

机构脚本接口字典

机构脚本接口字典

版本

更新日期

更新说明

文档状态

维护责任人

V1.0

2024.11.4

发布

使用中

适用范围

此文档适用 RBK 版本为 3.4.6.2105。

接口字典

序号

接口

接口名称

输入

输出

RBK版本

1

addDisableDepthId

禁用指定id数组的深度相机避障功能

id:list

无

3.4.6.21~

2

addDisableLaserId

禁用指定id数组的激光雷达避障功能

id:list

无

3.4.6.21~

3

addMoveTask

增加任务，对应3051

msg: str

int (0)

4

addMoveTaskList

增加任务队列，对应3066

msg: str

int (0)

3.3.5.80~

5

armBinTask

调用机械臂动作的服务

task_id: int, cmd: str

无

3.3.5.80~

6

armControl

调用控制机械臂运动状态接口，比如减速

json_str: str

str

3.3.5.80~

7

armPause

机器臂暂停运动

无

无

3.3.5.80~

8

armResume

机器臂恢复运动

无

无

3.3.5.80~

9

armStop

机器臂停止运动

无

无

3.3.5.80~

10

battery

获得电池数据

无

dict

3.3.5.80~

11

binDetection

库位检测

seq: int

无

3.3.5.80~

12

clearBlockError

清除阻挡52200错误

无

无

3.3.5.80~

13

clearContainer

清除车上特定库位或者背篓的状态

container_name: str

bool

3.3.5.80~

14

clearContainerByGoodsId

清除车上特定库位或者背篓的状态

goods_id: str

bool

3.3.5.80~

15

clearGData

清除全局字典变量

无

无

3.4.6.21~

16

clearDisableDepthId

清除禁用的深度相机

无

无

3.4.6.21~

17

clearDisableLaserId

清除禁用的激光

无

无

3.4.6.21~

18

clearError

清除特定编号的Error

code: int

无

3.3.5.80~

19

clearGoodsShape

去除agv身上的状态

无

bool

3.3.5.80~

20

clearNotice

清除Notice

code: int

无

3.3.5.80~

21

clearWarning

清除特定编号的Warning

code: int

无

3.3.5.80~

22

controller

获得控制器数据

无

dict

23

Di

获得Di数据

无

dict

3.3.5.80~

24

disableMotor

电机去使能

name: str

无

3.4.6.21~

25

Do

获得Do数据

无

dict

3.3.5.80~

26

doRec

进行识别

filename: str

无

3.3.5.80~

27

doRecWithAngle

进行识别,包含识别机构在agv坐标系下的角度

filename:str, a:float

无

3.3.5.80~

28

doRecWithRegion

进行识别,包含识别的有效范围

filename:str, x: float, y: float, yaw: float, radius:float

无

3.4.6~

29

enableMotor

电机使能

name: str

无

3.4.6.21~

30

errorExits

查询特定编号的Error是否存在

code: int

无

3.3.5.80~

31

fork

获得货叉数据

无

dict

3.3.5.80~

32

forkGoods

货叉上加载或者卸载货物模型及货物检测DI

load: bool, recfile: str

无

3.3.5.80~

33

getArmlInfo

返回机械臂信息

无

dict

3.3.5.80~

34

getBinDetectionResult

获取库位检测结果

无

dict

3.4.6.21~

35

getCanFrame

获取当前的 CanFrame

无

dict

3.4.6.21~

36

getContainers

获取当前车子上库位或者背篓货物的状态

无

list

3.3.5.80~

37

getCount

获得当前任务已经循环的次数

无

int

3.3.5.80~

38

getCurrentAdvancedArea

机器人运行时，当前所在高级区域的属性

无

dict

39

getCurrentPathProperty

机器人运行时，当前路线上的属性

无

dict

3.4.6.21~

40

getCurrentTaskStatus

获取当前任务状态

无

无

3.4.6.21~

41

getGData

获取全局字典

无

dict

3.4.6.21~

42

getDistanceSensor

获取距离节点信息

无

dict

3.3.5.80~

43

getForkPressure

货叉测得重量

无

float

3.3.5.80~

44

getForkPressureADC

货叉压力传感器adc值

无

float

3.3.5.80~

45

getForkTipObsDist

获取叉尖相机超托情况，0.7m内最近点云聚类的最远点距离，需要配合triggerCameraOn 和 triggerCameraOff

json_str:str

float

3.4.6.21~

46

getLastMoveStatus

获取上一个周期的运行状态、障碍物状态

无

dict

3.4.6.21~

47

getLM

获取点位坐标

name: str, flag: bool

tuple

3.3.5.80~

48

getMinDynamicObs

获得离机器最近的一个动态障碍物坐标。 如果没有障碍物反馈0,0.

无

list

3.3.5.80~

49

getModbusData

读取寄存器数据

type:str, addr:int, size:int

list

3.4.6.21~

50

getMoveStatus

获取当前周期的运行状态、障碍物状态

无

dict

3.4.6.21~

51

getMsg

获取消息名称

type_name: str

dict

3.3.5.80~

52

getNearestLaserPoint

获取与指定激光距离最近的激光点与激光中心的距离和朝向

laser_id: int

无

3.4.6.21~

53

getNextSpeed

获取当前NavSpeed的速度

无

dict

3.3.5.80~

54

getRecFile

获得识别文件的原始数据

name: str

dict

3.3.5.80~

55

getRecFileFromTask

通过任务获取识别文件

无

str

3.3.5.80~

56

getRecResult

获取识别结果

无

dict

3.3.5.80~

57

getRecResults

获取第 index 个识别结果

index: int

dict

58

getRecResultSize

获取识别结果的个数

无

int (0)

3.4.6.21~

59

getRecStatus

获取识别状态

无

int

3.3.5.80~

60

getRobotFile

获得模型文件的原始数据

无

dict

3.3.5.80~

61

getTriggleScriptArgs

获取TriggleScript的参数

无

str

3.3.5.80~

62

getTriggleScriptName

获取TriggleScript的名称

无

str

3.3.5.80~

63

goForkPath

叉车依据规划的路径导航，需要先调用 resetGoForkPath

无

无

3.3.5.80~

64

goMapPath

按地图路线行走

task: str

int

3.4.6.21~

65

goPath

控制AGV移动

无

bool

3.3.5.80~

66

hasGoods

获取身上是否有货物的状态

无

bool

3.3.5.80~

67

hasTriggleScript

监测是否有TriggleScript触发

无

bool

3.3.5.80~

68

initForkCollisionCheck

初始化货叉移动时的碰撞检测，读取模型文件fork设备BackLaser相应的参数

无

无

3.3.5.80~

69

isAllMotorsReached

所有电机是否到位

无

bool

3.3.5.80~

70

isAnyErrorExists

检测是否有错误存在

无

无

3.3.5.80~

71

isMotorPositionReached

电机是否到达特定位置

motor_name: str, pos: float, stopDI: int

bool

3.3.5.80~

72

isMotorReached

查看电机是否到位，需要在setMotorPosition或者setMotorSpeed后使用

motor_name: str

bool

3.3.5.80~

73

isMotorStop

查询电机是否停止

motor_name: str

bool

3.3.5.80~

74

isPathReached

agv**是否完成线路**

无

bool

3.3.5.80~

75

jack

获得货叉数据

无

dict

3.3.5.80~

76

laserCollision

检测激光点是否和自身碰撞

ids: list

bool

3.3.5.80~

77

loadStatus

屏蔽指定距离相机的3d点云，通知相机插件读取货叉当前高度

dist:float

无

3.4.6.21~

78

loc

获得定位数据

无

dict

3.3.5.80~

79

logDebug

将字符串输出到log文件中，等级为Debug

ss: str

无

3.3.5.80~

80

logError

将字符串输出到log文件中，等级为Error

ss: str

无

3.3.5.80~

81

logInfo

将字符串输出到log文件中，等级为Info

ss: str

无

3.3.5.80~

82

logWarn

将字符串输出到log文件中，等级为Warning

ss: str

无

3.3.5.80~

83

magnetic

获得磁条数据

无

dict

3.3.5.80~

84

moveTask

获得任务信息以字典类型返回

无

dict

3.3.5.80~

85

navSpeed

获得当前速度数据

无

dict

3.3.5.80~

87

noticeExits

查询特定编号的Notice是否存在

code: int

无

3.3.5.80~

88

odo

获得里程数据

无

dict

3.3.5.80~

89

openSpeed

让agv按vx,vy,vw行走，此函数考虑了碰撞检测

vx: float, vy: float, vw: float

无

3.3.5.80~

90

pgv

获得pgv数据

无

dict

91

publishSpeed

将当前电机控制方案，进行速度规划然后下发

无

bool

3.3.5.80~

92

recAdjustYTheta

在有限空间内来回调整,使车子对准目标点

task: str

int

3.3.5.80~

93

recAndGoPathDi

识别并且行走

task: str

int

3.3.5.80~

94

RecognizeBarCode

获取一维码信息

name: str, id: str

str

3.3.5.80~

95

recTargetObs

识别目标点一定空间范围内是否有物体

x: float, y: float, yaw: float, filename:str

无

3.4.6~

96

resetGoForkPath

重置叉车去往识别点的路径规划

x: float, y: float, yaw: float, back_dist: float, min_ahead_dist: float, ahead_dist: float

无

3.3.5.80~

97

resetGoMapPath

行走的动作

无

无

3.3.5.80~

98

resetLocalShelfArea

取消顶升上的货架

无

bool

3.3.5.80~

99

resetMotor

将电机重置为不启用状态

motor_name: str

bool

3.3.5.80~

100

resetPath

让agv沿着规划的线路行驶

无

无

3.3.5.80~

101

resetRec

重置识别模块

无

无

3.3.5.80~

102

resetRecAdjustYTheta

在有限空间内来回调整,使车子对准目标点

无

无

3.3.5.80~

103

resetRecAndGoPathDi

重置识别行走的动作

无

无

3.3.5.80~

104

resetTriggleScript

重置Triggle信息

无

无

3.3.5.80~

105

rfid

获得rfid数据

无

dict

3.3.5.80~

106

robokitVersion

获取 robokit 版本号, from: 3.3.5.11

无

str

3.3.5.80~

107

scannerCode

调用机械臂识别二维码的接口

task_id: int

无

3.3.5.80~

108

sendCanFrame

DSP提供sendCanFrame接口

channel: int, can_id: int, dlc: int, extend: bool, can_string: str

无

3.3.5.80~

109

sensorPointCloud

获得后视激光点云信息以字典类型返回

无

dict

110

setBlockError

设置阻挡52200错误

无

无

3.3.5.80~

111

setBlockReason

设置阻挡原因

collision_type: int, x: float, y: float, id: int

无

3.3.5.80~

112

setContainer

设置车子上库位或者背篓货物

container_name: str, goods_id: str, desc: str

bool

3.3.5.80~

113

setDIValid

设置DI 是否生效

id: int, valid: bool

无

3.4.6.21~

114

setDO

控制DO的开关

id: int, status: bool

bool

3.3.5.80~

115

setError

输出53000的Error

ss: str

无

3.3.5.80~

116

setGData

设置全局字典变量（保存在内存中）

value:dict

无

3.4.6.21~

117

setGoForkForkPos

重置叉车去往识别点的路径规划

x: float, y: float, theta: float, hold_dir: float

无

3.4.6.21~

118

setGoodsShape

设置货物形状，并且告诉rbk车上装载有货物了。如果head,tail, width都小于等于0，则没有货物形状。货物的0，0点与小车的0，0点一样

head: float, tail: float, width: float

int (0)

3.3.5.80~

119

setInfo

输出脚本调试信息

ss:str

无

3.3.5.80~

120

setLocalShelfArea

加载顶升上的货物模型

object_model_path: str

bool

3.3.5.80~

121

setModbusData

写入寄存器数据

type:str, addr:int, data:list

bool

3.4.6.21~

122

setMotorCalib

电机标零

motor_name: str

无

3.4.6.21~

123

setMotorPosition

控制线性电机到特定位置

motor_name: str, pos: float, maxVel: float, stopDI: int

bool

3.3.5.80~

124

setMotorSpeed

让电机以某个速度运行

name: str, vel: float, stopDI: int

bool

3.3.5.80~

125

setNextSpeed

设置准备下发的速度

nav:str

bool

3.3.5.80~

126

setNotice

输出57300的Notice

ss: str

无

3.3.5.80~

127

setObsStopDist

设置避障距离

dist: float

无

3.4.6.21~

128

setPathBackMode

路径导航是否倒走

a:bool

无

3.3.5.80~

129

setPathHoldDir

路径导航时保持角度

a:float

无

3.3.5.80~

130

setPathMaxRot

路径导航的最大角速度

a:float

无

3.3.5.80~

131

setPathMaxSpeed

路径导航的最大速度

a:float

无

3.3.5.80~

132

setPathOnRobot

让agv在agv坐标系下以特定线路行走

x: list, y: list, angle: float

无

3.3.5.80~

133

setPathOnWorld

让agv在世界坐标系下以特定线路行走

x: list, y: list, angle: float

无

3.3.5.80~

134

setPathReachAngle

路径导航的到点角度精度

a:float

无

3.3.5.80~

135

setPathReachDist

路径导航的到点精度

a:float

无

3.3.5.80~

136

setPathUseOdo

路径导航是否用里程定位

a:bool

无

3.3.5.80~

137

setPickRobotError

多料箱车专用报错码: 53800~53899

code: int, ss: str

无

3.3.5.80~

138

setPickRobotWarning

多料箱车报警码: 55800~55899

code: int, ss: str

无

139

setSound

播放音乐

name:str, flag:bool

无

3.3.5.80~

140

setSoundCount

播放音乐

name:str, count:int

无

3.3.5.80~

141

setSteerAngle

转动舵角

name:str, angle:float

bool

3.4.6.21~

142

setUserError

用户报错码: 53900~53999

code: int, ss: str

无

3.3.5.81~

143

setUserWarning

用户报警码: 55900~55999

code: int, ss: str

无

3.3.5.80~

144

setVirtualDI

设置虚拟 DI 状态

index: int, status: bool

无

3.3.5.81~

145

setWarning

输出55300的Warning

ss: str

无

3.3.5.80~

146

sound

获得音频数据

无

dict

3.3.5.81~

147

speedDecomposition

将导航速度分解，目前只有单舵轮和双舵轮有效

nav:str

str

3.3.5.80~

148

stopCurrentBlock

终止当前调度的动作块

无

无

3.3.5.81~

149

stopMotor

停止所有非行走的电机

无

无

3.3.5.80~

150

stopRobot

让agv停下来

flag:bool

无

3.3.5.81~

151

stopSound

停止播放音乐

flag: bool

无

3.3.5.80~

152

switchMap

切换地图

map:str, switchPoint:str, center_x:float, center_y:float, initial_angle:float

无

3.3.5.81~

153

triggerCameraOff

实时识别、叉尖反向超托检测时配合 triggerCameraOn 使用

无

无

3.4.6.21~

154

triggerCameraOn

实时识别、叉尖反向超托检测时配合 triggerCameraOff 使用

无

无

155

updateModelParamsByJsonStr

推送机器人模型文件，先从r.getRobotFile()获取，更新时需要在模型名称后面带个时间戳用于区分，比如"model": "lidar-202507071331"

model_json_str:str

bool

3.4.6.21~

156

warningExits

查询特定编号的Warning是否存在

code: int

无

3.3.5.80~

接口说明

isAllMotorsReached

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例:

msg = isAllMotorsReached()

print(msg)

接口返回数据示例:

Output: True

isMotorReached

入口参数解释:

字段

数据类型

示例值

描述

motor_name

str

"motor0"

电机名称

方法调用示例：

msg = r.isMotorReached("motor0")

print(msg)

接口返回数据示例：

Output: True

isMotorPositionReached

入口参数解释：

字段

数据类型

示例值

描述

motor_name

str

"motor0"

电机名称

pos

pos

位置

stopDI

int

3

到位DI

方法调用示例：

msg = r.isMotorPositionReached("motor0", 0.6, 3)

print(msg)

接口返回数据示例：

Output: True

isMotorStop

入口参数解释：

字段

数据类型

示例值

描述

motor_name

str

"motor0"

电机名称

方法调用示例：

```
msg = r.isMotorStop("motor0")

print(msg)
```

接口返回数据示例：

Output: True

getMsg#

入口参数解释：

字段

数据类型

示例值

描述

type_name

str

" "

消息名称

方法调用示例：

r.getMsg("")

接口返回数据示例：

Output: True

getCount

接口

描述

参数

返回数据类型

getCount

获取当前任务循环的次数

无

int

方法调用示例：

msg = r.getCount()

print(msg)

接口返回数据示例：

Output: 16

odo

接口

描述

参数

返回数据类型

odo

获得里程数据

无

dict

方法调用示例:

msg = r.odo()

print(msg)

接口返回数据示例:

Output:

{

```
"angle": -4.5602541,
"cycle": 4651341,
"detect_skid": false,
"follow_err": false,
```

"header": {


```
"data_nsec": "415223169185955",  
"frame_id": "/odo",  
"pub_nsec": "415223169186125",
```

"seq": "0"

```
},  
"is_stop": true,
```

"motor_info": [

{

```
"calib": false,  
"can_id": 2,  
"can_router": 1,  
"current": 1,  
"emc": false,  
"encoder": 0,  
"err": false,  
"error_code": 0,  
"follow_err": false,  
"motor_name": "motor_Right",  
"passive": false,  
"position": 0,  
"raw_position": 0,  
"speed": 0,  
"stop": false,  
"temperature": 1,  
"type": 0,
```

"voltage": 0

```
},
```

{

```
        "calib": false,
        "can_id": 1,
        "can_router": 1,
        "current": 1,
        "emc": false,
        "encoder": 0,
        "err": false,
        "error_code": 0,
        "follow_err": false,
        "motor_name": "motor_Left",
        "passive": false,
        "position": 0,
        "raw_position": 0,
        "speed": 0,
        "stop": false,
        "temperature": 1,
        "type": 0,
```

"voltage": 0

}

```
    ],
    "vel_rotate": 0,
    "vel_x": 0,
    "vel_y": 0,
    "x": 26.302389137252263,
```

"y": -16.080223023014746

}

返回字段解释：

字段

数据类型

示例值

描述

angle

float

-4.5602541

小车当前角度，单位 rad

cycle

int

4651341

detect_skid

bool

false

小车是否打滑

follow_err

bool

false

is_stop

bool

true

小车是否静止

vel_rotate

float

0

旋转速度，单位 rad/s

vel_x

float

0

x 方向速度，单位 m/s

vel_y

float

0

y 方向速度，单位 m/s

x

float

26.302389137252263

世界坐标系下 x 轴坐标，单位 m

y

float

-16.080223023014746

世界坐标系下 y 轴坐标，单位 m

header 字段解释：

字段

数据类型

示例值

描述

data_nsec

string

"415223169185955"

数据获取的时间戳

frame_id

string

/odo

执行的函数名称

pub_nsec

string

"415223169186125"

发布时间戳，单位 纳秒

seq

string

"0"

目前没有用

motor_info 字段解释：

字段

数据类型

示例值

描述

calib

bool

false

是否标零完成

can_id

int

1

can 的 id

can_router

int

1

can 的 路由

current

int

1

电流, 单位 A

emc

bool

false

是否急停

encoder

int

0

编码器

err

bool

false

是否报错

error_code

int

0

错误码

follow_err

bool

false

是否电机跟随异常

motor_name

string

"motor_Left"

电机名称

passive

bool

false

是否为被动电机(指不执行，仅反馈的电机)

position

float

0

位置, 单位: (linear 为 m; steer、spin、rotation 为 rad)

raw_position

float

0

未经处理的电机位置, 单位: (linear 为 m; steer、spin、rotation 为 rad)

speed

float

0

速度, 单位: (linear 为 m/s; steer、spin、rotation 为 rad/s)

stop

bool

false

是否停止

temperature

float

1

电机温度

type

int

0

电机类型

0=WALK

1=SPEED

2=SPIN

3=LINEAR

4=ROTATION

5=DO

voltage

int

0

电机电压，单位 V

loc

接口

描述

参数

返回数据类型

loc

获取AGV定位数据

无

dict

接口返回数据示例:

Output:

{

```
"angle": -3.01641845703125,  
"confidence": 0.2693849443659164,  
"correction_errs": [0.004627956391526823, -0.012603017829776946,  
0.035735549737506714],
```

"header": {

```
"data_nsec": "417519229609708",  
"frame_id": "",  
"pub_nsec": "0",
```

"seq": "0"

```
},  
"in_forbidden_area": false,  
"loc_method": 0,  
"loc_state": 0,  
"reliabilities": [0, 0, 0, 0],  
"similarity": 0,  
"update_reason": 0,  
"x": 14.005484914171664,
```

"y": -5.584118684568884

}

返回字段解释:

字段

数据类型

示例值

描述

angle

float

-3.01641845703125

小车当前角度

confidence

float

0.2693849443659164

小车当前的置信度

correction_errs

list

0.004627956391526823,

0.035735549737506714

小车的定位的协方差

in_forbidden_area

bool

false

禁止区域，默认为 false（保留）

loc_method

int

0

定位方式

0=2D 激光

1=2D SLAM

2=PGV

3=反光柱

4=3D 激光

5=条形码

loc_state

int

0

定位状态

0=Normal

1=Skidding

2=LowConfidence

reliabilities

list

0, 0, 0, 0

定位可靠性

similarity

int

0

定位相似度

update_reason

enum

0

默认值（不区分 1, 2, 3）

1

里程计定位（保留）

2

雷达定位（保留）

3

雷达融合里程计定位（未保留）

4

二维码定位

x

float

14.005484914171664

世界坐标系下 x 轴坐标

y

float

-5.584118684568884

世界坐标系下 y 轴坐标

navSpeed

接口

描述

参数

返回数据类型

navSpeed

获取当前速度数据

无

dict

方法调用示例:

```
msg = r.navSpeed()
```

```
print(msg)
```

接口返回数据示例:

Output:

```
{
```

```
"header": {
```

```
    "data_nsec": "418162132735146",  
    "frame_id": "",  
    "pub_nsec": "418162132735146",
```

```
"seq": "0"
```

```
    },  
    "is2move": true,
```

```
"motor_cmd": [
```

```
{
```

```
        "can_id": 0,
        "can_router": 0,
        "io_cmd": 3,
        "motor_name": "motor_Left",
```

"type": 0, "value": 0

```
    },
```

{

```
        "can_id": 0,
        "can_router": 0,
        "io_cmd": 3,
        "motor_name": "motor_Right",
        "type": 0,
```

"value": 0

}

```
    ],
    "rotate": 0,
    "x": 0,
```

"y": 0

}}

返回字段解释：

字段

数据类型

示例值

描述

is2move

bool

true

是否将要运动

rotate

float

0

角速度，单位 rad/s

x

float

0

x 方向上的速度，单位 m/s

y

float

0

y 方向上的速度，单位 m/s

motor_cmd 字段解释:

字段

数据类型

示例值

描述

can_id

int

0

can 的 id

can_router

int

0

can 的路由

io_cmd

int

3

Do 类型电机的控制指令。1 表示正向；2 表示负向；3表示停止；0表示空闲

motor_name

string

"motor_Right"

电机名称

type

int

0

电机类型

0=WALK

1=SPEED

2=SPIN

3=LINEAR

4=ROTATION

5=DO

value

int

0

速度控制的电机为速度

位置控制的电机为位置

battery

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

battery_data = r.battery()

print(msg)

接口返回数据示例：

Output:

{

```
"charge_current": -1,
"charge_voltage": 1,
"cycle": 0,
```

"is_charging": false

```
"max_charge_current": 0,
"max_charge_voltage": 0,
"percetage": 1,
"temperature": 0,
```

"user_data": ""

}

返回字段解释：

字段

数据类型

示例值

描述

charge_current

int

-1

当前充电状态

charge_voltage

int

1

充电电压

cycle

int

0

电池循环次数

is_charging

bool

false

是否正在充电中

max_charge_current

int

0

允许充电的最大电流(-1 = 该电池不支持此功能), 单位 A

max_charge_voltage

int

0

允许充电的最大电压(-1 = 该电池不支持此功能), 单位 V

percetage

int

1

电池电量百分比

temperature

int

0

电池温度

user_data

string

""

电池用户自定义数据

rfid

接口

描述

参数

返回数据类型

rfid

获得rfid数据

无

dict

方法调用示例：

msg = r.rfid()

print(msg)

接口返回数据示例：

Output:

```
{  
  
  "rfid_nodes": []  
  
}
```

返回字段解释:

字段

数据类型

示例值

描述

rfid_nodes

list

[]

扫描到的所有 RFID 标签 id, 如果没扫描到 RFID 标签, 则为空数组

magnetic

接口

描述

参数

返回数据类型

magnetic

获得磁条数据

无

dict

方法调用示例：

```
msg = r.magnetic()
```

```
print(msg)
```

接口返回数据示例：

Output:

```
{  
  
"magnetic_nodes": []  
}
```

返回字段解释：

字段

数据类型

示例值

描述

magnetic_nodes

list

[]

磁条节点信息

Di

接口

描述

参数

返回数据类型

Di

获取机器人DI数据

无

dict

方法调用示例:

msg = r.Di()

print(msg)

接口返回数据示例:

Output:

{

"max_node": 0,

"node": [

{

```
"forbidden": false,
"func": "",
"id": 8,
"maxdist": 0,
"mindist": 0,
"posx": [],
"posy": [],
"range": 0,
"shape": "",
"source": "virtual",
"status": false,
"type": "",
"x": 0,
"y": 0,
"yaw": 0,
```

"z": 0

```
},
```

{

```
"forbidden": false,
"func": "",
"id": 7,
"maxdist": 0,
"mindist": 0,
"posx": [],
"posy": [],
"range": 0,
"shape": "",
"source": "virtual",
"status": false,
"type": "",
"x": 0,
"y": 0,
"yaw": 0,
```

"z": 0

```
},
```

```
{
```

```
    "forbidden": false,  
    "func": "",  
    "id": 6,  
    "maxdist": 0,  
    "mindist": 0,  
    "posx": [],  
    "posy": [],  
    "range": 0,  
    "shape": "",  
    "source": "virtual",  
    "status": false,  
    "type": "",  
    "x": 0,  
    "y": 0,  
    "yaw": 0,
```

"z": 0

```
},
```

```
{
```

```
    "forbidden": false,  
    "func": "",  
    "id": 5,  
    "maxdist": 0,  
    "mindist": 0,  
    "posx": [],  
    "posy": [],  
    "range": 0,  
    "shape": "",  
    "source": "virtual",  
    "status": false,  
    "type": "",  
    "x": 0,  
    "y": 0,  
    "yaw": 0,
```

"z": 0

}

]

}

返回字段解释:

字段

数据类型

示例值

描述

max_node

int

0

node 字段解释:

字段

数据类型

示例值

描述

forbidden

bool

false

是否被禁止

func

string

""

DI 的功能

stop

slowdown

none

id

int

5

DI 的 id

maxdist

float

0

arc 型 DI 的最大可用距离,单位 m

mindist

float

0

arc 型 DI 的最小可用距离, 单位 m

posx

list

[]

碰撞条形状, x坐标

posy

list

[]

碰撞条形状, y坐标

range

float

0

arc 型 DI 的可用角度,单位 degree

shape

string

""

DI 的形状

arc

vertex

source

string

"virtual"

虚拟 DI

status

bool

false

status: 表示高低电平

true = 高电平

false = 低电平

type

string

""

DI 的类型

collision

infrared

ultrasonic

fallingDown

goodsDetect

ignoreTask

none

x

float

0

arc 型 DI 在机器人坐标系下的 x 坐标，单位 m

y

float

0

arc 型 DI 在机器人坐标系下的 y 坐标，单位 m

yaw

float

0

偏航角，单位：rad

z

float

0

arc 型 DI 在机器人坐标系下的 z 坐标, 单位 m

Do

接口

描述

参数

返回数据类型

Do

获取机器人DO数据

无

dict

方法调用示例:

msg = r.Do()

print(msg)

接口返回数据示例:

Output:

{

"max_node": 0,

"node": [

{

```
"func": "",  
"id": 6,  
"source": "",
```

"status": false

```
},
```

{

```
"func": "",  
"id": 5,  
"source": "",
```

"status": false

```
},
```

{

```
"func": "",  
"id": 4,  
"source": "",
```

"status": true

```
},
```

{

```
"func": "",  
"id": 3,  
"source": "",
```

"status": false

```
},
```

```
{
```

```
"func": "",  
"id": 2,  
"source": "",
```

"status": false

```
},
```

```
{
```

```
"func": "",  
"id": 1,  
"source": "",
```

"status": true

```
},
```

```
{
```

```
"func": "",  
"id": 0,  
"source": "",
```

"status": true

}

]

}

返回字段解释:

字段

数据类型

示例值

描述

max_node

int

0

node 字段解释:

字段

数据类型

示例值

描述

func

string

"stop" 表示阻挡

"slowdown" 表示减速

id

int

0

DO 的 id

source

string

"""

status

bool

true

true 为高电平, false 为低电平

pgv

接口

描述

参数

返回数据类型

pgv

获得pgv数据

无

dict

方法调用示例:

```
msg = r.pgv()
```

```
print(msg)
```

接口返回数据示例:

Output:

```
{
```

```
"pgvs": [
```

```
{
```

```
    "device_address": 1,  
    "error_code": 0,
```

```
"header": {
```

```
    "data_nsec": "3277108309375",  
    "frame_id": "",  
    "pub_nsec": "0",
```

```
"seq": "0"
```



```
    },
    "is_DMT_detected": True,
    "is_bar_code": False,
```

"pgv_info": {

```
    "angle_unit": 0.1,
    "coordinate": "code",
    "func": "adjustLocalization",
    "is_upside": True,
    "x": -0.00034550696844757667,
    "xrange": 0.1,
    "xunit": 0.0001,
    "y": 0.0006722651102830519,
    "yaw": 3.137873184961928,
    "yrange": 0.1,
    "yunit": 0.0001,
```

"z": 0

```
    },
    "tag_diff_angle": 3.1413638434659164,
    "tag_diff_x": 0.03655032772140157,
    "tag_diff_y": -0.017536177506792275,
    "tag_value": 894,
```

"warning_code": 0

}

]

}

返回字段解释:

字段

数据类型

示例值

描述

device_address

int

1

PGV的ID

error_code

int

0

报错代码

is_DMT_detected

bool

True

是否检测到该二维码

is_absolute_X_pos_valid

bool

True

tag_diff_angle

float

1774

PGV 与 二维码的角度之差 单位: rad

tag_diff_x

float

256

PGV 与 二维码的在 x 方向的差值 单位: m

tag_diff_y

float

47

PGV 与 二维码的在 y 方向的差值 单位: m

tag_value

int

1

二维码的 id 号

warning_code

int

0

报警码

pgv_info

header字段解释:

字段

数据类型

示例值

描述

data_nsec

int

1

获取数据时的时间戳

frame_id

pub_nsec

数据发布时间戳，单位 纳秒

seq

pgv_info 字段解释:

字段

数据类型

示例值

描述

angle_unit

double

0.1

读码相机反馈的角度单位

coordinate

string

code

读码相机反馈数据的坐标系，例如code是二维码坐标系

func

string

adjustLocalization

读码相机应用功能

is_upside

bool

true

是否为上视相机

x

double

相对车体x方向的安装位置

xrange

double

0.1

读码相机x方向的有效范围

xunit

double

0.0001

读码相机反馈的x方向单位

y

double

0.2

相对车体y方向的安装位置

yaw

double

90

相对车体安装的角度

yrange

double

0.1

读码相机y方向的有效范围

yunit

double

0.0001

读码相机反馈的y方向单位

z

double

0.1

相机镜头到二维码的距离

sound

接口

描述

参数

返回数据类型

sound

获得音频数据

无

方法调用示例:

msg = r.sound()

print(msg)

接口返回数据示例:

Output:

{

```
"loop": true,  
"sound_name": "navigation.wav",
```

"status": 2

}

返回字段解释:

字段

数据类型

示例值

描述

loop

bool

true

是否循环播放，true：循环播放，false：不循环播放

sound_name

string

"navigation.wav"

播放的音频名称

status

int

2

0：未播放音频，1：音频暂停播放中，2：音频播放中

controller

接口

描述

参数

返回数据类型

controller

获取AGV控制器数据

无

dict

方法调用示例：

msg = r.controller()

print(msg)

接口返回数据示例：

Output:

{

```
"autoCharge": false,
"brake": false,
"driverEmc": false,
"electric": false,
"emc": false,
"humi": 0,
"isExternalControl": false,
"isIMUCalibrating": false,
>manualCharge": false,
"softEMC": false,
"temp": 0,
"voltage": 0,
```

"voltagebyAdc": 0

}

返回字段解释：

字段

数据类型

示例值

描述

autoCharge

bool

false

是否处于自动充电状态

brake

bool

false

是否处于制动状态

driverEmc

bool

false

是否处于急停状态

electric

bool

false

继电器状态（仅SRC2000）

emc

bool

false

急停

humi

int

0

控制器湿度

isExternalControl

bool

false

是否处于外部控制状态

isIMUCalibrating

bool

false

是否处于 IMU 标定状态

manualCharge

bool

false

是否处于手动充电状态

softEMC

bool

false

软急停

temp

float

0

温度

voltage

float

0

控制器电压

voltagebyAdc

float

0

检测外部电压（充电桩版控制器支持）

fork

接口

描述

参数

返回数据类型

fork

获取AGV货叉数据

无

dict

方法调用示例:

```
msg = r.fork()
```

```
print(msg)
```

接口返回数据示例:

Output:

```
{
```

```
"fork_error": 0,  
"fork_val": 0,  
"forward_in_place": false,  
"forward_val": 0,  
"get_goods": false,  
"height": 0,  
"height_in_place": false,  
"incline": 0,  
"incline_in_place": false,  
"incline_val": 0,  
"pressure_actual": 0,
```

```
"pressure_adc": 0
```

```
}
```

返回字段解释:

字段

数据类型

示例值

描述

fork_error

int

0

fork_val

int

0

forward_in_place

bool

false

货叉前移后退是否到位, true = 到位, false = 未到位

forward_val

float

0

货叉前移后退距离, 单位: m

get_goods

bool

false

height

float

0

货叉高度, 单位 m

height_in_place

bool

false

货叉高度是否到位, true = 到位, false = 未到位（弃用，目前该位随机器人任务状态而变化；当机器人的任务状态为完成时为true，当机器人运动时为false）

incline

int

0

incline_in_place

bool

false

incline_val

int

0

pressure_actual

int

0

货叉称重数据，单位：kg

pressure_adc

int

0

jack

接口

描述

参数

返回数据类型

jack

获取AGV顶升数据

无

dict

方法调用示例：

msg = r.jack()

print(msg)

接口返回数据示例：

Output:

{

```
"emc": false,
"enable": false,
"errorCode": 0,
"height": 0,
"isFull": false,
"mode": false,
"speed": 0,
```

"state": 0

}

返回字段解释：

字段

数据类型

示例值

描述

emc

bool

false

enable

bool

false

顶升是否启用, false = 未启用, true = 启用

errorCode

int

0

顶升错误码

height

float

0

顶升当前的高度，单位: m

isFull

bool

false

顶升机构上是否有料, false = 无料, true = 有料

mode

bool

false

运行模式, false = 手动, true = 自动

speed

float

0

顶升机构电机速率, 单位: mm/s

state

int

0

运行状态, 0x00 = 上升中, 0x01 = 上升到位, 0x02 = 下降中, 0x03 = 下降到位, 0x04 = 停止, 0xFF = 执行失败

moveTask

接口

描述

参数

返回数据类型

moveTask

获取AGV任务数据

无

dict

方法调用示例:

```
msg = r.moveTask()
```

```
print(msg)
```

接口返回数据示例:

Output:

```
{
```

```
    "dec_obs_expansion": 0.2,  
    "max_acc": 1,  
    "max_dec": 1,  
    "max_rot": 0.782907095366736,  
    "max_rot_acc": 2.0943951023931953,  
    "max_rot_dec": 1.5707963267948966,  
    "max_speed": 0.7853981633974483,  
    "obs_dec_dist": 3,  
    "obs_dec_speed": 0.2,  
    "obs_expansion": 0.1,  
    "obs_stop_dist": 1,
```

```
"params": [  
    {
```

```
{
```

```
        "key": "operation",
```

```
"string_value": "Script"
```

```
    },
```

```
{
```

<pre>"key": "script_name",</pre>
<pre>"string_value": "lin/test.py"</pre>
<pre>},</pre>
<pre>{</pre>
<pre>"bool_value": true,</pre>
<pre>"key": "allow_free_go"</pre>
<pre>}</pre>
<pre>], "reach_angle": 0.008726646259971648, "reach_dist": 0.005, "reach_method": "forward", "skill_name": "Action",</pre>
<pre>"target_name": "9"</pre>
<pre>}</pre>

返回字段解释：

字段

数据类型

示例值

描述

dec_obs_expansion

float

0.2

减速碰撞预测的机器人膨胀宽度，单位 m

max_acc

float

1

最大加速度，单位 m/s^2

max_dec

float

1

最大减速度，单位 m/s^2

max_rot

float

0.782907095366736

最大转速度，单位 rad/s

max_rot_acc

float

2.0943951023931953

最大转加速度，单位 rad/s^2

max_rot_dec

float

1.5707963267948966

最大转减速度，单位 rad/s^2

max_speed

float

0.7853981633974483

最大速度，单位 m/s

obs_dec_dist

float

3

减速预测的距离，单位 m

obs_dec_speed

float

0.2

减速预测的速度，单位 m/s

obs_expansion

float

0.1

碰撞预测的机器人扩展宽度，单位 m

obs_stop_dist

float

1

碰撞预测的距离，单位 m

reach_angle

float

0.008726646259971648

到点角度，单位 rad

reach_dist

float

0.005

到点距离精度，单位 m

reach_method

string

"forward"

到点朝向

skill_name

string

"Action"

执行动作名

target_name

string

"9"

目标站点 id

params 字段:

字段

数据类型

示例值

描述

key

string

"operation"

执行动作

string_value

string

Script

脚本动作

key

string

"script_name"

脚本名称

string_value

string

"lin/test.py"

key

string

"allow_free_go"

是否允许自由导航

bool_value

bool

true

getArmlInfo

接口

描述

参数

返回数据类型

getArmlInfo

获得机械臂信息

无

dict

方法调用示例:

msg = r.getArmlInfo()

print(msg)

接口返回数据示例:

Output:

{

```
"DI": [],
"DO": [],
"enabled": false,
"robot_status_bits": [],
"safety_mode": 0,
"safety_status_bits": [],
"taskId": 0,
```

"task_status": 0

}

返回字段解释:

字段

数据类型

示例值

描述

DI

list

[]

机械臂 DI

0-7 = Standard input

8-15 = Configurable input

16-17 = Tool input

DO

list

[]

机械臂 DO

0-7 = Standard input

8-15 = Configurable input

16-17 = Tool input

enabled

bool

false

是否启用机械臂，true = 启用，false = 不启用

robot_status_bits

list

[]

机器人状态位

0 = Is power on

1 = Is program running

2 = Is teach button pressed

3 = Is power button pressed

safety_mode

int

0

安全模式

1 = normal mode

2 = release mode

3 = recovery mode

safety_status_bits

list

[]

安全状态位

0 = Is normal mode

1 = Is reduced mode

2 = Is protective stopped

3 = Is recovery mode

4 = Is safeguard stopped

5 = Is system emergency stopped

6 = Is robot emergency stopped

7 = Is emergency stopped

8 = Is violation

9 = Is fault

10 = Is stopped due to safety

taskId

int

0

任务 id

task_status

int

0

任务状态

getDistanceSensor

接口

描述

参数

返回数据类型

getDistanceSensor

获得距离传感器信息

无

dict

方法调用示例:

msg = r.getDistanceSensor()

print(msg)

接口返回数据示例:

Output:

{

"node": []


```
}
```

返回字段解释：

字段

数据类型

示例值

描述

node

list

[]

距离传感器节点信息

sensorPointCloud

接口

描述

参数

返回数据类型

sensorPointCloud

获得后视激光点云信息

无

dict

方法调用示例:

```
msg = r.sensorPointCloud()
```

```
print(msg)
```

接口返回数据示例:

Output:

```
{
```

```
"global_cluster": [
```

```
{
```

```
"header":
```

```
{
```

```
        "data_nsec": "423939395152396",  
        "frame_id": "",  
        "pub_nsec": "0",
```

```
"seq": "0"
```

```
    },  
    "id": "1",
```

```
"point": [
```

```
{
```

```
"is_obstacle": true,  
"rssi": 100,  
"x": 24.781345578357573,  
"y": -12.40421655491872,
```

"z": 0

```
},
```

{

```
"is_obstacle": true,  
"rssi": 100,  
"x": 26.920356728196996,  
"y": -13.95355462811719,
```

"z": 0

```
},
```

{

```
"is_obstacle": true,  
"rssi": 100,  
"x": 24.96300860843852,  
"y": -12.314360624573617,
```

"z": 0

```
},
```

{

```
"is_obstacle": true,  
"rssi": 100, "x": 25.154097313239248,  
"y": -12.319513761761758,
```

"z": 0

```
      },
```

```
{
```

```
      "is_obstacle": true,  
      "rssi": 100,  
      "x": 25.18682711936516,  
      "y": -12.206342445234426,
```

"z": 0

```
      },
```

```
{
```

```
      "is_obstacle": true,  
      "rssi": 100,  
      "x": 25.19041497687728,  
      "y": -12.073297463125794,
```

"z": 0

```
}
```

```
    ],
```

"type": 5

```
      },
```

```
{
```

"header": {

```
"data_nsec": "423939437107877",  
"frame_id": "",  
"pub_nsec": "0",
```

"seq": "0"

```
},  
"id": "",  
"point": [],
```

"type": 6

}

]

}

global_cluster 字段解释:

字段

数据类型

示例值

描述

id

string

"1"

点云类型的id号，如果是激光的话，为index，APIObstacle则为障碍物名称

type

int

5

点云数据的类型

Ultrasonic = 0

Laser = 1

Fallingdown = 2

Collision = 3

Infrared = 4

VirtualPoint = 5

APIObstacle = 6

ReservedPoint = 7

DiUltrasonic = 8

DepthCamera = 9 ReservedDepthCamera = 10

DistanceNode = 11

ADCollision = 12

head字段解释:

字段

数据类型

示例值

描述

data_nsec

string

"423939395152396"

获取数据的时间戳

frame_id

string

""

目前就是空

pub_nsec

string

"0"

数据发布时间戳

seq

string

"0"

point 字段解释：

字段

数据类型

示例值

描述

is_obstacle

bool

true

是否阻碍，true：阻碍，false：不阻碍

rss_i

int

100

信号强度

x

float

25.19041497687728

点云在世界坐标系下的 x 坐标轴

y

float

-12.073297463125794

点云在世界坐标系下的 y 坐标轴

z

float

0

点云在世界坐标系下的 z 坐标轴

noticeExits

入口参数解释：

字段

数据类型

示例值

描述

code

int

57300

Notice 编号

方法调用示例：

r.warningExits(57300)

errorExits

入口参数解释：

字段

数据类型

示例值

描述

code

int

53000

Error编号

方法调用示例：

r.warningExits(53000)

warningExits

入口参数解释：

字段

数据类型

示例值

描述

code

int

Warning编号

方法调用示例:

r.warningExits(55300)

isPathReached

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例:

msg = r.isPathReached()

print(msg)

接口返回数据示例:

Output: True

setInfo

入口参数解释：

字段

数据类型

示例值

描述

ss

str

""

/

方法调用示例：

self.states["key"] = "value"

r.setInfo(json.dumps(self.state))

getNextSpeed

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

```
msg = r.getNextSpeed()
```

```
print(msg)
```

接口返回数据示例：

Output:

{

"header": {

```
    "data_nsec": "419669544810406",  
    "frame_id": "",  
    "pub_nsec": "419669544810406",
```

"seq": "0"

```
    },  
    "is2move": false,
```

"motor_cmd": [

{

```
        "can_id": 0,
        "can_router": 0,
        "io_cmd": 3,
        "motor_name": "motor_Left",
```

"type": 0, "value": 0

```
    },
```

{

```
        "can_id": 0,
        "can_router": 0,
        "io_cmd": 3,
        "motor_name": "motor_Right",
        "type": 0,
```

"value": 0

}

```
    ],
    "rotate": 0,
    "x": 0,
```

"y": 0

}

返回字段解释：

字段

数据类型

示例值

描述

is2move

bool

false

是否将要运动

rotate

float

0

角速度，单位 rad/s

x

float

0

x 方向上的速度，单位 m/s

y

float

0

y 方向上的速度，单位 m/s

getTriggleScriptName

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

msg = r.getTriggleScriptName()

print(msg)

接口返回数据示例：

Output: "scriptName"

getTriggleScriptArgs

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

msg = r.getTrigggleScriptArgs()

print(msg)

接口返回数据示例：

Output: "args"

hasTrigggleScript

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例:

```
msg = r.hasTriggleScript()

print(msg)
```

接口返回数据示例:

Output: True

getRecStatus

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例:

```
msg = r.getRecStatus()

print(msg)
```

接口返回数据示例:

Output: 0

返回字段解释:

字段

数据类型

示例值

描述

无

int

0

0 刚刚初始化, 1识别中, 2.获得结果, 3识别出错, -1 未知错误

robokitVersion

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

msg = r.robokitVersion()

print(msg)

接口返回数据示例：

Output: ""3.3.5.11""

getArmlInfo

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getArmlInfo()

接口返回数据示例:

output:

```
{  
  
  "DI": [],  
  "DO": [],  
  "arm_mode": 0,  
  "enabled": false,  
  "json_string": "",  
  "robot_status_bits": [],  
  "safety_mode": 0,  
  "safety_status_bits": [],  
  "taskId": 0,  
  
  "task_status": 0  
  
}
```

返回数据解释:

返回值

数据类型

描述

DI

list

当前DI状态

DO

list

当前DO状态

arm_mode

int

机器臂的模式

enabled

bool

使能状态

safety_mode

int

安全模式状态

taskId

int

当前执行的taskId

task_status

int

当前任务状态

getMinDynamicObs

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getMinDynamicObs()

接口返回数据示例：

output：

[0, 0]

返回数据解释：

返回值

数据类型

描述

0

int

坐标x，单位为m

0

int

坐标y，单位为m

getRobotFile

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getRobotFile()

接口返回数据示例：

output：

模型文件的原始数据以字典类型返回，由于返回的数据量较多，详情请见下方txt文件

getRobotFile.txt

返回数据解释：

返回值

数据类型

描述

参考上面的txt文件

dict

所有模型文件的数据

getRecFile

入口参数解释：

字段

数据类型

示例值

描述

name

"shelf/s0001.shelf"

识别文件名称

方法调用实例：

r.getRecFile("shelf/s0001.shelf")

接口返回数据示例：

output：

获得识别文件的原始数据以字典类型返回，由于返回的数据量较多，详情请见下面的txt文件

getRecFile.txt

返回数据解释：

返回值

数据类型

描述

参考上面的txt文件

dict

识别文件的原始数据

noticeExits

入口参数解释：

字段

数据类型

示例值

描述

code

int

533000

Notice 编号

方法调用实例：

r.noticeExits(533000)

接口返回数据示例：

output：

True

返回数据解释：

返回值

数据类型

描述

True

bool

所要查询的Notice存在

getLM

入口参数解释：

字段

数据类型

示例值

描述

name

str

"LM4"

站点或者库位名称

flag

bool

True

True 返回的坐标是地图坐标系， False返回的坐标是机器人坐标系

方法调用实例：

r.getLM("LM7", True)

接口返回数据示例：

output：

(0, 0, 0, -1)

返回数据解释:

返回值

数据类型

描述

0

int

x

0

int

y

0

int

theta

-1

int

id(-1表示不存在)

RecognizeBarCode

入口参数解释：

字段

数据类型

示例值

描述

name

str

"tag/t0001.tag"

其中输入为识别文件的目录

id

int

123456

其中输入为识别任务的唯一id

方法调用实例：

r.RecognizeBarCode("tag/t0001.tag", "123")

接口返回数据示例：

output：

{

```
"barCode":"string",  
"id":"string",
```

"status": 0

}

返回数据解释:

返回值

数据类型

描述

barCode

str

表示识别的结果

id

str

为当前识别的任务 id

status

int

0 表示成功, 1表示识别中, 2表示识别失败

getCanFrame

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getCanFrame()

接口返回数据示例：

output:

{

```
'Canerror': [],
'Channel': 0,
'DLC': 0,
'Data': '',
'Direction': False,
'Extended': False,
'ID': 0,
'Remote': False,
```

'Timestamp': 0

}

返回数据解释：

返回值

数据类型

描述

{}

dict

获取当前的 CanFrame

isAnyErrorExists

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.isAnyErrorExists()

getCurrentTaskStatus

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例:

r.getCurrentTaskStatus()

getNearestLaserPoint

入口参数解释:

字段

数据类型

示例值

描述

laser_id

int

0

激光 id 号

方法调用实例：

r.getNearestLaserPoint(0)

binDetection

入口参数解释：

字段

数据类型

示例值

描述

seq

int

26446717

时间戳

方法调用实例：

r.binDetection(26446717)

getBinDetectionResult

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getBinDetectionResult()

接口返回数据示例：

output：

```
{  
  
  "bins": []  
}
```

返回数据解释：

返回值

数据类型

描述

bins

list

库位状态

getForkTipObsDist

入口参数解释：

字段

数据类型

示例值

描述

json_str

str

json.dumps({"taskID":1})

json格式的taskID

方法调用实例：

r.getForkTipObsDist(json.dumps({"taskID":1}))

接口返回数据示例：

output：

2

返回数据解释:

返回值

数据类型

描述

2

float

UNKNOWN_ERROR=-1, 未知错误 未开启 triggerOn, 模型文件改变中

INIT=0, 初始化中

RUNNING=1, 识别运行中

SUCCESS=2, 检测成功

FAIL=3, 检测失败

NO_AVAILABLE_DATA=4

getMoveStatus

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getMoveStatus()

接口返回数据示例：

output：

{

```
'actual_reach_angle': 0,
'actual_reach_dist': 0,
'advance_regions': [],
'area_name': ['6'],
'block_id': 0,
'block_reason': 1,
'block_x': -10.303100000000004,
'block_y': 2.9848326895455,
'blocked': False,
'closest_label': '',
'closest_target': '',
'containers': [],
'dist2goal': 0,
'finished_path_name': [],
'info': '',
'modules': [],
'nearest_obstacles': [],
'removed_regions': [],
'running_status': 0,
'slow_id': 0,
'slow_reason': 0,
'slow_x': 0,
'slow_y': 0,
'slowed': False,
'target_angle': 0,
'target_dist': 0,
'target_label': '',
'target_name': 'LM1',
'target_x': 0,
'target_y': 0,
'task_id': '',
'task_status': 2,
'task_type': 100,
```

'unfinished_path_name': []

}

返回数据解释:

返回值

数据类型

描述

{}

dict

运行状态、障碍物状态

getLastMoveStatus

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getLastMoveStatus()

接口返回数据示例：

output：

{

```
'actual_reach_angle': 0,  
'actual_reach_dist': 0,  
'advance_regions': [],  
'area_name': ['6'],  
'block_id': 0,  
'block_reason': 1,  
'block_x': -10.303100000000004,  
'block_y': 2.9848326895455,  
'blocked': False,  
'closest_label': '',  
'closest_target': '',  
'containers': [],  
'dist2goal': 0,  
'finished_path_name': [],  
'info':  
'{"currentBlockId":"","info":"","objectFile":"","require":null}',  
'modules': [],  
'nearest_obstacles': [],  
'removed_regions': [],
```

'robot_region': {

```
'name': '',
```

'point': [{

```
'x': -5.595058234040283,
```

'y': 4.251778463384683

}, {

```
'x': -6.531726966448368,
```

'y': 2.7901534479552708

}, {

```
'x': -5.822805155969355,
```

'y': 2.3358475305084925

```
}, {
```

```
    'x': -4.886136423561271,
```

```
'y': 3.7974725459379055
```

```
}}
```

```
},
```

```
'robot_shape': {
```

```
    'head': 1.367,  
    'radius': 0,  
    'shape': 1,  
    'tail': 0.369,
```

```
'width': 0.842
```

```
    },  
    'running_status': 0,  
    'slow_id': 0,  
    'slow_reason': 0,  
    'slow_x': 0,  
    'slow_y': 0,  
    'slowed': False,  
    'target_angle': 0,  
    'target_dist': 0,  
    'target_label': '',  
    'target_name': 'LM1',  
    'target_x': 0,  
    'target_y': 0,  
    'task_id': '',  
    'task_status': 5,
```

```
'task_status_package': {
```

```
        'closest_label': '',
        'closest_target': 'LM1',
        'distance': -1,
        'info': '',
        'percentage': 0,
        'source_label': '',
        'source_name': '',
        'target_label': '',
        'target_name': '',
```

'task_status_list': [{

```
        'status': 5,
        'task_id': '',
```

'type': 100

}}

```
    },
    'task_type': 100,
```

'unfinished_path_name': []

}

返回数据解释：

返回值

数据类型

描述

{}

dict

上一个周期的运行状态、障碍物状态

getCurrentPathProperty

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getCurrentPathProperty()

接口返回数据示例：

output：

{

```
'3DCameraHole': -1,
'3DCameraObstacle': -1,
'HoldDir': 999.0,
'collisionGroupId': -1,
'collisionPointThreshold': -1,
'collisionPointThresholdCamera': -1,
'decObsExpansion': -1.5,
'direction': 'Forward',
'emergencyStopDist': -1.0,
'fallingdown': -1,
'forbidDNodeGroupId': -1,
'forbidLaserGroupId': -1,
'forbiddenRotAngle': 999.0,
'goAngle': -1.0,
'goodsCollision': -1,
'goodsDir': 999.0,
'infrared': -1,
'instanceName': '',
'laser': -1,
'laserDetectionWidth': '',
'length': -99.0,
'loadDecObsExpansion': -1.5,
'loadHoldDir': 999.0,
'loadMaxAcc': -1.0,
'loadMaxDec': -1.0,
'loadMaxRot': -1.0,
'loadMaxRotDec': -1.0,
'loadMaxSpeed': -1.0,
'loadObsDecDist': -1.5,
'loadObsExpansion': -1.5,
'loadObsStopDist': -1.5,
'loadStopAcc': -1.0,
'load_max_rot_acc': -1.0,
'maxacc': -1.0,
'maxdec': -1.0,
'maxrot': -1.0,
'maxrotacc': -1.0,
'maxrotdec': -1.0,
'maxspeed': -1.0,
'obsDecDist': -1.5,
'obsDecSpeed': -1.5,
'obsExpansion': -1.5,
'obsStopDist': -1.5,
'parkRequired': False,
'reachAngle': -1.0,
'reachDist': -1.0,
'relaxGoalReachDist': -1.0,
'safeState': -1,
'soundName': '',
'stopAcc': -1.0,
'targetObsDist': -1.0,
'ultrasonic': -1,
'unLoadHoldDir': 999.0,
'virtualLaser': -1,
```

'width': -1.0

}

返回数据解释:

返回值

数据类型

描述

{}

dict

机器人运行时，当前路线上的属性

getCurrentAdvancedArea

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.getCurrentAdvancedArea()

接口返回数据示例：

output:

{

```
'3DCameraHole': -1,  
'3DCameraObstacle': -1,  
'FG_RealTimeAvoid': -1,  
'class_name': '',  
'collisionGroupId': -1,  
'collisionPointThreshold': -1,  
'collisionPointThresholdCamera': -1,  
'decObsExpansion': -1.5,  
'emergencyStopDist': -1.0,  
'fallingdown': -1,  
'forbidDNodeGroupId': -1,  
'forbidLaserGroupId': -1,  
'forbidden': 0,  
'forbiddenRotAngle': 999.0,  
'id': -1,  
'infrared': -1,  
'laser': -1,  
'laserDetectionWidth': '',  
'loadDecObsExpansion': -1.5,  
'loadMaxAcc': -1.0,  
'loadMaxDec': -1.0,  
'loadMaxRot': -1.0,  
'loadMaxRotAcc': -1.0,  
'loadMaxRotAec': -1.0,  
'loadMaxSpeed': -1.0,  
'loadObsDecDist': -1.0,  
'loadObsExpansion': -1.5,  
'loadObsStopDist': -1.0,  
'loadStopAcc': -1.0,  
'maxSpeed': -1.0,  
'maxacc': -1.0,  
'maxdec': -1.0,  
'maxrot': -1.0,  
'maxrotacc': -1.0,  
'maxrotdec': -1.0,  
'modbusDoGroupId': -1,  
'obsDecDist': -1.5,  
'obsDecSpeed': -1.5,  
'obsExpansion': -1.5,  
'obsStopDist': -1.5,  
'openSound': -1,  
'shieldLaser': -1,  
'shieldVLaser': -1,  
'soundName': '',  
'stopAcc': -1.0,  
'ultrasonic': -1,
```

'virtualLaser': -1

}

返回数据解释：

返回值

数据类型

描述

{}

dict

机器人运行时，当前所在高级区域的属性

setDO

入口参数解释：

字段

数据类型

示例值

描述

id

int

1

DO的id

status

bool

True

是否打开这个DO

方法调用示例：

r.setDO(1, True)

setMotorSpeed

入口参数解释：

字段

数据类型

示例值

描述

name

str

"motor0"

电机名称

vel

float

0.5

电机速度

stopDI

int

3

到位DI

方法调用示例：

```
r.setMotorSpeed("motor0", 0.5, 3)
```

setMotorPosition

入口参数解释：

字段

数据类型

示例值

描述

motor_name

str

"motor0"

模型文件中的电机名称

pos

float

1.2

发送目标点位置也可能是角度

maxVel

float

0.5

运行过程中的最大速度不能超过模型文件中的最大速度

stopDI

int

3

如果这个StopDI触发则表示运动到位

方法调用示例:

```
r.setMotorPosition("motor0", 1.2, 0.5, 3)
```

resetMotor

入口参数解释:

字段

数据类型

示例值

描述

motor_name

str

"motor0"

电机名称/

方法调用示例:

r.resetMotor("motor0")

publishSpeed

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例:

r.publishSpeed()

setPathOnRobot

入口参数解释：

字段

数据类型

示例值

描述

x

list

/

线路的x坐标

y

list

线路的y坐标

angle

float

终点的朝向

方法调用示例：

x_list = [1.1, 2.2]

y_list = [3.3, 4.4]

angle = 1.2

r.setPathOnRobot(x_list, y_list, angle)

setPathOnWorld

入口参数解释：

字段

数据类型

示例值

描述

x

list

/

线路的x坐标

y

list

线路的y坐标

angle

float

终点的朝向

方法调用示例：

x_list = [1.1, 2.2]

y_list = [3.3, 4.4]

angle = 1.2

r.setPathOnWorld(x_list, y_list, angle)

stopRobot

入口参数解释：

字段

数据类型

示例值

描述

flag

bool

True

如果是True就是急停，如果是False则以StopAcc停下来

方法调用示例：

r.stopRobot(True)

setSound

入口参数解释：

字段

数据类型

示例值

描述

name

str

"music.mp3"

音频名称

flag

bool

False

是否循环播放

方法调用示例：

r.setSound("music.mp3",True)

setSoundCount

入口参数解释：

字段

数据类型

示例值

描述

name

str

"music.mp3"

音频名称

count

int

5

播放次数，需要大于0

方法调用示例：

r.setSoundCount("music.mp3",5)

stopSound

入口参数解释：

字段

数据类型

示例值

描述

flag

bool

True

如果为True则为停止播放音乐

方法调用示例：

r.stopSound(True)

doRec

入口参数解释：

字段

数据类型

示例值

描述

filename

str

"s0001.shelf"

识别文件

方法调用示例：

r.doRec("s0001.shelf")

doRecWithAngle

入口参数解释：

字段

数据类型

示例值

描述

filename

str

"s0001.shelf"

识别文件

a

float

20

识别机构在agv坐标系下的角度

方法调用示例：

```
r.doRecWithAngle("s0001.shelf", 20)
```

doRecWithRegion

入口参数解释：

字段

数据类型

示例值

描述

filename

str

"s0001.shelf"

识别文件

x

float

20

机器人坐标系下识别范围中心X坐标，单位 m

y

float

1.5

机器人坐标系下识别范围中心Y坐标，单位 m

theta

float

90

机器人坐标系下识别范围中心角度，单位 rad

radius

float

1.5

识别范围半径，单位 m

方法调用示例：

```
r.doRecWithRegion("plt/plt001.plt", 0.0, 0.0, 1.57, 1.5)
```

recTargetObs

入口参数解释：

字段

数据类型

示例值

描述

x

float

1.0

目标点世界坐标系下 x 坐标 m

y

float

1.0

目标点世界坐标系下 y 坐标 m

theta

float

1.57

目标点世界坐标系下朝向 rad

filename

str

"s0001.shelf"

识别文件名称 palletObject

方法调用示例：

recTargetObs(targetRobot[0], targetRobot[1], targetRobot[2], self.recFile)

goMapPath

入口参数解释：

注意，goMapPath 类似于插入了一个导航任务，如果需要在脚本上结束这个导航，需要满足脚本 Modlue 类下的status为 FINISHED,并且任务参数的script_stage 必须为3才可以。

字段

数据类型

示例值

描述

task

str

self.task = r.moveTask()

获取moveTask

方法调用实例：

self.task = r.moveTask()

status = r.goMapPath(json.dumps(self.task))

r.setNotice(status)

接口返回数据示例：

output:

1

返回数据解释：

返回值

数据类型

描述

0

int

任务状态： NONE

1

int

任务状态: RUNNING

2

int

任务状态: NEARTOGOAL

3

int

任务状态: FINISHED

4

int

任务状态: FAILED

5

int

任务状态: SUSPENDED

forkGoods

入口参数解释:

字段

数据类型

示例值

描述

load

bool

True

设置货叉上的货物检测DI

recfile

str

"shelf/s0001.shelf"

货物模型文件名称

方法调用实例：

```
r.forkGoods(True, "shelf/s0001.shelf")
```

armBinTask

入口参数解释：

字段

数据类型

示例值

描述

task_id

int

0

任务的id

cmd

str

任务的详细动作序列

方法调用实例：

r.armBinTask(1, "[]")

armStop

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.armStop()

armPause

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.armPause()

armResume

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.armResume()

armControl

入口参数解释：

字段

数据类型

示例值

描述

json_str

str

'{"type": "set_speed_slider"}'

控制指令，是json 字符串

方法调用实例：

data = {"type": "set_speed_slider"} r.armControl(json.dumps(data))

stopMotor

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.stopMotor()

scannerCode

入口参数解释：

字段

数据类型

示例值

描述

task_id

int

0

任务ID

方法调用实例：

r.scannerCode(1)

clearNotice

入口参数解释：

字段

数据类型

示例值

描述

code

int

533000

Notice的编号

方法调用实例：

stopCurrentBlock

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例:

r.stopCurrentBlock()

sendCanFrame

入口参数解释:

字段

数据类型

示例值

描述

channel

int

1

使用的端口，有1和2,对应控制器的CAN通道

can_id

int

0x601

发送报文的canid

dlc

int

8

发送报文的数据长度，一般为8

extend

bool

False

报文是否为扩展型，一般为False

can_string

str

"40 40 60 00 00 00 00 00"

报文数据区

方法调用实例：

r.stopCurrentBlock()

resetGoForkPath

入口参数解释：

字段

数据类型

示例值

描述

x

float

0

终点x坐标 m

y

float

0

终点y坐标 m

yaw

float

0

终点角度坐标 rad

back_dist

float

0

到终点后的后退距离

min_ahead_dist

float

0

栈板前直线距离 m

ahead_dist

float

0

到终点前的直线距离

方法调用实例：

r.resetGoForkPath(0.0,0.0,0.0,0.0,0.0,0.0)

goForkPath

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.goForkPath()

setGoForkForkPos

入口参数解释：

字段

数据类型

示例值

描述

x

float

0

货叉相对于里程中心的 x 轴坐标 m

y

float

0

货叉相对于里程中心的 y 轴坐标 m

theta

float

0

是货叉相对于里程中心的偏移角度 rad

hold_dir

float

0

是车体的横移角度 单位：°

方法调用实例：

```
r.setGoForkForkPos(0.0,0.0,0.0,0.0)
```

setMotorCalib

入口参数解释：

字段

数据类型

示例值

描述

motor_name

str

motor1

电机名称

方法调用实例：

r.setMotorCalib("motor")

setVirtualDI

入口参数解释：

字段

数据类型

示例值

描述

index

index

1

列表中相对第一个虚拟 DI 的索引

status

bool

True

虚拟 DI 状态

方法调用实例：

r.setVirtualDI(1,True)

enableMotor

入口参数解释：

字段

数据类型

示例值

描述

name

str

"lift"

电机名称

方法调用实例：

r.enableMotor("lift")

disableMotor

入口参数解释：

字段

数据类型

示例值

描述

name

str

"lift"

电机名称

方法调用实例：

r.disableMotor("lift")

setSteerAngle

入口参数解释：

字段

数据类型

示例值

描述

name

str

"steer1"

舵机名称

angle

float

1.57

角度位置, 单位rad

方法调用实例:

r.setSteerAngle("steer1", 1.57)

接口返回数据示例:

output:

True

返回数据解释:

返回值

数据类型

描述

True

bool

电机到位

updateModelParamsByJsonStr

入口参数解释：

字段

数据类型

示例值

描述

model_json_str

str

robotFile = r.getRobotFile()

json.dumps(robotFile)

机器人模型文件的json格式，先从r.getRobotFile()获取，更新时需要在模型名称后面带个时间戳用于区分，比如"model": "lidar-202507071331"

方法调用实例：

robotFile = r.getRobotFile

r.updateModelParamsByJsonStr(json.dumps(robotFile))

接口返回数据示例：

output：

True

返回数据解释：

返回值

数据类型

描述

True

bool

推送成功

triggerCameraOn

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.triggerCameraOn()

triggerCameraOff

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.triggerCameraOff()

loadStatus

入口参数解释：

字段

数据类型

示例值

描述

dist

float

1.0

需要屏蔽的点云距离叉尖相机的范围

方法调用实例：

r.loadStatus(1.0)

setDIValid

入口参数解释：

字段

数据类型

示例值

描述

id

int

1.0

DI 的序号

valid

bool

True

true表示生效，为 false 表示不生效

方法调用实例：

r.setDValid(1, True)

addDisableDepthId

入口参数解释：

字段

数据类型

示例值

描述

id

list

1

深度相机的id列表

方法调用实例：

r.addDisableDepthId(1)

addDisableLaserId

入口参数解释：

字段

数据类型

示例值

描述

id

list

1

深度相机的id列表

方法调用实例：

r.addDisableLaserId(1)

clearDisableDepthId

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.clearDisableDepthId()

clearDisableLaserId

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用实例：

r.clearDisableLaserId()

setLocalShelfArea

入口参数解释：

字段

数据类型

示例值

描述

object_model_path

str

"shelf/s0001.shelf"

货架模型文件名称

方法调用示例：

object_model_path = "shelf/s0001.shelf" r.setLocalShelfArea(object_model_path)

resetLocalShelfArea

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

r.resetLocalShelfArea()

logInfo

入口参数解释:

字段

数据类型

示例值

描述

ss

str

" "

输入的字符串

方法调用示例:

self.state = dict()

self.state["key"] = "value"

r.logInfo(json.dumps(self.state))

logWarn

入口参数解释:

字段

数据类型

示例值

描述

ss

str

" "

输入的字符串

方法调用示例：

self.state = dict()

self.state["key"] = "value"

r.logWarn(json.dumps(self.state))

logDebug

入口参数解释：

字段

数据类型

示例值

描述

ss

str

输入的字符串

方法调用示例：

self.state = dict()

self.state["key"] = "value"
r.logDebug(json.dumps(self.state))

setError

入口参数解释：

字段

数据类型

示例值

描述

ss

str

" "

注释字符串

方法调用示例：

r.setError(" ")

setUserError

入口参数解释：

字段

数据类型

示例值

描述

code

int

53900~53999

报错码， 如果超过这个范围，则会报notice

ss

str

" "

注释字符串

方法调用示例：

r.setUserError(53900, " ")

setPickRobotError

入口参数解释：

字段

数据类型

示例值

描述

code

int

53800~53899

报错码， 如果超过这个范围，则会报notice

ss

str

" "

注释字符串

方法调用示例：

r.setPickRobotError(53800, " ")

setWarning

入口参数解释：

字段

数据类型

示例值

描述

ss

str

" "

注释字符串

方法调用示例：

r.setWarning(" ")

setUserWarning

入口参数解释：

字段

数据类型

示例值

描述

code

int

55900~55999

报错码， 如果超过这个范围， 则会报notice

ss

str

" "

注释字符串

方法调用示例:

r.setUserWarning(55900, " ")

setPickRobotWarning

入口参数解释:

字段

数据类型

示例值

描述

code

int

55800~55899

报错码， 如果超过这个范围，则会报notice

ss

str

" "

注释字符串

方法调用示例:

r.setPickRobotWarning(55800, " ")

setNotice

入口参数解释:

字段

数据类型

示例值

描述

ss

str

" "

注释字符串

方法调用示例:

r.setNotice(" ")

clearNotice

入口参数解释:

字段

数据类型

示例值

描述

code

int

\

Notice的编号

方法调用示例：

r.clearNotice(51900)

clearError

入口参数解释：

字段

数据类型

示例值

描述

code

int

\

Error的编号

方法调用示例：

r.clearError(53900)

clearWarning

入口参数解释:

字段

数据类型

示例值

描述

code

int

\

Warning的编号

方法调用示例:

r.clearWarning(55900)

setNextSpeed

入口参数解释:

字段

数据类型

示例值

描述

nav

str

/

下发的速度，格式与从getNextSpeed或者navSpeed获得的格式相同

方法调用示例：

```
speed_info = r.navSpeed()

speed_info[rotate] = 1.3

speed_str = json.dumps(speed_info, ensure_ascii=False)

r.setNextSpeed(speed_str)
```

接口返回数据示例：

Output: True

返回字段解释：

字段

数据类型

示例值

描述

/

bool

True

如果成功转成下发速度则返回True

switchMap

入口参数解释：

字段

数据类型

示例值

描述

map

str

/

地图名称

switchPoint

str

重定位点位

center_x

float

重定位中心点 x 坐标 单位 m

center_y

float

重定位中心点 y 坐标 单位 m

initial_angle

float

重定位中心朝向 单位 degree

方法调用示例：

map_name = "example.smap"

switch_point = ""

center_x = 1.5

center_y = 2.1

initial_angle = 32.5

r.switchMap(map_name, switch_point, center_x, center_y, initial_angle)

接口返回数据示例：

Output: 0

返回字段解释：

字段

数据类型

示例值

描述

/

int

0

2没有进行切换，1切换中，0切换成功，-1不存在地图，-2切换失败

setGoodsShape

入口参数解释：

字段

数据类型

示例值

描述

head

float

2

货物头部长度

tail

float

2

货物的尾部长度

width

float

3.5

货物的宽度

方法调用示例：

```
r.setGoodsShape(head=2, tail=2, width=3.5)
```

setBlockReason

入口参数解释：

字段

数据类型

示例值

描述

collision_type

int

CollisionType.Ultrasonic

阻挡原因见rbk.py脚本中的CollisionType类

x

float

2

障碍物位置

y

float

3.5

障碍物位置

id

int

2

障碍物id

方法调用示例：

r.setBlockReason(collision_type=CollisionType.Ultrasonic, x=2, y=3.5, id=2)

setContainer

入口参数解释：

字段

数据类型

示例值

描述

container_name

str

"container_name"

库位或者背篓名称

goods_id

str

2

货物的id

desc

str

""

描述

方法调用示例：

r.setContainer(container_name="container_name", goods_id=2, desc="")

clearContainer

入口参数解释：

字段

数据类型

示例值

描述

container_name

str

"container_name"

库位或者背篓名称， container_name如果为All则全部清除

方法调用示例：

```
r.clearContainer(container_name="All")
```

接口返回数据示例：

Output: False

返回字段解释：

字段

数据类型

示例值

描述

/

bool

False

如果没有库位或者背篓， 则返回false

clearContainerByGoodsId

入口参数解释：

字段

数据类型

示例值

描述

goods_id

str

"container_name"

库位或者背篓名称， goods_id如果为All则全部清除

方法调用示例：

r.clearContainer(goods_id="All")

接口返回数据示例：

Output: True

返回字段解释：

字段

数据类型

示例值

描述

/

bool

True

如果没有库位或者背篓， 则返回false

setPickRobotError

入口参数解释：

字段

数据类型

示例值

描述

code

int

53800

报错码

ss

str

"error"

注释字符串

方法调用实例：

setPickRobotWarning

入口参数解释：

字段

数据类型

示例值

描述

code

int

55800

报错码

ss

str

"warning"

注释字符串

方法调用实例：

r.setPickRobotWarning(55800, "warning")

setObsStopDist

入口参数解释：

字段

数据类型

示例值

描述

dist

float

0.05

避障距离，单位 m

方法调用实例：

r.setObsStopDist(0.05)

addMoveTask

入口参数解释：

字段

数据类型

示例值

描述

task

json

/

任务运行导航的任务

方法调用示例(导航到某一个站点)：

task = {

"id": "LM1"

}

r.addMoveTask(json.dumps(task))

返回字段解释:

字段

数据类型

示例值

描述

无

int

0

/

addMoveTaskList

入口参数解释:

字段

数据类型

示例值

描述

task_list

json

/

任务运行导航的任务列表

方法调用示例：

task_list = [

{

 "source_id": "AP4",
 "id": "AP3",

 "task_id": "12345"

},

{

 "source_id": "AP3",
 "id": "AP2",

 "task_id": "123123"

}]

r.addMoveTaskList(json.dumps(task_list))

返回字段解释：

字段

数据类型

示例值

描述

无

int

0

/

clearBlockError

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

r.clearBlockError()

clearGData

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

r.clearGData()

clearGoodsShape

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

r.clearGoodsShap()

返回字段解释：

字段

数据类型

示例值

描述

无

bool

False

/

getContainers

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

c = r.getContainers()

print(c)

返回字段解释：

字段

数据类型

示例值

描述

c

list

/

当前车子上库位或者背篓货物的状态

返回值示例：

c = [{ 'container_name': '0', 'desc': '', 'goods_id': '1', 'has_goods': False}, { 'container_name': '1', 'desc':
'', 'goods_id': '123', 'has_goods': True}, { 'container_name': '2', 'desc': 'by script', 'goods_id': 'goods2',
'has_goods': True}]

getGData

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

dict_ = r.getGData()

返回字段解释：

字段

数据类型

示例值

描述

dict_

dict

{}

返回字典类型的变量

getForkPressureADC

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

p = r.getForkPressureADC()

print(p)

返回字段解释：

字段

数据类型

示例值

描述

p

float

0.6

货叉压力传感器adc值

getModbusData

入口参数解释：

字段

数据类型

示例值

描述

type

str

"01x"

modbus 类型

addr

int

202

读取时的寄存器起始地址

size

int

10

读取的数据长度

方法调用示例:

list_ = r.getModbusData("01x", 202, 10)

print(list_)

返回字段解释:

字段

数据类型

示例值

描述

list_

list

[]

寄存器数据

getRecFileFromTask

入口参数解释：

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

recfile = r.getRecFileFromTask()

print(recfile)

返回字段解释：

字段

数据类型

示例值

描述

recfile

str

"plt/p0001.plt"

识别文件

getRecResults

入口参数解释:

字段

数据类型

示例值

描述

index

int

3

获取第 index 个识别结果

方法调用示例:

result = r.getRecResults(3)

print(result)

返回字段解释:

字段

数据类型

示例值

描述

result

dict

{}

识别结果的结构体

返回值示例:

result = {

"x": 0.0, // 目标点地图坐标系或者相机坐标系下的x坐标 m "y": 0.0, // 目标点在地图坐标系或者相机坐标系下的y坐标 m "z": 0.0, // 目标点在地图坐标系或者相机坐标下的z坐标 m "yaw": 0.0, // 目标点朝外向量在地图坐标系或者相机坐标下xy平面的角度 rad

"type": "12345", //条形码的ID

}

getRecResultSize

入口参数解释:

字段

数据类型

示例值

描述

无

/

/

/

方法调用示例：

Num = r.getRecResultSize()

print(Num)

返回字段解释：

字段

数据类型

示例值

描述

Num

int

2

识别结果的个数

logError

入口参数解释:

字段

数据类型

示例值

描述

ss

str

""

输入的字符串

方法调用示例:

r.logError("123")

openSpeed

入口参数解释:

字段

数据类型

示例值

描述

`vx`

`float`

`0.6`

行走时的速度`vx`

`vy`

`float`

`0.6`

行走时的速度`vy`

`vw`

`float`

`0.2`

行走时的速度`vz`

方法调用示例:

`r.openSpeed(0.6, 0.6, 0.2)`