

# KBSW180128 Win32-

Speed regulation, setssystemparameter()

- - 
  -
- 
- 
- 

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



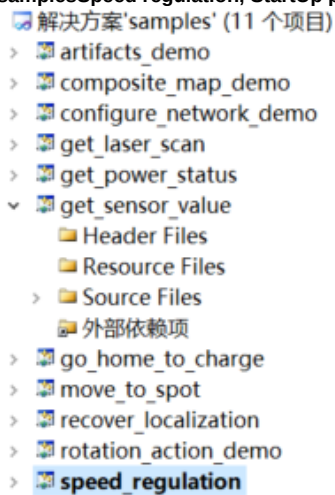
Visual Studio

Visual Studio 2010SP1.Net FrameworkSP1

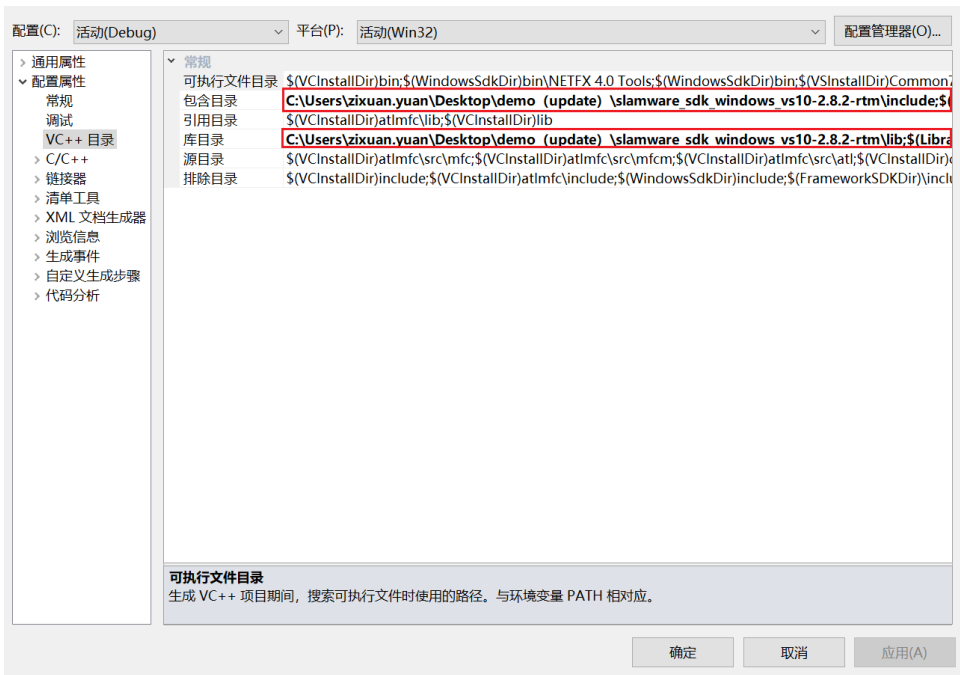
- - Slamware SDP mini
  - Slamware Slamware
  - Apollo/Ares/Athena

Win32-

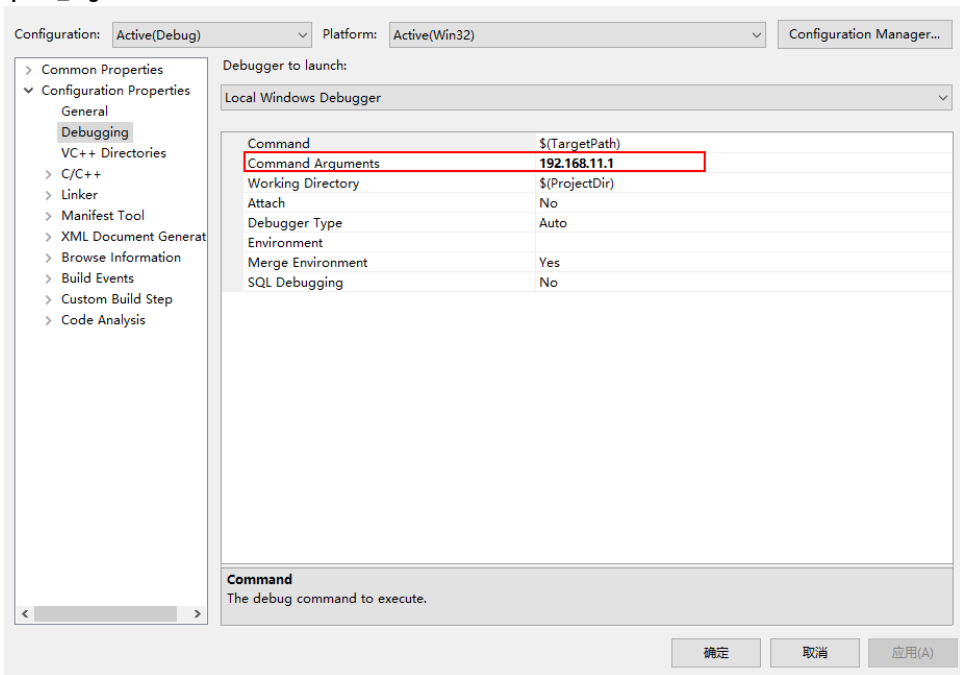
## 1. samplesSpeed regulation, StartUp project



## 2. speed\_regulation, Slamware SDK includelib



3. **speed\_regulation, Debuggingcommand Arguments 192.168.11.1**  
**speed\_regulation <IP address>**



4. **F5**
5. **Robostudio**

```

        SlamwareCorePlatform sdp = SlamwareCorePlatform::connect(ip_address, 1445);
        std::cout <<"SDK Version: " << sdp.getSDKVersion() << std::endl;
        std::cout <<"SDP Version: " << sdp.getSDPVersion() << std::endl;
        rpos::actions::MoveAction action = sdp.getCurrentAction();

        rpos::core::Location location1(1,0);
        rpos::core::Location location2(-1,0);
        rpos::core::Location location3(-1,2);
        rpos::core::Location location4(1,2);
        while (true)
        {
            if (action)
                action.cancel();

            action = sdp.moveTo(location1, false,true);
            if (action.getStatus() == rpos::core::ActionStatusError)
                std::cout << "Action Failed: " << action.getReason() << std::endl;
            bool bRet2 =sdp.setSystemParameter(SYSPARAM_ROBOT_SPEED,
SYSVAL_ROBOT_SPEED_HIGH);
            std::cout <<"Robot is moving to: (" << location1.x() <<" , "<<location1.y()
<<" ) on speed"<< " HIGH" << std::endl;
            action.waitForDone();

            action = sdp.moveTo(location2, false,true);
            if (action.getStatus() == rpos::core::ActionStatusError)
                std::cout << "Action Failed: " << action.getReason() << std::endl;
            bool bRet3 =sdp.setSystemParameter(SYSPARAM_ROBOT_SPEED,
SYSVAL_ROBOT_SPEED_LOW);
            std::cout <<"Robot is moving to: (" << location2.x() <<" , "<<location2.y()
<<" ) on speed"<< " LOW" << std::endl;
            action.waitForDone();

            action = sdp.moveTo(location3, false,true);
            if (action.getStatus() == rpos::core::ActionStatusError)
                std::cout << "Action Failed: " << action.getReason() << std::endl;
            bool bRet1 =sdp.setSystemParameter(SYSPARAM_ROBOT_SPEED,
SYSVAL_ROBOT_SPEED_MEDIUM);
            std::cout <<"Robot is moving to: (" << location3.x() <<" , "<<location3.y()
<<" ) on speed"<< " MEDIUM" << std::endl;
            action.waitForDone();

            action = sdp.moveTo(location4, false,true);
            if (action.getStatus() == rpos::core::ActionStatusError)
                std::cout << "Action Failed: " << action.getReason() << std::endl;
            bool bRet1 =sdp.setSystemParameter(SYSPARAM_ROBOT_SPEED, "0.3m/s");
            std::cout <<"Robot is moving to: (" << location4.x() <<" , "<<location4.y()
<<" ) on speed"<< " 0.3m/s" << std::endl; action.waitForDone();
        }
    }

```