

KBSW183301 win32-使用轨道优先模式导航

本页介绍了virtual_track_with_oa的用法, 包含轨道优先模式的简单介绍。轨道优先模式, 可以理解为道路导航里面的高速优先, 1. 在起始点和目标点之间, 如果存在轨道, 会优先上轨道 2. 如果轨道上遇到障碍物, 会下轨道绕行后再上轨道继续前进。

本页内容

- 运行环境准备
 - 软件平台
 - 硬件平台
- 例程下载
- 编译运行
- 代码描述

运行环境准备

- 软件平台
 - Visual Studio 2010 SP1
 - Slamware Windows SDK:Slamware Windows SDK
 - RoboStudio(用于显示地图):Robostudio installer
 - Sample Code:

使用更高版本的Visual Studio可能会带来编译异常。

使用Visual Studio 2010 (无SP1) 可能会因为无法与.Net Framework兼容而报编译错误, 此时增加SP1更新包即可解决问题

- 硬件平台

(以下任选其一)

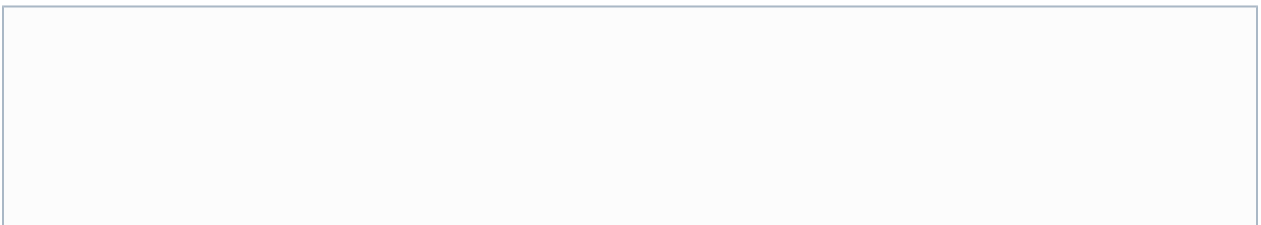
- Slamware SDP mini
- Slamware SDP
- Slamware 套装 (基于Slamware导航方案的用户机器人系统)
- Zeus/Apollo等底盘系统

例程下载

Win32-例程下载

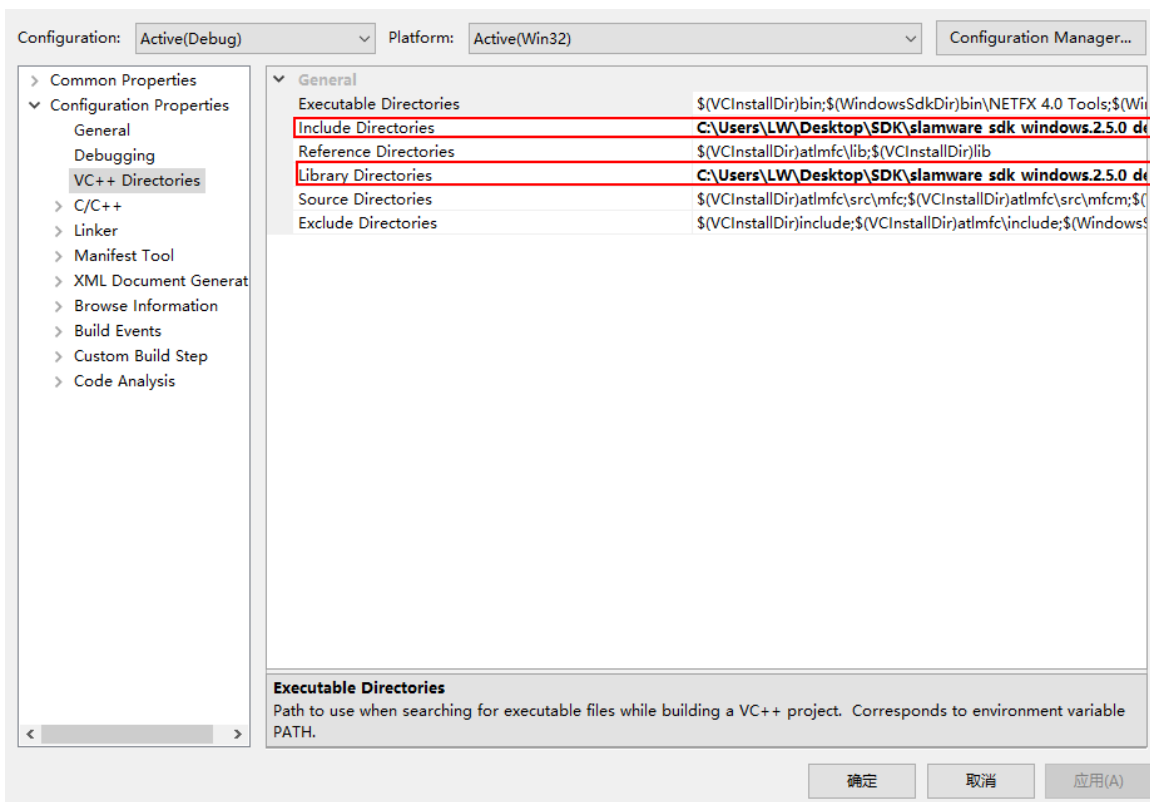
编译运行

1. 打开samples工程, 右键virtual_track_with_oa, 将此工程设置成StartUp project
2. 右键virtual_track_with_oa, 打开属性选项, 将Slamware SDK 的include目录和lib目录添加到工程

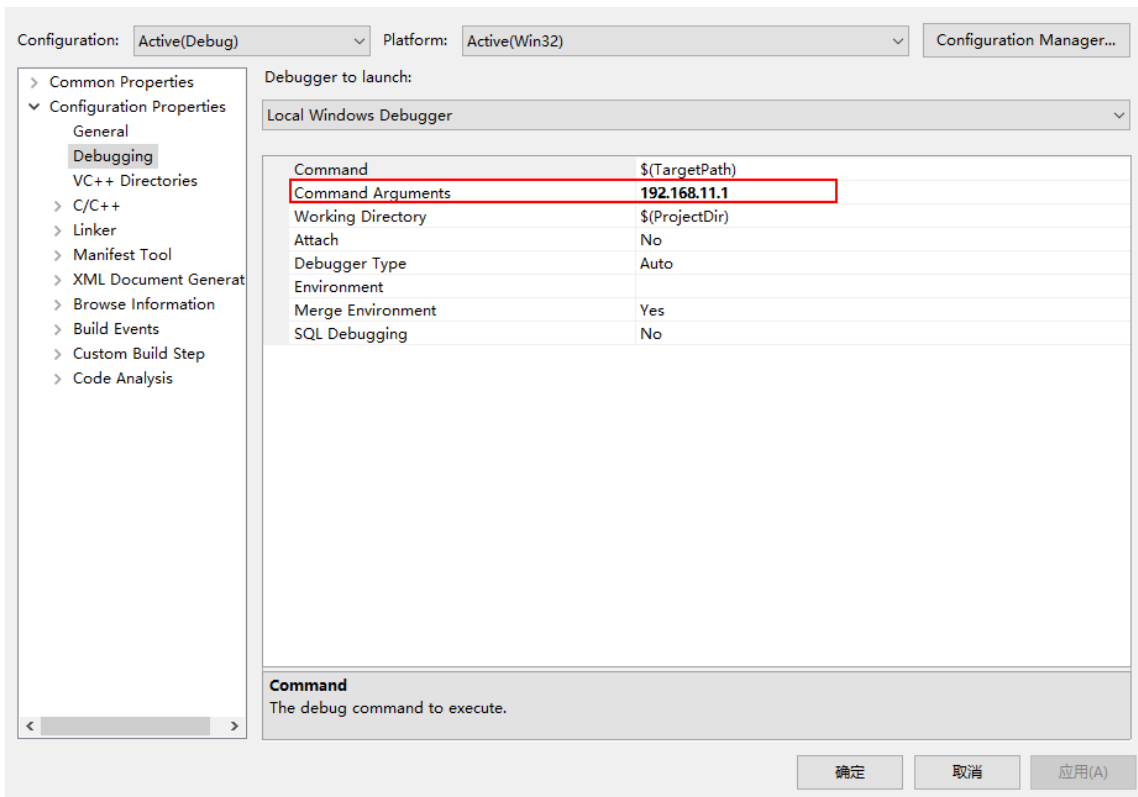


- Solution 'samples' (11 projects)
- > artifacts_demo
- > composite_map_demo
- > configure_network_demo
- > get_laser_scan
- > get_power_status
- > get_sensor_value
- > go_home_to_charge
- > move_to_spot
- > robot_health
- > rotation_action_demo
- > virtual_track_with_oa

Slamware SDK的include和lib目录无需复制到参考例程目录，只需在Visual Studio里指定路径即可。



3. 右键virtual_track_with_oa, 在Debugging页面中command Arguments处输入 192.168.11.1
格式说明：virtual_track_with_oa <IP address>



4. 点击F5运行
5. 可以连上Robostudio查看地图及机器人的运动
Your browser does not support the HTML5 video element

代码描述

- 画一条机器人当前位置x轴方向正向的，长6米的轨道，然后沿轨道走到目标点。

导航到目标点

```
SlamwareCorePlatform sdp =
SlamwareCorePlatform::connect(ip_address, 1445);
std::cout <<"SDK Version: " << sdp.getSDKVersion() << std::endl;
std::cout <<"SDP Version: " << sdp.getSDPVersion() << std::endl;

//draw a 6 meter virtual track
rpos::core::Pose sdp_pos = sdp.getPose();
rpos::core::Line line(rpos::core::Point(sdp_pos.x(),
sdp_pos.y()),rpos::core::Point(sdp_pos.x() + 6, sdp_pos.y()));
sdp.addLine(ArtifactUsageVirtualTrack, line);

rpos::actions::MoveAction action = sdp.getCurrentAction();
if (action)
    action.cancel();

rpos::features::motion_planner::MoveOptions options;
options.flag = MoveOptionFlag(MoveOptionFlagKeyPointsWithOA);
action = sdp.moveTo(rpos::core::Location(0, 0), options);
action.waitForDone();
if (action.getStatus() == rpos::core::ActionStatusError)
    std::cout << "Action Failed: " << action.getReason() <<
std::endl;
```