KBSW191014 Win32-Robot's health

This document introduces the demo project of "robot_health", including how to get and clear the error status information of robot.

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

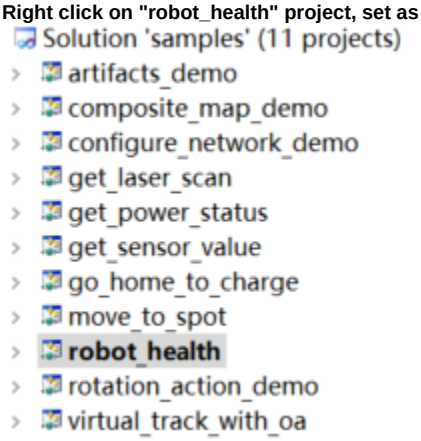
  - Slamware SDP mini
  - Slamware SDP
  - Slamware Kit
  - Zeus/Apollo robot base

Download

Win32-Demo

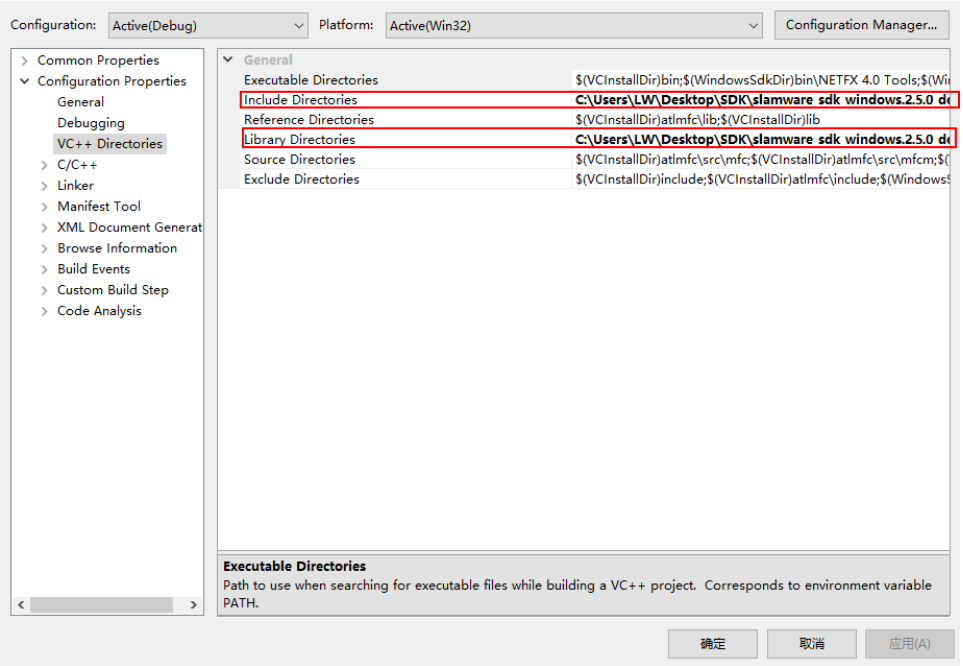
Compiling

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

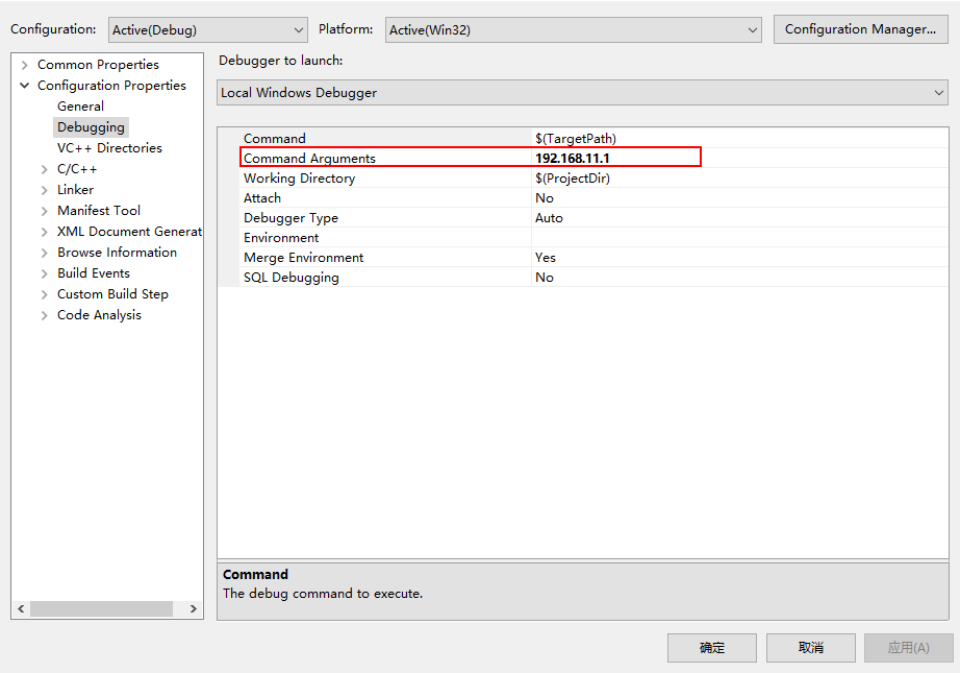


2. **Right click on "robot_health", then " Properties"configure "include" and "lib" directories to the corresponding folder path of Slamware SDK.**

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

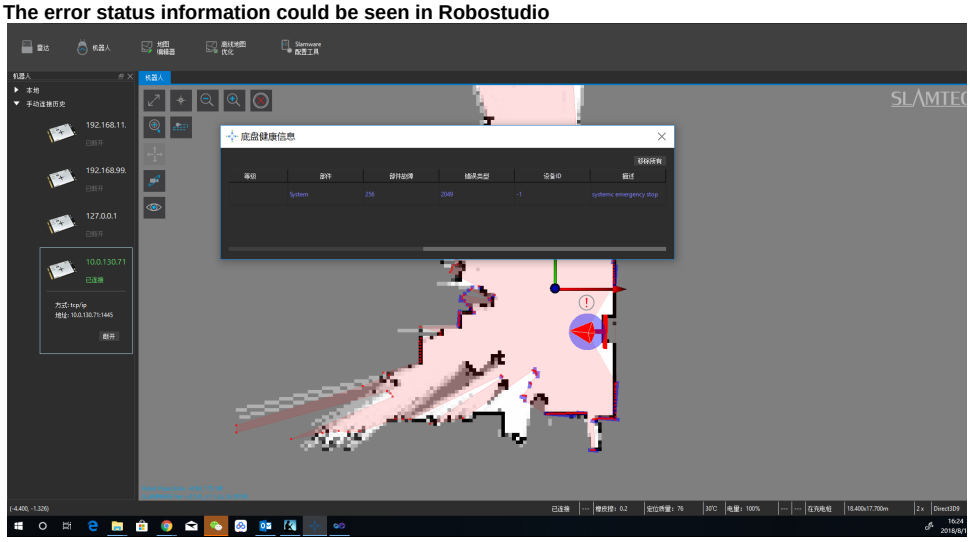


3. **Right click on "robot_status", then "properties"set "Command Arguments" as follows:**
   Syntax robot_health <IP address>

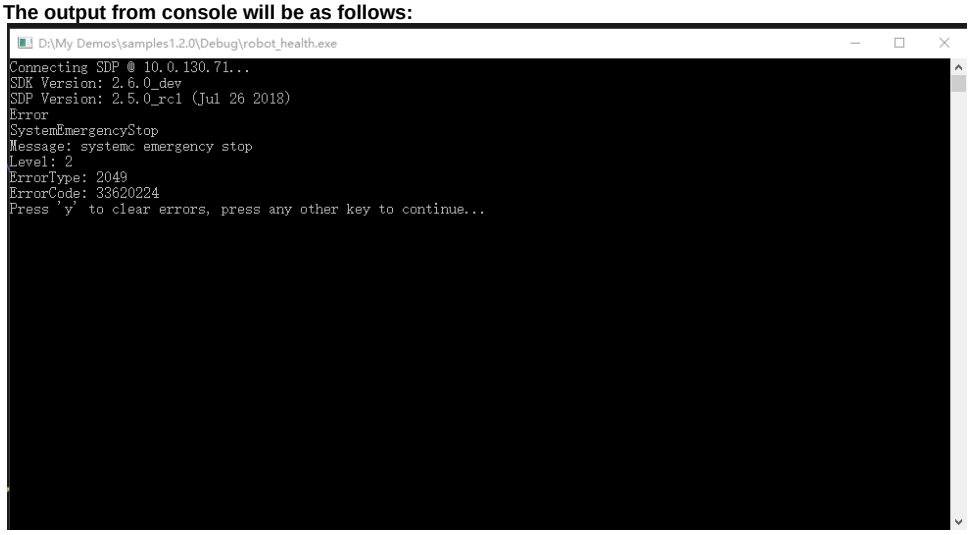


4. **Press the robot emergency stop button (take emergency stop as an example)Click " F5" to execute.**

5. **The error status information could be seen in Robostudio**



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



Code

- Print the error status information and clear it

Get robot health

```
SlamwareCorePlatform sdp = SlamwareCorePlatform::connect(ip_address, 1445);
        std::cout <<"SDK Version: " << sdp.getSDKVersion() << std::endl;
        std::cout <<"SDP Version: " << sdp.getSDPVersion() << std::endl;

        while(true){
                BaseHealthInfo robot_health = sdp.getRobotHealth();
                if(robot_health.hasError)
                        std::cout << "Error" << std::endl;
                if(robot_health.hasFatal)
                        std::cout << "Fatal" << std::endl;
                if(robot_health.hasWarning)
                        std::cout << "Warning" << std::endl;
                if(*robot_health.hasLidarDisconnected)
                        std::cout << "LidarDisconnected" << std::endl;
                if(*robot_health.hasSdpDisconnected)
                        std::cout << "SdpDisconnected" << std::endl;
                if(*robot_health.hasSystemEmergencyStop)
                        std::cout << "SystemEmergencyStop" << std::endl;
                for (auto it = robot_health.errors.begin();it != robot_health.errors.end(); ++
```

```
it) {
                                std::cout << "Message: " << it->message << std::endl;
                                std::cout << "Level: " << it->level << std::endl;
                                std::cout << "ErrorType: "<< it->componentErrorType <<std::endl;
                                std::cout << "ErrorCode: " << it->errorCode << std::endl;
                        }

                        int errors_size = robot_health.errors.size();
                        if(errors_size > 0){
                                std::cout << "Press 'y' to clear errors, press any other key to
continue..." << std::endl;
                                char is_error_clear;
                                std::cin >> is_error_clear;
                                if(is_error_clear == 'y' || is_error_clear == 'Y') {
                                        for (auto it = robot_health.errors.begin();it != robot_health.
errors.end(); ++ it) {
                                                sdp.clearRobotHealth(it->errorCode);
                                                std::cout << "Error: " << it->message << " cleared!" <<
std::endl;
                                        }
                                }
                        }
                }
```