

KBSW180127 Win32- /

artifacts_demo, /

- -
 -
 -
 -
 -
-

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













Visual Studio

Visual Studio 2010SP1.Net FrameworkSP1

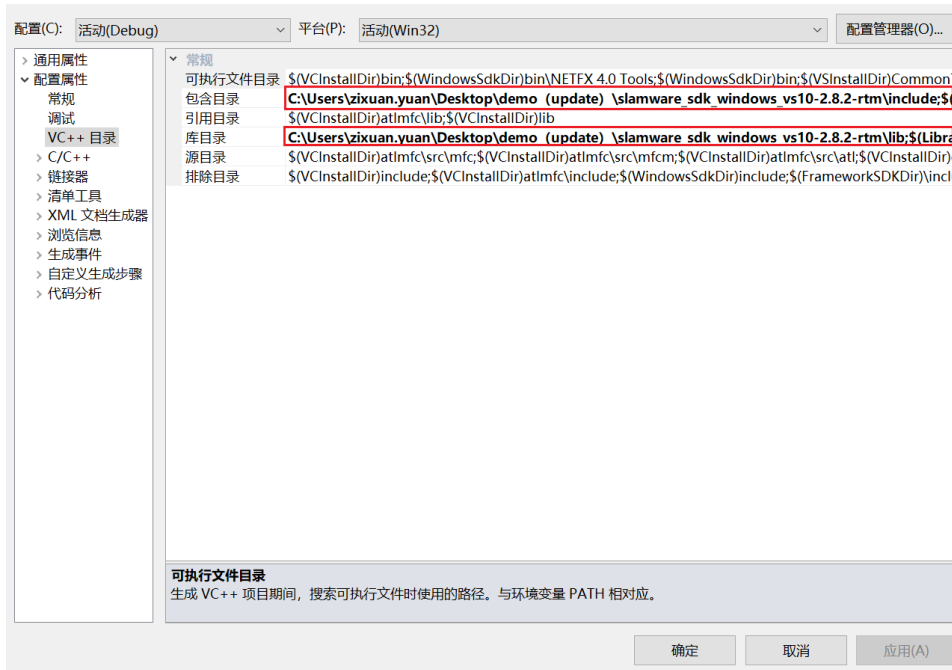
- - Slamware SDP mini
 - Slamware Slamware
 - Apollo/Ares/Athena
-

Win32-

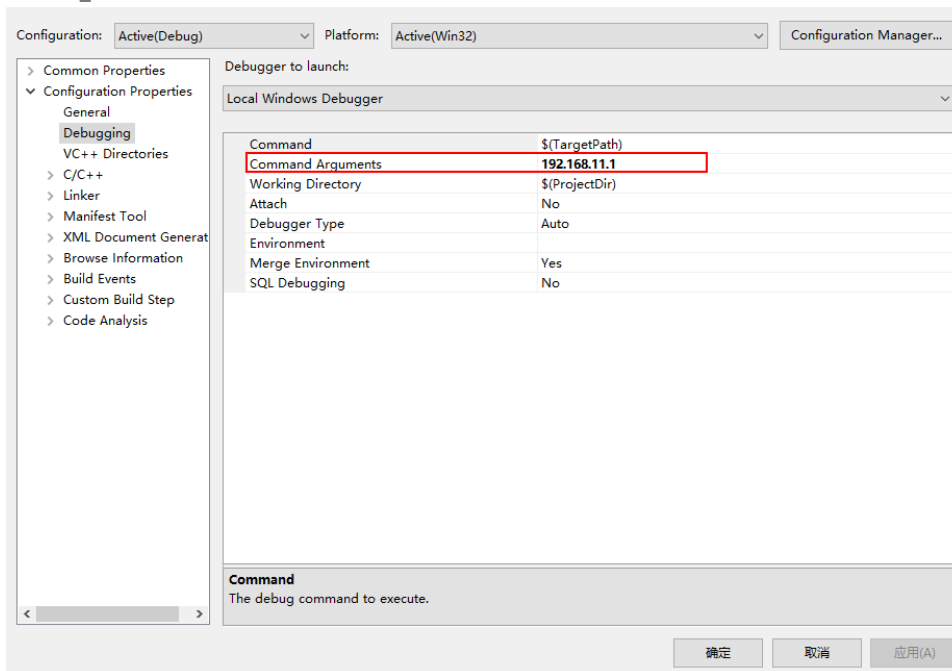
1. samplesartifacts_demo, StartUp project

-  Solution 'samples' (9 projects)
- >  **artifacts_demo**
- >  composite_map_demo
- >  configure_network_demo
- >  get_laser_scan
- >  get_power_status
- >  get_sensor_value
- >  go_home_to_charge
- >  move_to_spot
- >  rotation_action_demo

2. artifacts_demo, Slamware SDK includelib



3. artifacts_demo, Debuggingcommand Arguments 192.168.11.1 artifacts_demo <IP address>



4. F5

5. RoboStudio

Your browser does not support the HTML5 video element

6. Console

```
Connecting SDP @ 192.168.11.1...
SDK Version: 2.8.2_rtm
SDP Version: 2.8.2_rtm (Jun  3 2021)
Clearing existing tracks and walls...
Adding virtual walls...
Adding virtual tracks...
Moving virtual walls...
Get all tracks...
ID: 4
Start from (-1 , -1) to (-1 , 1 )
ID: 5
Start from (-1 , 1) to (1 , 1 )
ID: 6
Start from (1 , 1) to (1 , -1 )
ID: 7
Start from (1 , -1) to (-1 , -1 )
Delete track by ID, please enter track ID:
```

7. consoleIDID3

• /

/

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

•

```
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);
```

•

```
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);
```

•

```

std::cout << "Moving virtual 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);

```

•

```

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() << " ) "
    << 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;

```