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
 - Software
 Hardware
- Download
- Compiling
- Code

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.
 - Solution 'samples' (11 projects)
 - > 🖉 artifacts_demo
 - > I composite_map_demo
 - > I configure_network_demo
 - > I get_laser_scan
 - > I get_power_status
 - > 🔅 get sensor value
 - > I go home to charge
 - > I move to spot
 - > 🔝 robot_health
 - > I rotation action demo
 - > 🖾 virtual_track_with_oa
- 2. Right click on "virtual_track_with_oa", then " Properties"configure "include" and "lib" directories to the corresponding folder path of Slamware SDK.

figuration: Active(Debug)	 Platform 	Active(Win32)		~	Configuration Manager
Common Properties Configuration Properties General Debugging VC++ Directories > C/C++ > Manifest Tool > XML Document Generat > Srowse Information > Build Events > Custom Build Step > Code Analysis	✓ General Executable Directorie Include Directories Reference Directories Source Directories Exclude Directories	es 	\$(VCInstallDir)bin; C:\User\LW\Desi \$(VCInstallDir)atIm C:\User\LW\Desi \$(VCInstallDir)atIm \$(VCInstallDir)inclu	i(WindowsSdk ttop\SDK\sla ttop\SDK\sla ttop\SDK\sla fe\src\mfc;\$(VCInstall	Dir)bin\NETFX 4.0 Tools;\$(Wi mware sdk windows.2.5.0 d allDir)lib mware sdk windows.2.5.0 d CInstallDir)atlmfc\src\mfcm;\$(Dir)atlmfc\include;\$(Windows)
	Executable Directories Path to use when search	ning for executable files	s while building a VC++ projec	t. Correspond	ds to environment variable

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

Configuration:	Active(Debug)	✓ Platform	n: Active(Win32))	~	Configuration Manager			
> Common P	roperties	Debugger to launch:							
 Configuration Properties General Debugging VC++ Directories C/C++ Linker Manifest Tool XML Document Generat Browse Information Build Events Custom Build Step Code Analysis 	Local Windows Debugg	~							
	Command Command Arguments Working Directory Attach Debugger Type Environment Merge Environment SQL Debugging		\$(TargetPath) 192.168.11.1 \$(ProjectDir) No Auto Yes No						
٢	>	Command The debug command to	o execute.		确定	取消 应用(A)			

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;</pre>
                std::cout <<"SDP Version: " << sdp.getSDPVersion() << std::endl;</pre>
                //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;;</pre>
```