 警告码

"cargo\_status":，//整型 // 0没有托盘，1 有托盘

"missionid":，//整型//机器任务码

"system\_pause":，//bool // 机器运行 状态 true 运行 ，false 暂停

" exception":，//整型 流程异常 //0 正常，1取空异常 2.满入异常 3.其他异常 4. 表示识别坐标不对,5,任务异常 6.请拿走托盘

｝

* 1. **解绑机器任务指令（WMS->AGV）**

解绑机器任务指令任务

{

"cmdcode":"unbind\_robot\_mission ", //字符串，

"cmdid":1，//WMS的报文序号

"robotid":, 整型，机器人编号

｝

备注：小车执行了某个任务，出现故障，无法重新开机继续执行，需要把该机器跟任务解绑，由其他小车去执行 ，有箱子的，需要把箱子放回原来的位置

* 1. **转移机器任务到其他指定机器（WMS->AGV）**

{

"cmdcode":" move\_robot\_mission ", //字符串，

"cmdid":1，//WMS的报文序号

"src\_robotid":, 整型，原机器人编号

"dest\_robotid":, 整型，目标机器人编号

｝

备注：小车执行了某个任务，出现故障，无法重新开机继续执行，小车叉杠上又有箱子，在不知道箱子原来的位置 的情况下，可以把箱子放到一部新的小车，然后转移任务

返回数据

{

"cmdcode":" move\_robot\_mission ", //字符串，

"cmdid":1，//WMS的报文序号

"src\_robotid":, 整型，原机器人编号

"dest\_robotid":, 整型，目标机器人编号

"status":, 整型 0 成功 ，1 原机器编号不存在，2，目标机器编号不存在，3，目标机器存在其他任务，不能转移

｝

* 1. **Agv任务删除事件（WMS->AGV）**

Agv自动上传删除的任务

{

"cmdcode":" agv\_del\_mission", //字符串，

"cmdid":1，//agv的报文序号

"missiongroupid":1，//任务组编号

｝

备注：小车去放货时候，货位上已经存在货物， 造成满入异常，重启后会继续执行，如果人为拿走小车的货物，则删除这次任务

* 1. **获取异常指令（WMS->AGV）**

获取异常指令

{

"cmdcode":"getexception", //字符串，

"cmdid":1，//WMS的报文序号

｝

返回数据

{

"cmdcode":"getexception", //字符串，

"cmdid":1，//agv的报文序号

"robotid":, //整型，

" errorcode ":整型" 错误码

｝

* 1. **取消指令（cancel，WMS->AGV） 禁用**

取消任务

{

"cmdcode":"cancel ", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

｝

备注：当你取消LL和LS指令的任务时候，在机器已经拿到托盘了情况下，是取消不了的，**指令任务状态的status =1时候需要另外处理**

* 1. **强制取消指令（force\_cancel\_mission，WMS->AGV）禁用**

必须在确保执行该任务的机器关机状态下使用，一般用于手动操作

{

"cmdcode":" force\_cancel\_mission", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

｝

* 1. **删除机器所有任务指令（delete\_robot\_mission，WMS->AGV）**

必须在机器关机状态才能使用，一般用于手动操作

{

"cmdcode":" **delete\_robot\_mission** ", //字符串，

"cmdid":1, //WMS的报文序号

"robotids":[2001,2002] // 机器id数组

｝

* 1. **指令任务状态（status，AGV->WMS）**

任务状态由agv自动上传

{

"cmdcode":"status", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

"status":0, // 0开始执行, 1 表示第一动作完成 ,2.表示第二动作完成

"robotid":2001 //执行任务的机器

｝

备注：LL，LS，SL，SS都是拆分为两个动作，第一动作对应的是起点坐标，第二动作对应的是终点坐标，比如LL：status=0 表示开始执行，status=1 表示在起点坐标拿货完成，status=2表示在终点坐标放货完成； SS：status=0 表示开始执行，status=1 表示小车移动到起点坐标完成，status=2表示在小车移动到终点位置完成

* 1. **任务异常状态（exception，AGV->WMS）**

任务异常状态主要是执行任务时候，机器出现异常，由AGV自动上传

{

"cmdcode":"exception", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

"robotid":2001 //执行任务的机器

"errorcode ":1, //机器的错误码

｝

**备注**：当任务在执行中出错的时候，如果叉杠上没有托盘，该任务会自动解绑机器，重新分配给其他机器执行；如果叉杠上有托盘，该任务需要等待机器重启，机器重启之前没有人为干预，托盘还在叉杠上的话，会继续执行上次出错的任务，如果人为干预了，在重启机器之前把托盘拿掉，则系统会自动删除出错任务

* 1. **负载(取空)异常状态（loadexception，AGV->WMS）**

任务异常状态主要是执行LL和LS任务时候，机器取货完成的时候，发现机器没有取到托盘，在应用模式下，AGV系统会暂停所有机器，以便人工处理，由AGV自动上传

{

"cmdcode":"loadexception", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

"robotid":2001 //执行任务的机器

｝

* 1. **满入异常状态（onloadexception，AGV->WMS）**

任务异常状态主要是执行LL和SL任务时候，放货时撞到该货位的托盘

{

"cmdcode":"onloadexception", //字符串，

"cmdid":1，//AGV的报文序号

"missiongroupid":"1", 任务组编号

"robotid":2001 //执行任务的机器

｝

* 1. **启用机器（WMS->AGV）**

{

"cmdcode":" enable ", //字符串，

"cmdid":1, //WMS的报文序号

"robotids":[2001,2002] // 机器id数组

｝

* 1. **禁用机器（WMS->AGV）**

{

"cmdcode":" disable ", //字符串，

"cmdid":1, //WMS的报文序号

"robotids":[2001,2002] // 机器id数组

｝

* 1. **删除机器（WMS->AGV）**

{

"cmdcode":" delete ", //字符串，

"cmdid":1, //WMS的报文序号

"robotids":[2001,2002] // 机器id数组

｝

* 1. **开始启动所有机器（WMS->AGV）**

{

"cmdcode":" allstart ", //字符串，

"cmdid":1, //WMS的报文序号

｝

如果命令成功的话，在每个机器实时数据状态的system\_pause键值设置成true

* 1. **暂停所有机器（WMS->AGV）**

{

"cmdcode":" allpause", //字符串，

"cmdid":1, //WMS的报文序号

｝

如果命令成功的话，在每个机器实时数据状态的system\_pause设置成false

* 1. **获取地图数据（WMS->AGV）（内部使用）**

{

"cmdcode":" getmapdata", //字符串，

"cmdid":1, //WMS的报文序号

｝

返回数据

{

"cmdcode":" getmapdata", //字符串，

"cmdid":1, //AGV的报文序号

"mapdata":, //地图数据,string ,json文本

｝

* 1. **机器实时数据状态（AGV->WMS）**

AGV自动上传机器状态，如果某台机器出现异常情况，全部机器都会暂停

例如：

{"cmdcode":"robotinfo","cmdid":0,"robotarray":[

{"robotid":2003,"status":0,"batteryPower":0,"posX":0,"posY":0,"errorcode ":0,"enable":0,"warncode":255,"system\_pause":false},

{"robotid":2004,"status":0,"batteryPower":0,"posX":0,"posY":0,"errorcode ":0,"enable":0,"warncode":255,"system\_pause":false}

]}

由于数据传输比较频繁，这条指令不需要ack，cmdid设置为0，

robotarray 机器数组，" robotid ":整型," status":整型,"

"batterypower ":整型,"x":整型,"y":整型," errorcode ":整型," warncode":整型,"enable ":整型，system\_pause：(布尔类型, true:运行 false:暂停)

status状态：闲置0; 忙碌1; 错误3; 充电4

batterypower电量：电量的百分比

enable使能：刚开机时候是0表示禁用，手动禁用会变成1，2表示启用

errorcode错误码：

0 直行出线

1 停车过线

2 气压碰撞

3 叉杆撞托板

4 叉杆撞障碍物

5 小车顶货架

6. 红外避障

7 wifi通讯异常

8 SPI校验错误

9 SPI通讯超时

10 角度传感器超时

11 前摄像头超时

12 中摄像头超时

13 后摄像头超时

14 升降杆堵转

15 升降杆超界

16 控制器超时

17 电池电压低

18 充电器连接失败

19 电池充电电压异常

20 轮毂堵转

21 轮毂速度异常

22 升降杆保险管异常

23电源开关断开

24 紧急开关闭合

25 命令错误

**255 正常**

warncode警告码：

0: "前摄像头异常"

1: "中摄像头异常"

2: "后摄像头异常"

4: "电量低"

5: wifi 故障

6:表示该机器已经断线超过三分钟以上

**255: "正常"**

* 1. **其他错误（AGV->WMS）**

当WMS向AGV发送指令时候，AGV会辨别这个指令是否正确,然后把错误信息上传到WMS

{

"cmdcode":"error", //字符串，命令码"cmdid":10，//AGV的报文序号

"errortype": 1//1 命令格式错误，2 不存在这个命令， 3错误任务

"missiongroupid":"", 任务组编号

｝

**备注：**missiongroupid 只有在errortype 为3错误任务时候才使用

例如 ：

WMS->AGV：

{

"cmdcode":"LL",

"cmdid": 10,

"missiongroups" :[

{"startx":5,"starty":6,"startz":1,"endx":50,"endy":26,"endz":4,"robotid":0, "missiongroupid":"1" },

{"startx":5,"starty":28,"startz":2,"endx":24,"endy":26,"endz":380,"robotid":2002, "missiongroupid":"2" }

]

}

//AGV收到指令，ACK应答

AGV->WMS：{"cmdcode":"ack","cmdid":10}

//如果AGV发现missiongroupid 是”2”的任务不对，就会反馈

AGV->WMS：

{"cmdcode":"error","cmdid":1~999999, "errortype": 3,"missiongroupid":"2"}

//如果AGV发现这条指令格式解析不了，就会反馈

AGV->WMS：

{"cmdcode":"error","cmdid":1~999999, "errortype": 1,"missiongroupid":""}

//如果AGV发现这条指令cmdcode不存在，就会反馈

AGV->WMS：

{"cmdcode":"error","cmdid":1~999999, "errortype":2, "missiongroupid":""}

WMS->AGV: {"cmdcode":"ack","cmdid":1~999999}

1. **业务举例**
   1. **移位LL指令**

例如：Location

WMS需要把从原Location货位x=5,y=6,z=1的货物移动到新Location货位x=50,y=26,z=4，和原Location货位x=5,y=28,z=2的货物移动到新Location货位x=24,y=26,z=2，通信过程如下。

//WMS 发送指令

WMS->AGV：

{

"cmdcode":"LL",

"cmdid": 1,

"missiongroups" :[

{"startx":5,"starty":6,"startz":1,"endx":50,"endy":26,"endz":4,"robotid":0, "missiongroupid":"1" },

{"startx":5,"starty":28,"startz":2,"endx":24,"endy":26,"endz":2,"robotid":2002, "missiongroupid":"2" }

]

}

//AGV收到指令，ACK应答

AGV->WMS：

{"cmdcode":"ack","cmdid":1}

//第一组任务

//AGV随机分配任务给了2001，上传任务开始执行状态

AGV->WMS：

{"cmdcode":"status","cmdid":1，

"missiongroupid":"1","status":0, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":1}

//AGV上传拿货完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":2，

"missiongroupid":"1","status":1, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":2}

//AGV上传放货完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":3，

"missiongroupid":"1","status":2, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":3}

//第二组任务

//指定2002机器，AGV分配任务给了2002，上传任务开始执行状态

AGV->WMS：

{"cmdcode":"status","cmdid":4，

"missiongroupid":"2","status":0, "robotid":2002 }

//WMS收到指令，ACK应答

WMS->AGV：{"cmdcode":"ack","cmdid":4}

//AGV上传拿货完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":5，

"missiongroupid":"2","status":1, "robotid":2002 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":5}

//AGV上传放货完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":6，

"missiongroupid":"2","status":2, "robotid":2002 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":6}

* 1. **LS,SS,SL配套使用**

假设Location货位1坐标是x=10，y= 5，z=3，Station站点1的坐标是x=15，y=20，Station站点2的坐标x=16，y=22；

假如流程是：货位1----- >站点1----->站点2 -----> 货位1(或者其他货位)，一套任务中missiongroupid 必须是一样的，如果第一个任务已经分配了机器，后面指令必须指定机器

1. 从货位1 取货 到站点1 等待

WMS->AGV：

{

"cmdcode":"LS",

"cmdid": 1,

"missiongroups" :[

{"startx":10,"starty":5,"startz":3,"endx":15,"endy":20，"missiongroupid":"100" }

]

}

//AGV收到指令，ACK应答

AGV->WMS：

{"cmdcode":"ack","cmdid":1}

//AGV随机分配任务给了2001，上传任务开始执行状态

AGV->WMS：

{"cmdcode":"status","cmdid":1，

"missiongroupid":"100","status":0, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":1}

//AGV上传拿货完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":2，

"missiongroupid":"100","status":1, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":2}

//AGV上传 移动到站点1完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":3，

"missiongroupid":"100","status":2, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":3}

**机器会一直在站点1等待，WMS 处理完成这些货物之后，就会发消息给AGV，从站点1 移动到站点2，missiongroupid必须一样，robotid必须指定2001**

WMS->AGV：

{

"cmdcode":"SS",

"cmdid": 2,

"missiongroups" :[

{"startx":15,"starty":20, "endx":16,"endy":22,"robotid":2001,"missiongroupid":"100" }

]}

//AGV收到指令，ACK应答

AGV->WMS：

{"cmdcode":"ack","cmdid":2}

//AGV 指定2001执行，上传任务开始执行状态

AGV->WMS：

{"cmdcode":"status","cmdid":4，

"missiongroupid":"100","status":0, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":4}

//AGV上传移动到站点1状态（它已经在站点1位置，也会上传一个状态）

AGV->WMS：

{"cmdcode":"status","cmdid":5，

"missiongroupid":"100","status":1, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":5}

//AGV上传 移动到站点2完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":6，

"missiongroupid":"100","status":2, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":6}

**机器会一直在站点2等待，WMS 处理完成这些货物之后，就会发消息给AGV，从站点2把货物放回到货位1，missiongroupid必须一样，robotid必须指定2001**

WMS->AGV：

{

"cmdcode":"SL",

"cmdid": 3,

"missiongroups" :[

{"startx":16,"starty":22, "endx":10,"endy":5, "endz":3，

"robotid":2001,"missiongroupid":"100" }

]}

//AGV收到指令，ACK应答

AGV->WMS：

{"cmdcode":"ack","cmdid":3}

//AGV 指定2001执行，上传任务开始执行状态

AGV->WMS：

{"cmdcode":"status","cmdid":7，

"missiongroupid":"100","status":0, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":7}

//AGV上传移动到站点2状态（它已经在站点2位置，也会上传一个状态）

AGV->WMS：

{"cmdcode":"status","cmdid":8，

"missiongroupid":"100","status":1, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":8}

//AGV上传 把货物放回到货位1完成状态

AGV->WMS：

{"cmdcode":"status","cmdid":9，

"missiongroupid":"100","status":2, "robotid":2001 }

//WMS收到指令，ACK应答

WMS->AGV：

{"cmdcode":"ack","cmdid":9}

* 1. **LS,SL配套使用**

跟LS，SS，SL配置使用类似，少了SS这个环节而已