# KBSW180137 Win32- Recover Localization

This document introduces the demo project of "recover localization", how to call the recoverlocalization() API to have the robot re-position to the correct localization on the map. Before using this feature, please note the following:

1. The success rate of relocalization is related to the environment. Before re-positioning, it is recommended to use a clear environmental map with a sharp outline.

2.For the case where there is a similar environment in the map area, for example, there are offices 1 and 2 with the same environment layout on the map, if the robot is actually in the office 1, it may relocate to the office 2 during the relocalization process. In this case, it is recommended to modify the office environment to make a distinction on the map when building the map (moving tables and chairs, adding pots, etc.).

Content

- IDE Preparation
  - ° Software <sup>o</sup> 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:



#### Hardware

Either one of following

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

### Download

Win32-Demo

Compiling

- 1. Right click on "recover\_localization" project, set as StartUp project.
  - 🗔 解决方案'samples' (11 个项目)
  - > I artifacts\_demo
  - > I composite\_map\_demo
  - > I configure\_network\_demo
  - > 3 get\_laser\_scan
  - > I get\_power\_status
  - get sensor value
    - 🗀 Header Files
    - Resource Files
    - > Source Files
    - 📴 外部依赖项
  - > I go\_home\_to\_charge
  - > I move\_to\_spot
  - > 🖉 recover localization
  - > I rotation action demo
  - > Speed\_regulation
- 2. Right click on "recover\_localization", then " Properties" configure "include" and "lib" directories to the corresponding folder path of Slamware SDK.

| onfiguration: Active(Debug)   | <ul> <li>Platform: Active(Win32)</li> </ul>  | <ul> <li>✓ Configuration Manager</li> </ul>   |  |  |  |  |
|---|--|---|--|--|--|--|
| <ul> <li>Common Properties</li> <li>Configuration Properties<br/>General<br/>Debugging</li> <li>VC++ Directories</li> <li>C/C++</li> <li>Linker</li> <li>Manifest Tool</li> <li>XML Document Generat</li> <li>Browse Information</li> <li>Build Events</li> <li>Custom Build Step</li> <li>Code Analysis</li> </ul> | <ul> <li>✓ General<br/>Executable Directories</li> <li>Include Directories</li> <li>Reference Directories</li> <li>Library Directories</li> <li>Source Directories</li> <li>Exclude Directories</li> </ul> | \$(VCInstallDir)bin;\$(WindowsSdkDir)bin\NETFX 4.0 Tools;\$(Wi<br>C:\Users\LW\Desktop\SDK\slamware sdk windows.2.5.0 d<br>\$(VCInstallDir)atmfc\ib;\$(VCInstallDir)lib<br>C:\Users\LW\Desktop\SDK\slamware sdk windows.2.5.0 d<br>\$(VCInstallDir)atmfc\rc\mfc;\$(VCInstallDir)atmfc\src\mfcm;\$<br>\$(VCInstallDir)atmfc\src\mfc;\$(VCInstallDir)atmfc\src\mfcm;\$<br>\$(VCInstallDir)include;\$(VCInstallDir)atmfc\src\mfc)<br>\$(VCInstallDir)include;\$(VCInstallDir)atmfc\src\mfc) |  |  |  |  |
| < >   | Executable Directories<br>Path to use when searching for executable fi<br>PATH.  | les while building a VC++ project. Corresponds to environment variable  |  |  |  |  |

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

| Configuration:  | Active(Debug)  | ~   | Platform:  | Active(Win3 | 32) |   | ~  | Configuratio | n Manager |
|---|----------------|---|--|-------------|-----|---|----|--------------|-----------|
| > Common F  | Properties     | Debugger to la  | unch:  |             |     |   |    |              |           |
| ✓ Configurat<br>Genera  | ion Properties | Local Windows Debugger  |  |             |     |   |    | ~            |           |
| <ul> <li>Common Properties</li> <li>Configuration Properties<br/>General<br/>Debugging<br/>VC++ Directories</li> <li>C/C++</li> <li>Linker</li> <li>Manifest Tool</li> <li>XML Document Generat</li> <li>Browse Information</li> <li>Build Events</li> <li>Code Analysis</li> </ul> |                | Command<br>Command<br>Working D<br>Attach<br>Debugger<br>Environmer<br>Merge Env<br>SQL Debug | Arguments<br>irectory<br>Type<br>nt<br>ironment<br>ironment<br>gging | xecute.     |     | \$(TargetPath)<br>192.168.11.1<br>\$(ProjectDir)<br>No<br>Auto<br>Yes<br>No |    |              |           |
|   |                |   |  |             |     |   | 确定 | 取消           | 应用(A)     |

- 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

• When the actual position deviates from the theoretical position of the map, the demo project will start re-positioning operation. If the recover is successful, you can see in the robostudio that the robot re-positions to the correct position and the status will turn to the corresponding one.

```
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>
                rpos::actions::MoveAction action = sdp.getCurrentAction();
                if (action)
                         action.cancel();
                //recover localization by giving an rectangle area; (0,0,0,0) represents the entire
map area.
                action = sdp.recoverLocalization(rpos::core::RectangleF(0,0,0,0));
                while(true)
                 {
                         switch (action.getStatus())
                                  {
                                 case rpos::core::ActionStatusError:
                                 std::cout << "Action Failed: " << action.getReason() << std::endl;</pre>
                                 break;
                                 case rpos::core::ActionStatusRunning:
                                 std::cout <<"Current status: Running" << std::endl;</pre>
                                 break;
                                 case rpos::core::ActionStatusFinished:
                                 std::cout <<"Current status: Finished" << std::endl;</pre>
                                 break;
                                 default :
                                 std::cout <<"Status Unknown" << std::endl;</pre>
                                 break;
                                 }
                 }
```