The easiest way to join RoboCup Virtual Robot League for beginners

By Team Hinomiyagura

Last update : Thu Oct 23 17:08:04 JST 2014

[[Overview of this documentation]]

An aim of this page is spreading a virtual robot environment to develop and to evaluate response robots with RoboCup virtual robot league(VRL)'s simulation environment. By 2015, the VRL use a simulator "USARSim". After 2016 the VRL maybe use a simulator "Gazebo". For controlling your robot, you can use every thing which you can use. Our recommending one is ROS, then we have been writing following documentations for USARSim and Gazebo which are using with ROS. You can read both of them from following links.

This page's aim is spreading a virtual robot environment to develop and to evaluate response robots with the USARSim(Unified System for Automation and Robot Simulation) and the ROS(Robot Operating System).
It's the same way of preparing the environment to participate RoboCup Rescue Virtual Robot League.

Best regards,
Team Hinomiyagura.

[[Gazebo and ROS]]

This page's goal is making you to be a user of the Gazebo and the ROS.
Then, this page contains four big topics.

In near future, from 1 to 2 will be used for the participator in RoboCup virtual robot league.

We think the easiest way to be a user of something is playing with it.
Then at first, this page presents a simple installation steps of the Gazebo and the ROS.
And next, this page shows how to use the Gazebo and the ROS with simple tutorials.

[Hardware Requirements]

At first, we have to say you should prepare a powerful PC.
Followings are our hardware environments.

- Core i7 (Quad Cores) 4GHz, 8GB memory, SSD, nVidia's GPU(GTX 680).

[Installation of Gazebo and ROS]

Installation of Gazebo and ROS is consist of 2 steps : installing OS the Ubuntu and installing Gazebo with ROS.

1. Install Ubuntu
   You can select the version of Ubuntu : 12.04 or 14.04.
   But you have to select 64Bit version. Make it sure!
   Upgrade each software packages after installation of Ubuntu by following commands.

   ```
   $ sudo apt-get update
   $ sudo apt-get upgrade
   ```

   One more check by using "software update manager" in "System Tools".
   Often you can see some update informations.

   Maybe, you should download and install the nVidia video driver by yourself. Next software packages are useful : cachefilesd , sysinfo, aptitude

2. Install Gazebo and ROS
   Following commands lead you to finish.

   [Ubuntu 12.04]
   ```
   $ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu precise main" > /etc/apt/sources.list.d/ros-latest.list'
   $ sudo sh -c 'echo "deb http://packages.osrfoundation.org/drc/ubuntu precise main" > /etc/apt/sources.list.d/drc-latest.list'
   $ wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
   $ wget http://packages.osrfoundation.org/drc.key -O - | sudo apt-key add -
   $ sudo apt-get update
   $ sudo apt-get install drcsim-hydro
   ```

   [Ubuntu 14.04]
   ```
   $ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main" > /etc/apt/sources.list.d/ros-latest.list'
   ```

```
$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/drc/ubuntu trusty main" > /etc/apt/sources.list.d/drc-latest.list'
$ wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
$ wget http://packages.osrfoundation.org/drc.key -O - | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install drcsim-hydro
```

It's done. :-)

[Documents]

- RoboCup Rescue Virtual Robot League:http://www.robocuprescue.org/wiki/index.php?title=VRCompetitions
- Gazebo DRC simulator installation:http://gazebosim.org/tutorials?tut=drcsim_install

[How to use the Gazebo and the ROS(Check for the installation)]

Following sample commands shows you some great virtual world. You must execute a command:"source /usr/share/drcsim/setup.sh" for preparing some environment variables of ROS at once after you create a new terminal which you want to use it with ROS commands.

1. 8 tasks of Darpa Robotics Challenge.
   Now you can see 8 tasks by typing following commands.
   The important thing in running Gazebo is retrying.
   If you can't see any thing on the gazebo screen, kill the gazebo by pressing control key and c key simultaneously(this is used to be presented as "ctrl-c"). And re-execute the last command.

   ```
   [Darpa Robotics Challenge Task1-8]
   $ source /usr/share/drcsim/setup.sh
   $ roslaunch drcsim_gazebo drc_practice_task_1.launch
   $ roslaunch drcsim_gazebo drc_practice_task_2.launch
   $ roslaunch drcsim_gazebo drc_practice_task_3.launch
   $ roslaunch drcsim_gazebo drc_practice_task_4.launch
   $ roslaunch drcsim_gazebo drc_practice_task_5.launch
   $ roslaunch drcsim_gazebo drc_practice_task_6.launch
   $ roslaunch drcsim_gazebo drc_practice_task_7.launch
   $ roslaunch drcsim_gazebo drc_practice_task_8.launch
   ```

2. Walking Robot Atlas.
   This needs two terminals.

   ```
   [Terminal 1] $ source /usr/share/drcsim/setup.sh
   $ VRC_CHEATS_ENABLED=1 roslaunch drcsim_gazebo atlas_sandia_hands.launch

   [Terminal 2] $ source /usr/share/drcsim/setup.sh
   $ roslaunch drcsim_gazebo keyboard_teleop.launch
   ```

[Documents]

- http://gazebosim.org/tutorials?tut=drcsim_keyboard_teleop&cat=drcsim

[Files and Folders which you should know]

Followings are important directories and files.

1. World file

   [World fiel's directory]
   - /opt/ros/hydro/share/drcsim_model_resources/worlds/

   [World fiels]
   - drc_practice_task_1.world
   - drc_practice_task_2.world
   - drc_practice_task_3.world
   - drc_practice_task_4.world
   - drc_practice_task_5.world
   - drc_practice_task_6.world
   - drc_practice_task_7.world
   - drc_practice_task_8.world

2. Model file

   [Model file's directory]
   - ~/.gazebo/models/
   - /opt/ros/hydro/share/drcsim_model_resouces/gazebo_models/environments/

   [Model files]
   - drc_vehicle_xp900/model.config , drc_vehicle_xp900/model.sdf
   - sun/model.config , sun/model.sdf

3. Mesh file

   [Mesh file(.dae file)'s directory]
   - ~/.gazebo/models/MODELNAME/meshes/

- /opt/ros/hydro/share/drcsim_model_sources/gazebo_models/environments/MODELNAME/meshes/

4. launch file

   [Mesh file(.dae file)'s directory]
   - /opt/ros/hydro/share/drcsim_gazebo/launch/

[Documents]

- http://gazebosim.org/tutorials/?tut=attach_meshes

[[ROS tips, samples of SLAM with ROS]]

This chapter shows some samples and some tips of ROS.
They are used for both of real robots and virtual robots.

1. Using SLAM on the ROS with a URG for the real robot
2. Using SLAM on the ROS with a kinect for the real robot
3. ROS Tips 1: ROS basic tips. They must help you in your ROS work.
4. ROS Tips 2: Sample files for using ROS with USARSim which made by us.

[Using SLAM on ROS with an URG for the real robot]

This section shows how to use SLAM on the ROS with an URG(HOKUYO) for the real robot.
You need a real URG.
This operation is consist of 2 steps :

1. Checking the connection between the URG and the PC
2. Create SLAM map with URG

[Checking the connection between the URG and the PC]

1. Install the driver for the URG.

   ```
   $ sudo apt-get install ros-fuerte-laser-drivers
   ```

2. Prepare the URG. The URG needs DC +5V power from it's 8Pin connector, but not from USB.

3. Connect the URG with the PC by USB.

4. Check the URG's logical connection. Is there a file "/dev/ttyACM0"?. You can use following command on a terminal to see the file.
   If you could not find the file, please check the connection of the URG and it's power supply.

   ```
   $ ls -l /dev/ttyACM0
   ```

5. Change the parmission of the /dev/ttyACM0 to access the file from ROS program. Every time when you connect the URG with the PC, you have to do this again.

   ```
   $ sudo chmod a+rw /dev/ttyACM0
   ```

   You are looking forward to make a following file which having following 2 lines with a editor to chmod them automatically. You need use sudo.

   ```
   - /etc/udev/rules.d/50-usb-serial.rules
       KERNEL=="ttyUSB*",MODE="0666"
       KERNEL=="ttyACM*",MODE="0666"
   ```

6. Let's see the image from the URG. Please type following commands on the individual terminals.

   - TERMINAL 1 : $ roscore
   - TERMINAL 2 : $ rosrun hokuyo_node hokuyo_node
     (Executing an interface program for the URG, if you got some error messages, see the following trouble shooting)
   - TERMINAL 3 : $ rosrun rviz rviz -d `rospack find hokuyo_node`/hokuyo_test.vcg
     (Executing a visualizer program, you should adjust the location and size of the image)

[Create SLAM map with URG]

1. Create a helpful launch file for using the URG.
   Prepare a new package for a new launch file (please read appendix at the last of this page).
   The new launch file is for Hector mapping package to add the necessary information that the URG does not have.

   1. Read the appendix at the last of this page, create a new package and launch folder.
      For the following description, please name the new package as "using_urg".

   2. Create a new launch file which includes following 10 XML lines.
      For the following description, please name the new launch file as "4urg.launch".

   ```
   <?xml version="1.0"?>
   <launch>
   ```

```
<node pkg="hector_mapping" type="hector_mapping" name="hector_mapping" output="screen">
<param name="pub_map_odom_transform" value="true"/>
<param name="map_frame" value="map" />
<param name="base_frame" value="base_frame" />
<param name="odom_frame" value="base_frame" />
<param name="/use_sim_time" value="true" />
</node>
<node pkg="tf" type="static_transform_publisher" name="map_nav_broadcaster" args="0 0 0 0 0 0 base_frame laser
100"/>
<node pkg="rviz" type="rviz" name="rviz" args="-d $(find hector_slam_launch)/rviz_cfg/mapping_demo.vcg"/>
</launch>
```

2. Type following commands. Each command needs the individual terminal.
    - TERMINAL 1 : $ roscore
    - TERMINAL 2 : $ rosrun hokuyo_node hokuyo_node
    - TERMINAL 3 : $ roslaunch using_urg with_urg.launch

[Trouble Shooting for this section (about using URG)]

At first, please read here.
But if you got an error message about "Timeout" of the URG response, please read the following trouble shoot too.
If you saw the error message about "Timeout" from the hokuyo_node of ROS, you should check the spin of the optical part inside of the URG. And if you need, you should restart the URG. After disconnected USB, we often saw stopping the spinning optical part inside of the URG.
And we sow the hokuyo_node's "Timeout" error message. We unplugged and replugged the the power plug of the power code for the URG to restart the URG. And then the error message was gone.

[Documents]

- http://answers.ros.org/question/64644/how-mapping-using-hokuyo-lidar-urg-04lx-and-hector_slam/
- google about "hector_geotiff"

[Using SLAM on ROS with a Kinect for the real robot]

This section shows how to use SLAM on the ROS with a Kinect for the real robot.
You do not need PC1, but you need a Kinect.

[Install Kinect Driver]

Please type the following command on the terminal of the PC to install the kinect driver.

```
$ sudo apt-get install ros-fuerte-openni-kinect ros-fuerte-freenect-stack
```

[Try SLAM with Kinect]

1. Please connect the Kinect with the PC, and type following commands on the individual terminal.
    - TERMINAL 1 : $ roslaunch freenect_launch freenect.launch
    - TERMINAL 2 : $ rosrun rviz rviz

   Follows are operations on the RViz window.

2. Set the "Fixed Frame" under the "Global Options" at the left of the RViz window into "/camera_link".

3. Push "Add" button at the left bottom on the RViz window, and select "Image" from appeared "Display Type List".

4. Select an information type for "Image Topic" in "Displays" pain at the left of the RViz window.
   Click the right side of "Image Topic", and select an information type from appeared list.
   On appeared window, at first, click the "camera" to open the tree list, and second, click the "rgb" and "depth" to open sub tree list.
   And third, you can select "Image_color", "Image_mono", "Image_rect_raw", and so on for "Image Topic".

5. Push "Add" button at the left bottom on the RViz window, and select "Pointcloud2" from appeared "Display Type List".

6. Select an information type for "PointCloud2" in "Displays" pain at the left of the RViz window.
   Click the right side of "Topic", and select an information type from appeared list.
   On appeared window, at first, click the "camera" to open the tree list, and second, click the "depth_registered" and "depth" to open sub tree list.
   And third, you can select "Points" for "PointCloud2".

7. Even if you can or can not see any thing on the RViz window, at once shutdown RViz with saving it's config.

8. Shutdown "roslaunch freenect_launch freenect.launch" with ctrl+c while focus on the terminal window which it's running.

9. Please return to the operation number 2.

10. Please keep doing operation number 2 ~ 8 at several times, with often disconnection and reconnection the Kinect.

[Trouble Shooting for this section (about using kinect)]

May be, the 1st time, you won't see any thing on the RViz window. But please keep trying cyclically operation number 1 ~ 7 by when you can see some images from the Kinect.

[Documents]

- http://robotics.stackexchange.com/questions/1025/cant-see-kinect-data-in-ros

[ROS Tips 1 (For comfortable developing on ROS)]

Following tips are very short text to avoid being into sleep.

[1. What is the difference between rosrun and roslaunch?]

"rosrun" and "roslaunch" have same purpose that is execution of ros commands.
The difference between rosrun and roslaunch is the method to write commands and options.

1. rosrun

   ```
   $ rosrun "package-name" "program-name" [option1] [option2] ...
   ```

2. roslaunch

   ```
   $ roslaunch "package-name" "launch-file-name"
   ```

A launch file includes the program name and some option strings in XML.
It can help you to save typing same command and options. (^_^)

[2. Launch file]

Launch files will help you to save typing options at every time in executing ros commands.
A launch file like a batch file of DOS, but it is wrote in XML.
Do not worry, XML in launch file is very easy.

1. Creating a launch file.

   1. Create an empty package for folder of your new launch file.
      Please replace "new-package-name".

      ```
      $ roscd
      $ cd sandbag
      $ roscreate-pkg "new-package-name"
      $ cd "new-package-name"
      $ rosmake "new-package-name"
      $ mkdir launch
      $ cd launch
      ```

   2. Create a new launch file.
      Do not forget that you have to be at the above launch folder.

      ```
      $ gedit "new-launch-file-name".launch
      ```

      You can see a sample of inside of a launch file which execute some ROS nodes in the past section "Create SLAM map with URG".

2. Using ROS command with launch file.
   After you made your launch file at once, following is just you have to do.

   ```
   $ roslaunch "new-package-name" "launch-file-name"
   ```

[ROS Tips 2 (Sample files for USARSim)]

You can find some samples made by us in here.

1. Map data : SampleMaps.zip
   It was used for some experiments in a J-SOFT article.
   After download and extract, you find some bat files and udk files. You should move bat files to under "C:/UDK/UDK-201X-XX/USARRunMaps". And You should move udk files to under "C:/UDK/UDK-201X-XX/USARGames/Content/Maps".

[[USARSim and ROS]]

This chapter's goal is making you to be a user of the USARSim and the ROS.
Then, this page contains four big topics.

1. How to install the USARSim
2. How to install the ROS
3. How to use the USARSim and the ROS(Check for the installation)

From 1 to 3 is for the participator in RoboCup virtual robot league.
And from 2 to 4 is for the participator in RoboCup real robot league.

We think the easiest way to be a user of something is just only using it.

Then at first, this page presents a simple installation steps of the USARSim and the ROS.
And next, this page shows how to use the USARSim and the ROS with simple tutorials.

At the last of this page, we put ROS tips sections.
ROS Tips1 is written for ROS basic tips. They must help you in your ROS work.
ROS Tips2 has sample files made by us.

[RoboCup Rescue Virtual Robot League Informations]

- RoboCup Rescue Virtual Robot League:http://www.robocuprescue.org/wiki/index.php?title=VRCompetitions

[Hardware Requirements]

At first, we have to say you should prepare two PCs.
We are using two note PCs for confortable computer environment to run both the USARSim and the ROS. Of course, we can run two PCs virtually on one real PC by using like "Virtual Box" software. Our experience shows that it places high load on the PC. Followings are our hardware environments.

- PC1 : This machine is for the USARSim. PC1 needs GPU power. We used to select nVidia's GPU for PC1. Core i3 and 4GB memory are enough for PC1.
- PC2 : This machine is for the ROS. PC2 needs CPU power and memory. Faster CPU is better for PC2. And 8GB memory is recommended for PC2.

We has been selecting nVidia's GPU for PC2, too. Because we bought a pair of PC always.(^_^)

[USARSim Installation on PC1]

Installation of USARSim is consist of 3 steps :

1. UDK installation
2. Copy USARSim package over UDK
3. Make WSS

[Required softwares]

- UDK version : 2013-07
- USARSim : current

[UDK Installation]

1. Download DirectX End-User Runtime Web Installer from here, and install it.

2. Download UDK(Version 2013-07) from following URL, and install it.

   - http://www.unrealengine.com/ja/udk/

3. Download Visual c++ 2008 Redistributable Package (**Please use 2008 version, not 2010**)

   - (64bit OS):http://www.microsoft.com/en-us/download/details.aspx?id=15336
   - (32bit OS):http://www.microsoft.com/ja-jp/download/details.aspx?id=5582

4. Install telnet client of Windows (For easy debugging of USARSim)

   - Control Panel -> Programs and Features -> Turn Windows features on or off(At the left side menu) -> check "Telnet Client" on -> OK

5. Install UDK

   - Run "UDKInstall-2013-07.exe"
     Click "Confirm" -> check "PSGameUT3Title", and click "IENext" -> click "Install" -> click "IENext" -> check "Return to the Desktop" and click "Done"

   - To use an old version : Run "UDKInstall-2012-02-BETA.exe"
     Click "Confirm" -> click "Install" -> check "Return to the Desktop" and click "Done"

6. Read the document

   - http://udn.epicgames.com/Three/TechnicalHomeJP.html

[Copy USARSim package over UDK]

1. Download the USARSim package from the SourceForge.

   $ git clone git://git.code.sf.net/p/usarsim/code usarsim-code
   (Under proxy, please try "$ git clone http://git.code.sf.net/p/usarsim/code usarsim-code")

   To get an old version (not recommended) for example version1.3 :
   - Open "http://sourceforge.net/projects/usarsim/" , follow the links "Files" -> "usarsim-UDK" , download "USARSimFull_UDKV1.3.zip"
   - Extract the downloaded zip file.

2. Copy all of folders and files in the extracted usarsim folder into C:/UDK/UDK-201x-xx/

3. Run "make_clean.bat" (Followings are results of combinations some versions of UDK with some versions of USARSim)

    - UDK-2012-02 and USARSim-v1.3 no error
    - UDK-2012-02 and USARSim-current warning
    - UDK-2013-07 and USARSim-v1.3 error in class 'Canvas' -> need patching canvas.uc
    - UDK-2013-07 and USARSim-current no error

4. Run xxx.bat in USARRunMaps folder
   At first running, some setting files are created.
   If you want to make some changes in usarsim configuration files, you need this first run.

[Make WSS]

If you could not find WSS.exe in C:/UDK/UDK-201x-xx/USARTools/WSS, you can make by following steps.

1. Download Microsoft Visual C++ Express 2010 Web Installer from here, and install it.

2. Download Qt (version 4.3.4 or later) for Microsoft Visual C++ Express 2010 from here, and install it.

3. Download MinGW (any version is ok) from here, and install it.
   MinGW is needed for getting libws2_32.obj.
   And copy libws2_32.a to the distation folder with changing it's filename.

    - `copy "C:/MinGW/lib/libws2_32.a" "C:/Program Files(x86)/Microsoft Visual Studio 10.0/VC/lib/libws2_32.obj"`

4. Next, compile programs in following three folders.
   Repeat step 5 and step 6 in following three folders.

    - Main Program Folder : C:/UDK/UDK-201x-xx/USARTools/WSS
    - Plugin1 Folder : C:/UDK/UDK-201x-xx/USARTools/WSS/plugins/DistanceOnlyPropagationModel
    - Plugin2 Folder : C:/UDK/UDK-201x-xx/USARTools/WSS/plugins/ObstaclePropagationModel

5. Execute "Qt Command Prompt" on the PC1, and type the following commands on the Dos terminal.

    - `cd C:/UDK/UDK-201x-xx/USARTools/WSS`

    - qmake (the WSS document says using the command "qmake -t vcapp", but "-t vcapp" is needless for making by Visual Studio Command Prompt)

6. Execute "Visual Studio Command Prompt(2010)" on the PC1, and type the following commands on the Visual Studio Command Prompt.

    - `cd C:/UDK/UDK-201x-xx/USARTools/WSS`

    - `nmake`

7. After compiling three programs, copy Qt Libraries into the same holder where WSS.exe is.

    - `copy C:/Qt/4.x.x/bin/QtCored4.dll C:/UDK/UDK-201x-xx/USARTools/WSS/debug`

    - `copy C:/Qt/4.x.x/bin/QtGuid4.dll C:/UDK/UDK-201x-xx/USARTools/WSS/debug`

8. Finally, to run the WSS.exe, copy following files into one folder which the WSS.exe exists in.

    - WSS.exe
    - QtCored4.dll
    - QtGuid4.dll
    - "plugins" folder
    - wss_distanceonlymodel.dll in plugins folder
    - wss_obstaclemodel.dll in plugins folder

[USARSim documents]

- USARSimhttp://sourceforge.net/apps/mediawiki/usarsim/index.php?title=Main_Page
- WSS:http://sourceforge.net/apps/mediawiki/usarsim/index.php?title=Wireless_Simulation_Server
- WSS manual:http://usarsim.cvs.sourceforge.net/viewvc/usarsim/usarsim/Tools/WSS/documentation/WSS.pdf
- MetricTool:http://sourceforge.net/apps/mediawiki/usarsim/index.php?title=MetricTool

[ROS Installation on PC2]

Installation of ROS is consist of 2 steps :

1. ROS installation
2. Setting

[Required softwares]

- Ubuntu version : 12.04
- ROS version : fuerte

[ROS Installation]

- Please type following commands on the terminal window of PC2.

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu precise main" > /etc/apt/sources.list.d/ros-latest.list'
$ wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install ros-fuerte-desktop-full
```

[ROS Settings]

- Please type following commands on the terminal window of PC2.

```
$ echo "source /opt/ros/fuerte/setup.bash" >> ~/.bashrc
$ . ~/.bashrc
$ sudo apt-get install python-rosinstall python-rosdep ros-fuerte-brown-remotelab ros-fuerte-control ros-fuerte-common-msgs
ros-fuerte-slam-gmapping ros-fuerte-hector-slam
$ rosws init ~/fuerte_workspace /opt/ros/fuerte
$ echo "source ~/fuerte_workspace/setup.bash" >> ~/.bashrc
$ mkdir ~/fuerte_workspace/sandbox
$ source ~/fuerte_workspace/setup.bash
$ rosws set ~/fuerte_workspace/sandbox
$ source ~/fuerte_workspace/setup.bash
$ roscd
$ cd sandbox
$ git clone git://git.assembla.com/usarsim.git
(Under the proxy, try "git clone http://git.assembla.com/usarsim.git")
$ rosmake usarsim_inf
```

[ROS documents]

- Open http://www.ros.org/wiki/ and follow the links "Installation" -> "Fuerte installation" -> "Ubuntu" (**See Fuerte version**)

[ROS Tutorial]

- If you want More information, see Tutorials.
- http://www.ros.org/wiki/roscpp
- http://www.ros.org/wiki/rospy

[How to use USARSim and ROS : Check for the installation]

For installation check, create an image of SLAM with gmapping by using USARSim(on the PC1) and ROS(on the PC2). This operation is consist of 2 steps :

1. On PC1(Windows) : Executing USARSim
2. On PC2(Ubuntu) : Executing ROS packages

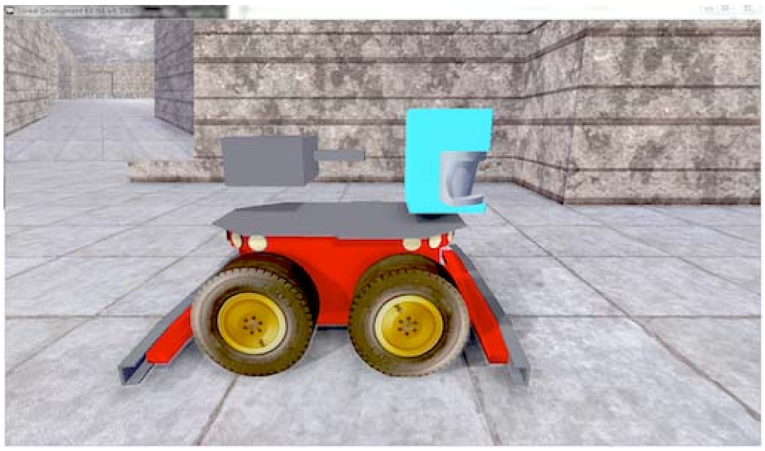
[PC1(Windows) SIDE]

```
1. Check the IP Adress of PC1
2. Execute the UDK with a map. I.E. "C:/UDK/UDK-201x-xx/USARRunMaps/VMAC2011.bat"
To quit the UDK, push TAB key, type "exit", and push enter key
```

[PC2(Ubuntu) SIDE]

```
1. Edit /usarsim/usarsim_inf/launch/usarsim.launch to set the PC1's IP Adress on a terminal window.

    $ roscd usarsim_inf/launch
    $ vi usarsim.launch
    $ replace "tweety" with the IP Adress of PC1 (Example : value="tweety" -> value="192.168.1.11")

2. And do following commands. Use different terminal window or different terminal tab for each command pair.

- Terminal 1 : $ roscore
        (Executing a hub program of ROS)

- Terminal 2 : $ roslaunch usarsim_inf usarsim.launch
        (Executing an interface program between ROS and USARSim)

- Terminal 3 : $ rosrun gmapping slam_gmapping scan:=lms200
        (Executing a gmapping SLAM program)

- Terminal 4 : $ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
        (Executing a robot UI program on ROS. Now you can adjust robot speed and input robot moving key (^_^)b )

- Terminal 5 : $ rosrun map_server map_saver
        (Save a snapshot map image from SLAM. You should run this command occasionally to renew the map image)

- Terminal 6 : $ eog ./map.pgm
        (Please run eog at same directory of Terminal 5. The program "eog" displays map.pgm with automatic renewing)
```

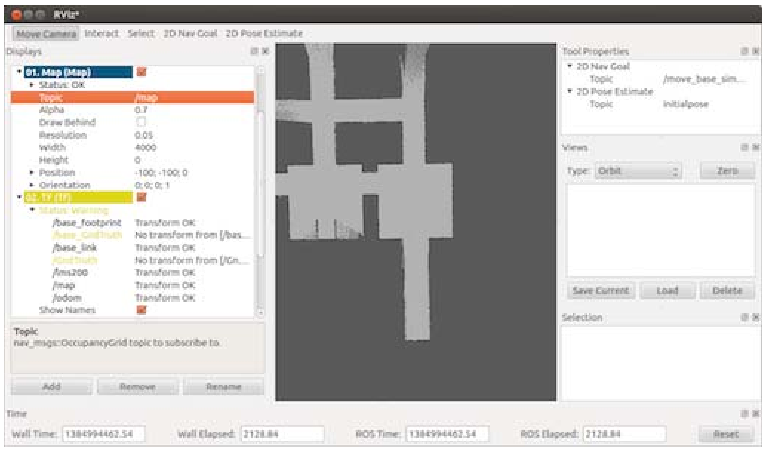
After Terminal 2 roslaunch, you can see a "P3AT" robot appearing on the USARSim screen like the following picture.



Instead of "rosrun map_server map_saver" and "eog ./map.pgm" , you can use RViz.

```
- Terminal 7 : $ rosrun rviz rviz
        * Add map in the Displays list(left side of the rviz window)
        * Select the topic of the added map as "nav_msgs::OcuupancyGrid topic to subscribe to"
        * Turn over the map image
```

It will give you an image like the following.



```
01. Map (Map)
  Status: OK
  Topic          /map
  Alpha          0.7
  Draw Behind
  Resolution     0.05
  Width          4000
  Height         0
  Position       -100; -100; 0
  Orientation    0; 0; 0; 1
02. TF (TF)
  Status: Warning
    /base_footprint   Transform OK
    /base_GndTruth    No transform from [/bas...
    /base_link        Transform OK
    /GndTruth         No transform from [/Gn...
    /ms200            Transform OK
    /map              Transform OK
    /odom             Transform OK
  Show Names

Topic
nav_msgs::OccupancyGrid topic to subscribe to.

Tool Properties
  2D Nav Goal
    Topic        /move_base_sim...
  2D Pose Estimate
    Topic        initialpose

Type: Orbit

Wall Time: 1384994462.54   Wall Elapsed: 2128.84   ROS Time: 1384994462.54   ROS Elapsed: 2128.84
```

[Trouble Shooting for whole of this page]

The biggest trouble will be caused by the firewall or the unti-virus software. They often cut the network connection between PC1 and PC2. Please stop or lower the security level. For example, on PC1(Windows), change the kind of the network from "Public" to "Private".