KBSW180140 Win32- Virtual Wall/Virtual Track Operation(Add, Delete, Modify

This document introduces the demo project of "artifacts_demo", explaining user operation of Virtual Wall/Virtual Track, including how to add/delete /modify artifacts.

Content

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

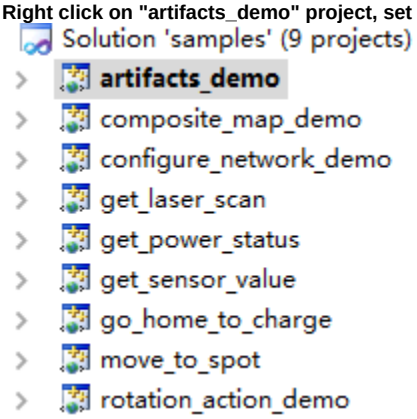
  - Slamware SDP mini
  - Slamware SDP
  - Slamware Kit
  - Zeus/Apollo robot base

Download

Win32-Demo

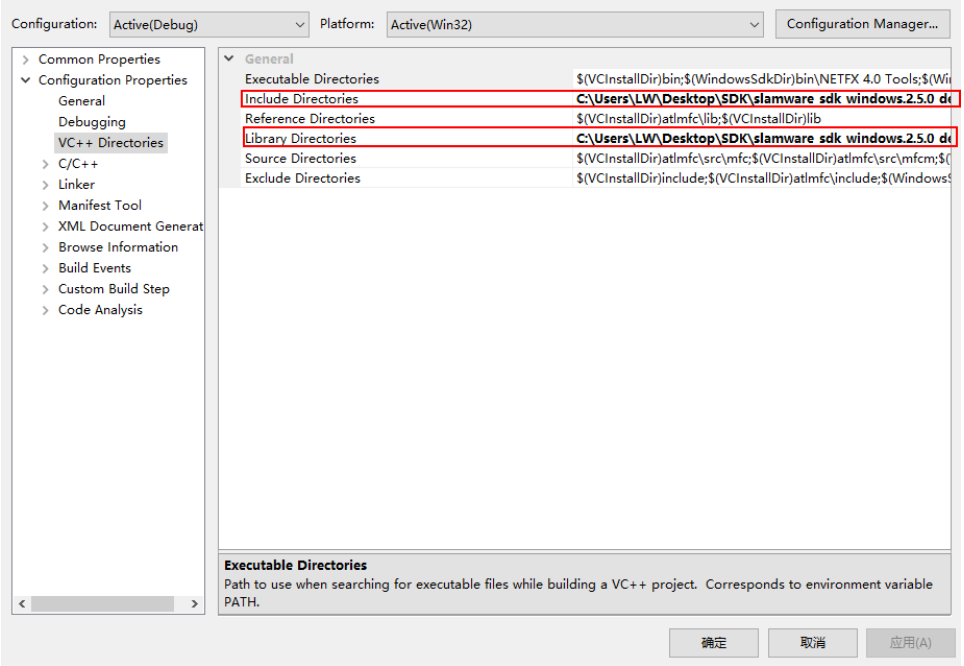
Compiling

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

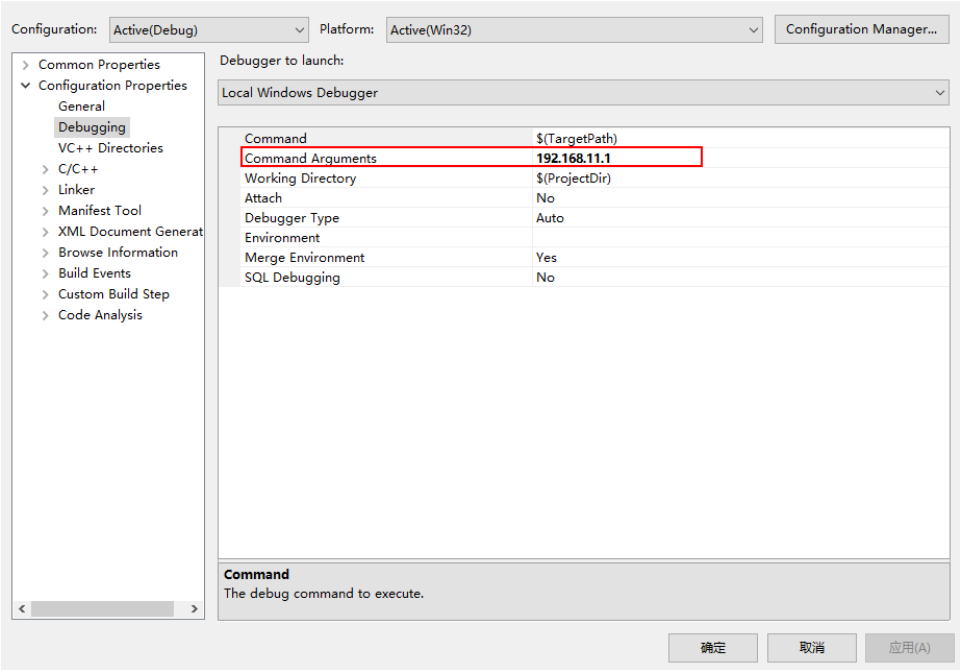


2. **Right click on "artifacts_demo", then " Properties"configure "include" and "lib" directories to the corresponding folder path of Slamware SDK.**

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

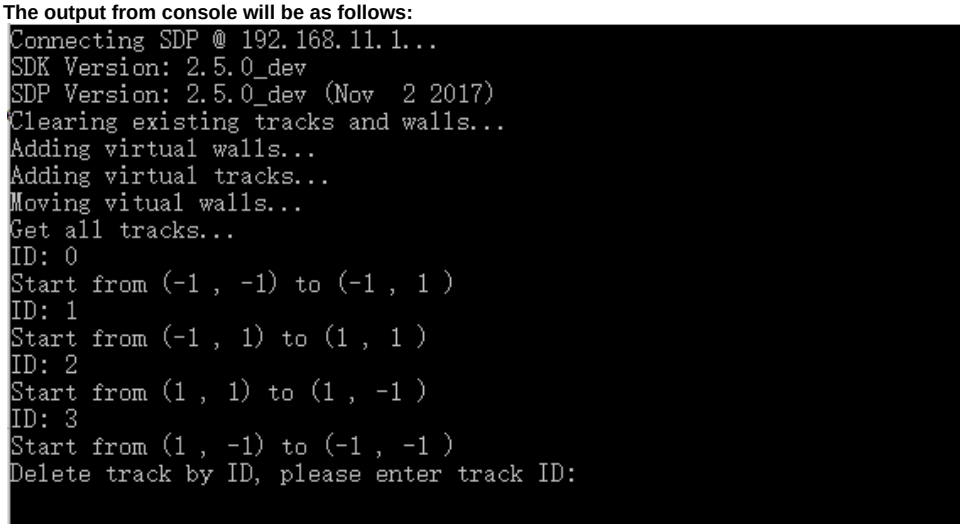


3. **Right click on "artifacts_demo", then "properties"set "Command Arguments" as follows:**
   Syntax artifacts_demo <IP address>



4. **Click " F5" to execute.**
5. **In robostudio, the output will be as follows:**
   Your browser does not support the HTML5 video element

6. **The output from console will be as follows:**



7. **Delete one of the virtual tracks by selecting one of the IDs. the ID selected in the video is 3.**

Code

- Delete all virtual tracks/walls.

/

```
            SlamwareCorePlatform sdp = SlamwareCorePlatform::connect(argv[1], 1445);
    std::cout << "Clearing existing tracks and walls..." << std::endl;
            sdp.clearLines(ArtifactUsageVirtualTrack);
            sdp.clearLines(ArtifactUsageVirtualWall);
```

- Add virtual walls

```
            std::cout << "Adding virtual walls..." << std::endl;
            std::vector<Line> walls;
            //add a 8 * 8 virtual wall square
            walls.push_back(Line(Point(-4, -4), Point(-4, 4)));
            walls.push_back(Line(Point(-4, 4), Point(4, 4)));
        walls.push_back(Line(Point(4, 4), Point(4, -4)));
            walls.push_back(Line(Point(4, -4), Point(-4, -4)));
            sdp.addLines(ArtifactUsageVirtualWall, walls);
```

- Add virtual tracks

```
            std::cout << "Adding virtual tracks..." << std::endl;
            std::vector<Line> tracks;
            //add a 2 * 2 virtual track square
            tracks.push_back(Line(Point(-1, -1), Point(-1, 1)));
            tracks.push_back(Line(Point(-1, 1), Point(1, 1)));
            tracks.push_back(Line(Point(1, 1), Point(1, -1)));
            tracks.push_back(Line(Point(1, -1), Point(-1, -1)));
            sdp.addLines(ArtifactUsageVirtualTrack, tracks);
```

- Modify a virtual wall (similar with virtual wall modification)

```
                std::cout << "Moving vitual walls..." << std::endl;
                //sleep 5 seconds for displaying purpose only, not necessary
                boost::this_thread::sleep_for(boost::chrono::milliseconds(5000));
                std::vector<Line> get_walls = sdp.getLines(ArtifactUsageVirtualWall);
                //shrink virtual wall square from 8 * 8 to 6 *6
                for (std::vector<Line>::iterator it = get_walls.begin() ; it != get_walls.end(); ++it)
{
            it->startP().x() *= 0.75f;
            it->startP().y() *= 0.75f;
                }
                sdp.moveLines(ArtifactUsageVirtualWall, get_walls);
```

- Delete virtual tracks(similar with virtual wall deletion

```
                std::cout << "Get all tracks..." << std::endl;
                std::vector<Line> get_tracks = sdp.getLines(ArtifactUsageVirtualTrack);
                for (std::vector<Line>::iterator it = get_tracks.begin() ; it != get_tracks.end(); ++it) {
                        std::cout << "ID: " << it->id() << std::endl;
                        std::cout << "Start from (" << it->startP().x() << " , " << it->startP().y() << ") "
<< "to ("
                                        << it->endP().x() << " , " << it->endP().y() << " ) " << std::
endl;
                }

                std::cout << "Delete track by ID, please enter track ID:" << std::endl;
                int id;
            bool is_found = false;
                std::cin >> id ;
                for (std::vector<Line>::iterator it = get_tracks.begin() ; it != get_tracks.end(); ++it) {
                        if (id == it->id()) {
                                sdp.removeLineById(ArtifactUsageVirtualTrack, id);
                                is_found = true;
                                break;
                        }
                }
                if(!is_found)
                        std::cout << "Wrong ID" << std::endl;
```