KBSW191012 Win32 - Virtual Track with OA Navigation

This document introduces the demo project of “virtual_track_with_oa”, including a brief introduction to Virtual Track with OA Navigation mode. It can be understood as high-speed priority in road navigation. 1. Between the starting point and the target point, if there is a track, it will take precedence on the track. 2. If there are obstacles on the track, it will go off the track and then go on the track.

Content

IDE Preperation

- ### 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

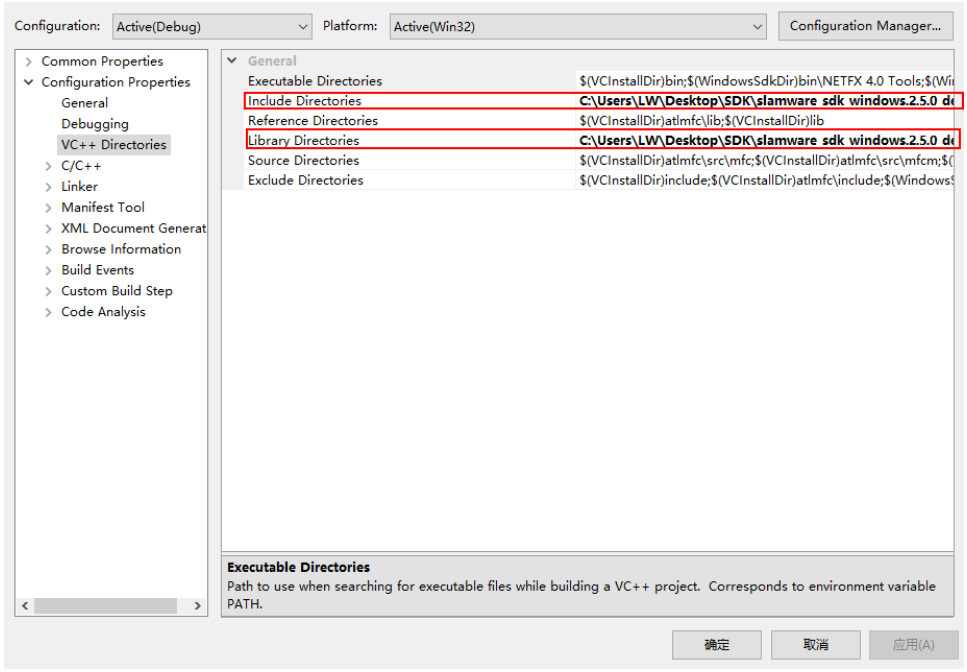
Win32-Demo

Compiling

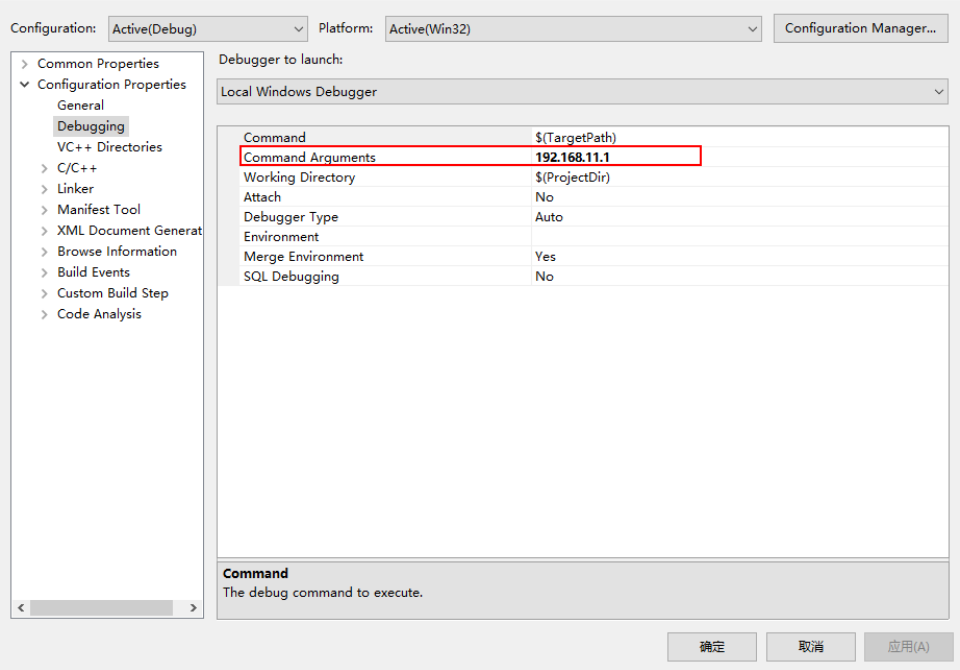
1. **Right click on "virtual_track_with_oa" project, set as StartUp project.**

2. **Right click on "virtual_track_with_oa", 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 "virtual_track_with_oa", then "properties"set "Command Arguments" as follows:**
   Syntax virtual_track_with_oa <IP address>



4. **Click " F5" to execute.**
5. **Robot's motion could be seen in Robostudio.**
   Your browser does not support the HTML5 video element

Code

- Draw a 6-metre track in the forward direction of the x-axis of the robot's current position, and then move along the track to the target point.

Virtual Track with OA Navigation

```
                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.waitUntilDone();
                if (action.getStatus() == rpos::core::ActionStatusError)
                        std::cout << "Action Failed: " << action.getReason() << std::endl;;
```