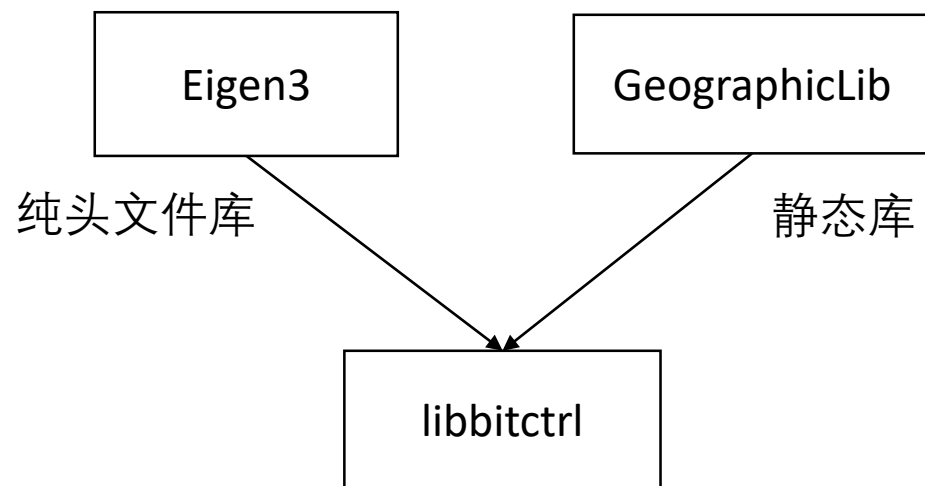


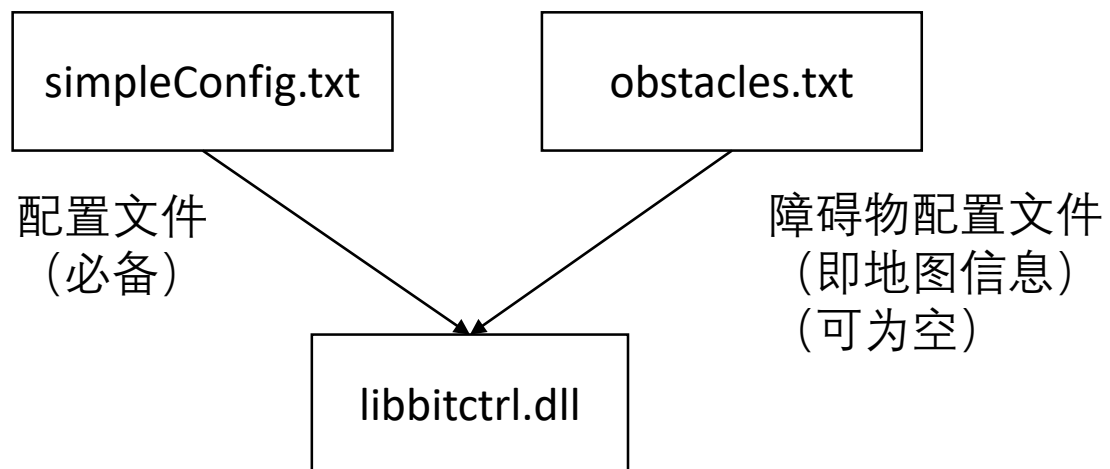
控制器程序说明

2019-07

库依赖



运行时依赖



基本使用流程

1. 设置相关目录（见目录设置）
2. 初始化控制器（见初始化与终止）
3. while 运行中
 1. 若需要，设置新任务（见任务设置）
 2. 更新本车状态、更新邻居状态
 3. 传入邻居通讯信息、人的信息等（见信息传递）
 4. 调用控制器计算函数（见计算与状态）
 5. 检查控制器状态（见计算与状态）
 6. 读取控制器要发送的信息（见信息传递）
 7. 将控制器的消息发给其他机器人
4. 终止控制器运行（见初始化与终止）

其余见：其他

目录设置

- ControllerSetConfigDir
设置配置文件simpleConfig.txt与obstacles.txt所在目录
- ControllerSetLogDir
设置要输出的log文件的所在目标

初始化与终止

- ControllerInit
传入当前机器人的编号，内部控制器初始化，
会读取配置文件simpleConfig.txt和obstacles.txt
如配置文件读取失败会直接挂掉
- GPSSetAnchor
手动设置当前的GPS原点信息
调用ControllerInit时，会从simpleConfig.txt中读出原点信息并覆盖现有
设置，故手动设置时，需在ControllerInit后调用此函数
- GPSGetAnchor
读取当前的GPS原点信息
- ControllerStop
终止控制器运行，再次调用相关函数前需再次进行初始化

任务设置

- **ControllerSetPath**
设置当前任务的路径信息
对于任务A, B, 需先调用此函数, 再调用SetFunction
- **ControllerSetTargetPoint**
设置当前任务的目标点信息
对于任务D, 需先调用此函数, 再调用SetFunction
- **ControllerSetFunction**
设置主任务与辅助任务, 主任务唯一, 辅助任务可叠加
主任务包括: A. 单车轨迹跟踪, B. 编队轨迹跟踪, C. 跟随人,
D. 原地转向目标
辅助任务有: a. 人工遥控干预, b. 故障检测, c. 演示出现故障
可以通过此函数取消当前主任务、辅助任务

信息传递

- **ControllerSetObstacles**
设置临时检测到的障碍物信息，即obstacles.txt中未列出的障碍物
不需要把其他车的位置传进来作为障碍，车车之间避碰会另行考虑
每次调用都会覆盖上次调用传入的数据
每次调用计算函数ControllerCompute都会清空传入的数据
- **ControllerSetHumanInput**
开启人为干预任务时，用此函数传入人的输入
- **ControllerSetHumanPosition**
开启跟随人任务时，用此函数传入当前人的位置
- **ControllerHandleMsg**
传入别的控制器发来的“控制器消息”
- **ControllerGetMsg**
读取控制器要发给别的车的“控制器消息”，确保buffer长度不小于80字节

- ControllerCompute

传入当前时间、本车基本状态、邻居车基本状态

计算得当前的控制量（线速度，角速度）

注意：传入的位置都是局部坐标，坐标转换函数见“其他”

- ControllerState

0 正常

1 正常, 但已到达目的地/完成预定任务, 可以停止当前任务

2 其他错误, 需要立即终止运行

3 检测出本机故障 (故障检测任务开启时)

- ControllerTaskProgress

返回当前任务的完成进度

0表示任务刚开始, 1表示任务已完成

其他

- **ControllerPause**
暂停、继续当前控制器的运行
控制器暂停后，调用计算函数等会返回0控制量
继续后，可以继续完成当前任务
- **GPS2Local**
将GPS坐标转换为局部坐标
- **GPSDistance**
计算两GPS坐标间距离