

KBSW180136 Win32 - Navigate to Target PointStandard Mode and Virtual Track Mode

This document introduces the demo project of "move_to_spot", including how to move in standard mode and virtual track mode.

Content

- [IDE Preparation](#)
 - [Software](#)
 - [Hardware](#)
 - [Download](#)
 - [Compiling](#)
 - [Code](#)
-

IDE Preparation

- Software
 - Visual Studio 2010 SP1
 - Slamware Windows SDK:[Slamware Windows SDK](#)
 - RoboStudio(for map display):[Robostudio installer](#)
 - Sample Code:

 Higher version of Visual Studio will cause errors. sometime you will need to upgrade SP1 package to make your VS compatable with .Net Framework.

- Hardware

Either one of following

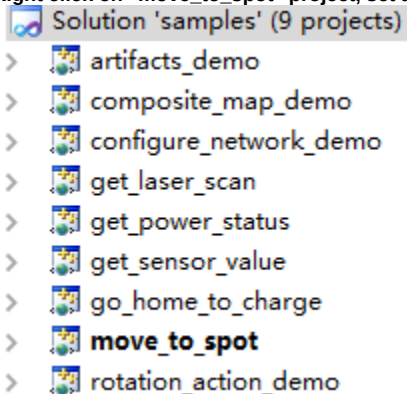
- Slamware SDP mini
- Slamware SDP
- Slamware Kit
- Zeus/Apollo robot base

Download

[Win32-Demo](#)

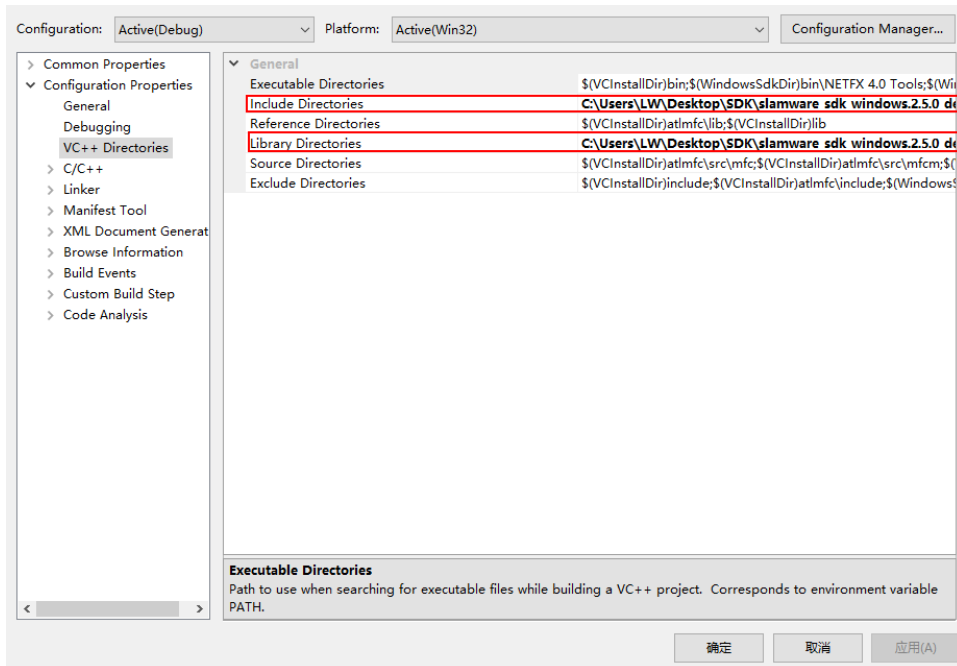
Compiling

1. Right click on "move_to_spot" project, set as StartUp project.

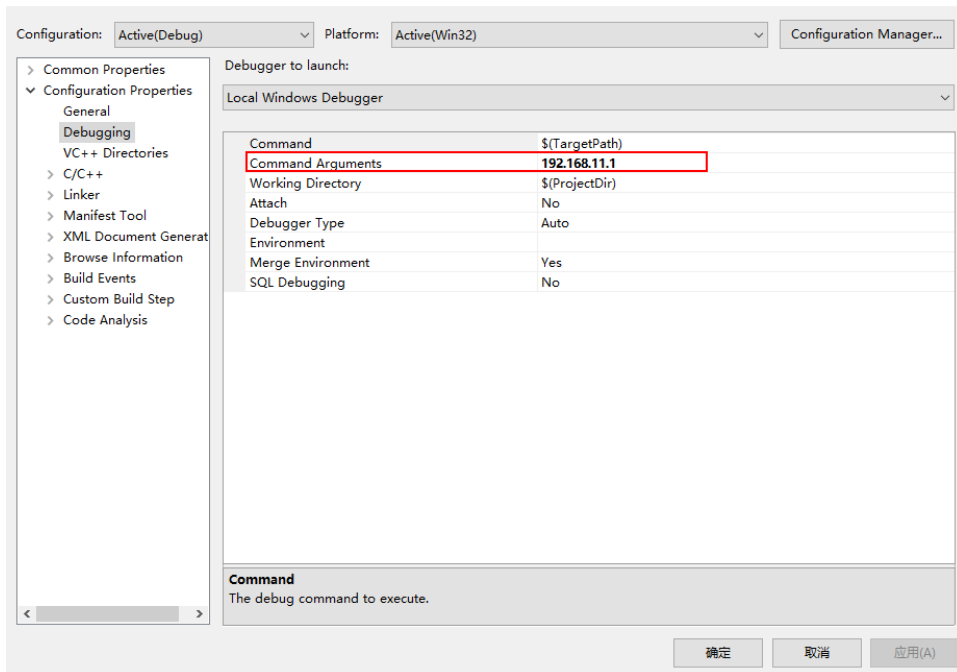


2. Right click on "move_to_spot",then " Properties"configure "include" and "lib" directories to the corresponding folder path of Slamware SDK.

 It's not necessary to copy files to the project directory, user will only need to configure the path of SDK.



3. Right click on "move_to_spot" set "Command Arguments" as follows:
Syntax Description move_to_spot <IP address>



4. Click " F5" to execute.
5. Robot motion and map could be seen on Robostudio.
Your browser does not support the HTML5 video element

Code

- Robot moves firstly to point(2, 0) in **standard mode**, and then back to point (0,0) in **virtual track mode**, if any obstacle is detected, it will be automatically avoided with the standard mode or it will stops on virtual tracks on virtual track mode.

```

SlamwareCorePlatform sdp = SlamwareCorePlatform::connect(argv[1], 1445);
std::cout << "SDK Version: " << sdp.getSDKVersion() << std::endl;
std::cout << "SDP Version: " << sdp.getSDPVersion() << std::endl;
rpos::actions::MoveAction action = sdp.getCurrentAction();
if (action)
    action.cancel();
//move to location (2, 0), not on virtual track
rpos::features::motion_planner::MoveOptions options;
options.flag = MoveOptionFlag(MoveOptionFlagMilestone | MoveOptionFlagPrecise);
action = sdp.moveTo(rpos::core::Location(2, 0), options);
action.waitUntilDone();
if (action.getStatus() == rpos::core::ActionStatusError)
    std::cout << "Action Failed: " << action.getReason() << std::endl;
//draw a virtual track from (0, 0) to (2, 0), then move to (0, 0) via virtual track
rpos::core::Line line(rpos::core::Point(0,0),rpos::core::Point(2,0));
sdp.addLine(ArtifactUsageVirtualTrack, line);
options.flag = MoveOptionFlag(MoveOptionFlagKeyPoints | MoveOptionFlagPrecise);
action = sdp.moveTo(rpos::core::Location(0, 0), options);
action.waitUntilDone();
if (action.getStatus() == rpos::core::ActionStatusError)
    std::cout << "Action Failed: " << action.getReason() << std::endl;

```