Tutorial to setup SITL Gazebo for Arducopter on Windows

Version 0.8
Aurélien ROY
01/09/2015

Content

I. Goal

The first part of this tutorial is intended to Windows users. It is a walktrough the setup of an Ubuntu virtual machine on VMWare, the setup of a shared folder that contains the Ardupilot project, so it is accessible from both Windows and Ubuntu, and the initial configuration of Ubuntu.
The second part covers the download and installation of Ardupilot's SITL environment, and of ROS (Robot Operating System) that contains the Gazebo simulator. And finally it will guide you through the launch of a Gazebo simulation.

II. Installation requirements

You will need:

- A good internet connection
- At least 40 Gb of free space on your hard drive
- A powerful computer (at least 6 Gb of RAM)
- 2 hours before you…

It is also possible to setup the Gazebo SITL on a less powerful computer if you install Linux Ubuntu directly (dual boot, no emulation).

III. Beforehand downloads

VMWare, version 12.0 (~60 Mo)
https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0

For a 32 bits system, use the previous version:
https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0

Ubuntu, version 14.04.3 (~ 1 Go)
(similar to the 14.04.2 advised by ArduCopter)

64 bits version:
http://releases.ubuntu.com/14.04/ubuntu-14.04.3-desktop-amd64.iso
32 bits version:
http://releases.ubuntu.com/14.04/ubuntu-14.04.3-desktop-i386.iso

IV. Acknowledgements

Instructions and scripts have been inspired, or come partially from, the following sources:

- http://dev.ardupilot.com/wiki/simulation-2/sitl-simulator-software-in-the-loop/setting-up-sitl-on-windows/
- https://pixhawk.org/dev/ros/sitl
- *SITL_Manual_Setup_v3.1.doc*, available on http://www.rcgroups.com/forums/showthread.php?t=2190616

And ROS commands come from:

- http://dev.ardupilot.com/wiki/using-rosgazebo-simulator-with-sitl/
- https://pixhawk.org/dev/ros/sitl

Many thanks to Alex Buyval for creating the initial Arducopter simulation on Gazebo, to the PX4 team for all their work on ROS/Gazebo, to Randy Mckay for his VMWare instructions, and finally to the whole Ardupilot community !

V. Limitations

A few directory paths had to be hard coded in the shell scripts that come along this tutorial (at the beginning of *step_3_install_arducopter.sh* and *step_5_create_ros_workspace.sh*).

The installation as described by the tutorial works for a single instance of the Ardupilot project, because it stores the path to Ardupilot project in the PATH environment variable.

VI. Setting up the Virtual Machine

These instructions are for Windows users.
It will setup a full Linux virtual machine, with graphical interface (contrarily to the basic Vagrant / VirtualBox emulation).

1. Configuration of VMWare

Configuration summary

Summary of the configuration of the virtual machine:

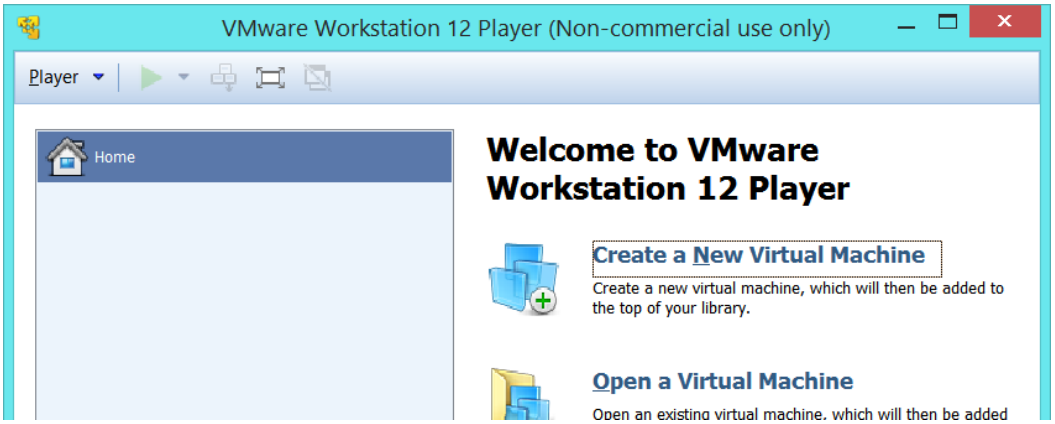
Page / Parameter		Value
Install from		
	Disc image	ubuntu-14.04.3-desktop-amd64.iso
Personalize Linux		
	Full name	apm
	User name	apm
	Password	apm
Name VM		
	Virtual machine name	SITL
	Location	c:/px4/Virtual Machines/SITL
Disk Capacity		
	Disk capacity	30.0 Gb
		Split virtual disk into multiple files
Customize hardware		
	Memory	3072 Mb
	Processors	4
	Network adapter	NAT

Note :
The user name and password have been chosen for an easy use of the SITL.

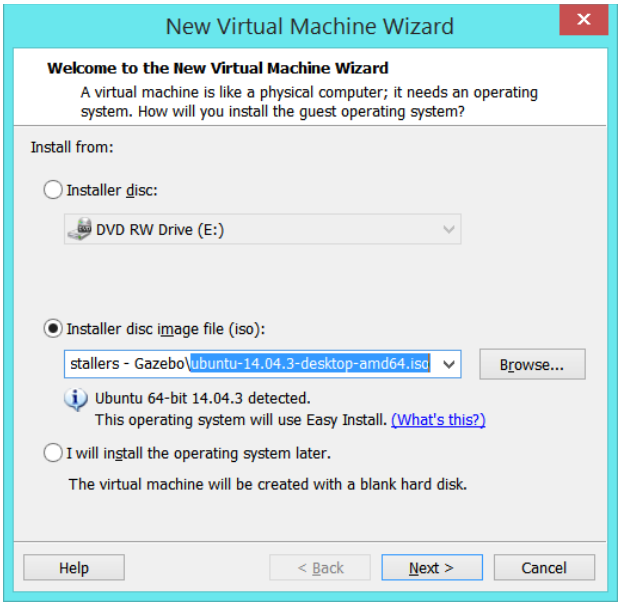
Note :
In my case I preferred to place all software / source codes relevant to Arducopter in the c:/px4 directory. But you can place it anywhere you wish, just change the paths in the rest of the document.

Steps

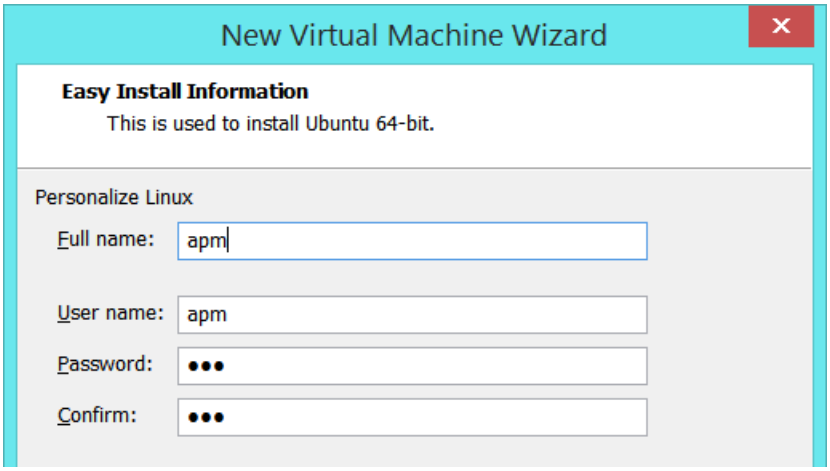
In VMWare, click on « Create a New Virtual Machine »:



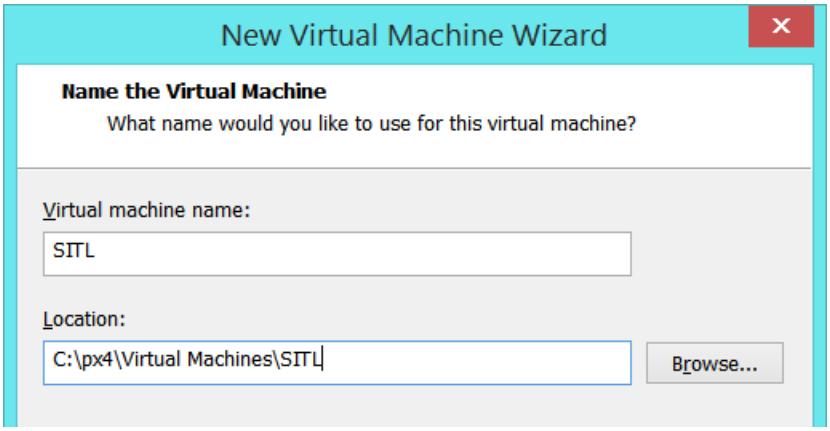
Select the Ubuntu 14.04.3 iso :



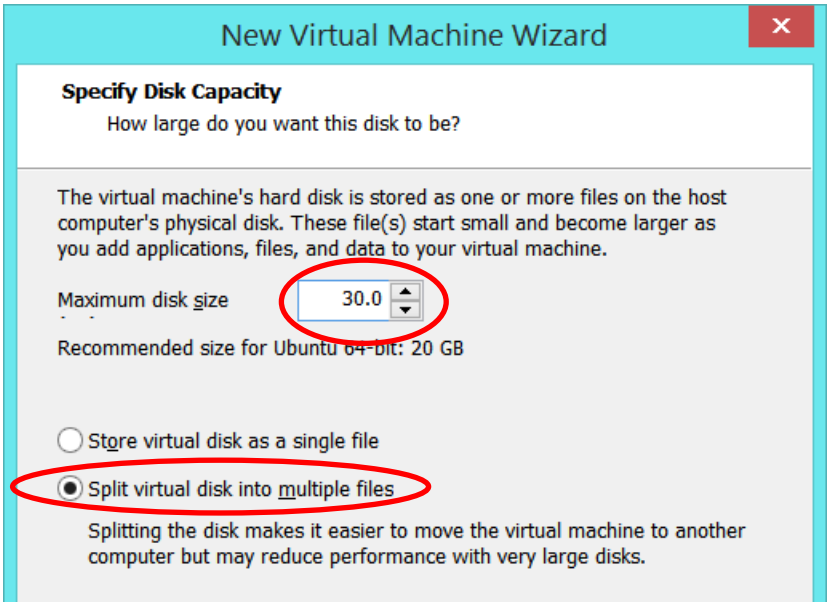
Enter the login « apm » and password « apm » :



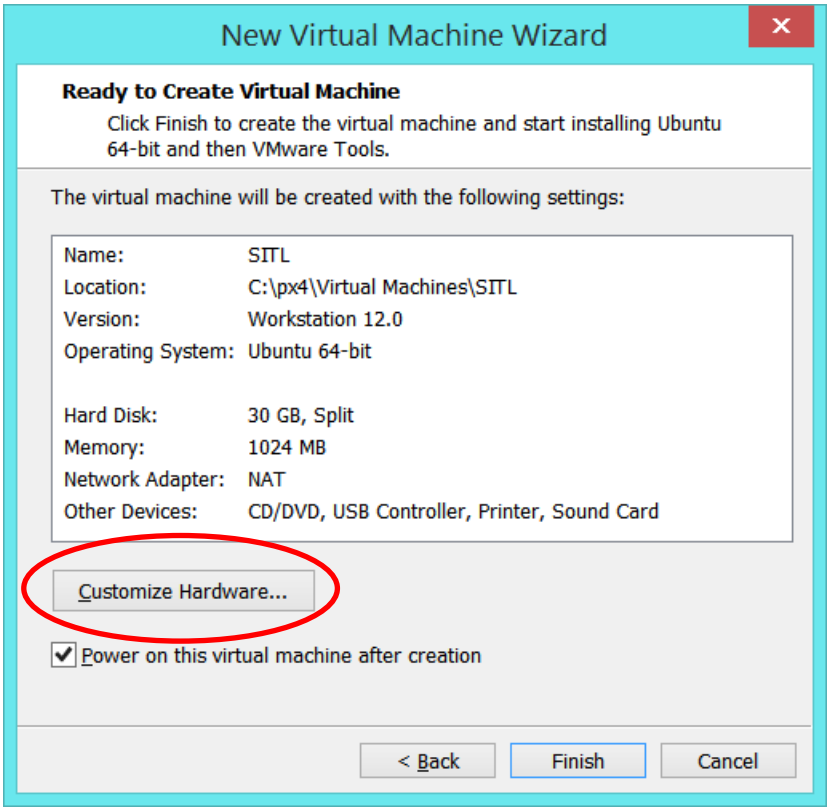
Name the machine « SITL » and place in « C:\px4\Virtual Machines\SITL »:



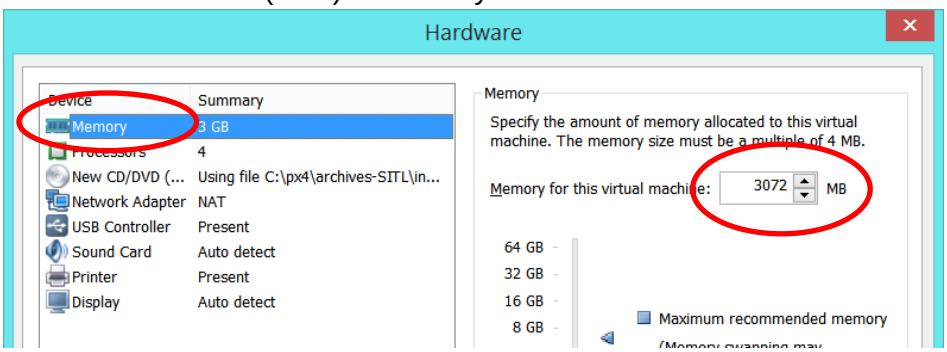
Give her 30.0 Gb and make sure to active « Split virtual disk into multiple files » :



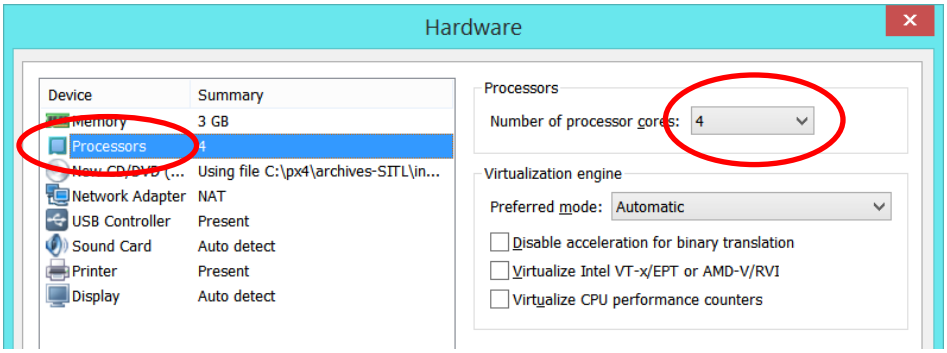
On the summary page, click on « Customize Hardware… » :



Allocate it 3072 MB (3GB) of memory:



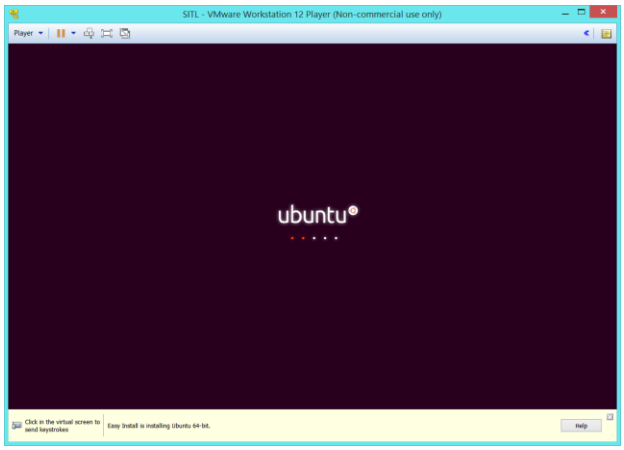
Define 4 processors:

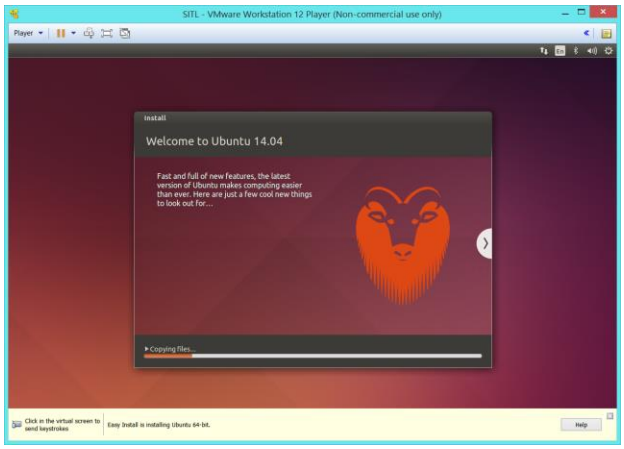


Leave the Network Adapter on NAT.
Accept the *Hardware* window, and click on *Finish* to complete the creation of the virtual machine.

2. 1st launch of Ubuntu

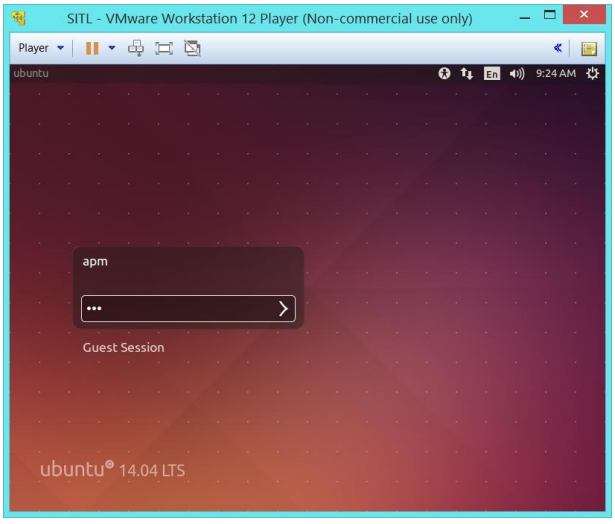
By default VMWare starts Ubuntu right after the creation of the VM.
Ubuntu starts to installs itself:

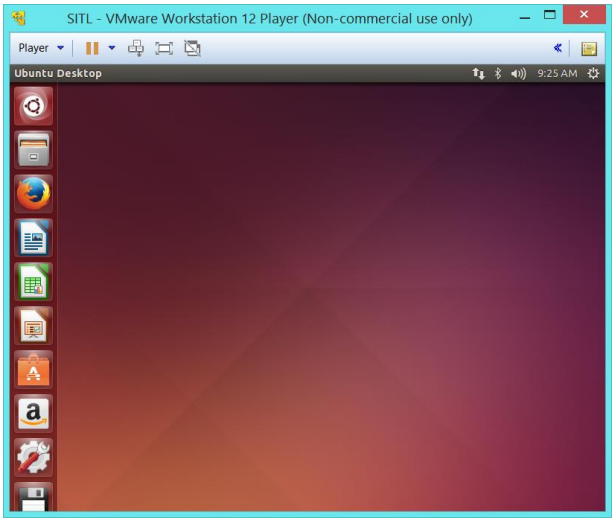




> *Note :*
> The hard drive load may jump to 100% during the installation, or on the login page after each startup of Ubuntu. Avoid launching cumbersome processes at the same time.

Log on Ubuntu (password « apm »):

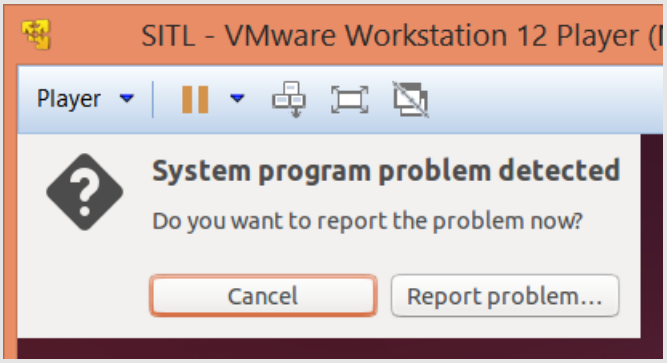




> *Note :*
> VMWare may propose to connect your camera or other devices to Ubuntu. Refuse (unless you need them, like later on for the joystick).

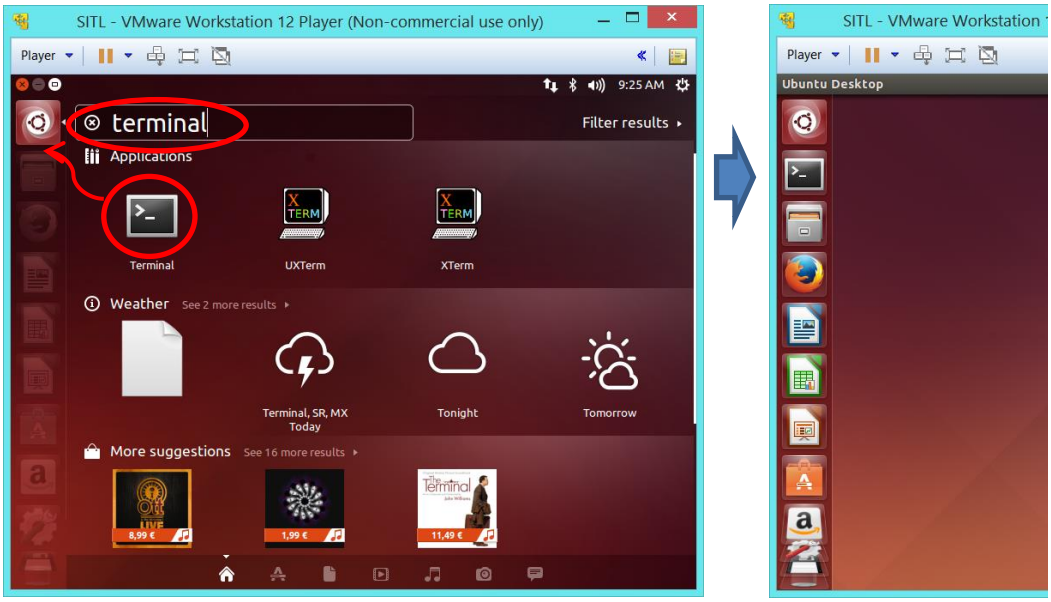
Note :

Some errors may occur on VMWare, e.g. “Mouse thread did not respond to grab request”, or “Could not connect to X”, etc., and even sometimes result in a fatal crash.



Do not despair, just click OK, and if required re-launch VMWare and the virtual machine.

Click on the Ubuntu logo on the top left corner, and type « terminal » in the type box that appeared to search for the software. Then drag and drop terminal icon in the taskbar to create a shortcut:



Update the time and time zone on the top right corner.

The following shell scripts are required to setup the environment:

- *step_1_configure_ubuntu.sh*
- *step_2_create_shared_folder.sh*
- *step_3_install_arducopter.sh*
- *step_4_install_ros.sh*
- *step_5_create_ros_workspace.sh*

To retrieve them, type in a terminal window the commands (from any directory, e.g. ~/Documents):

```
git clone https://github.com/AurelienRoy/ardupilot_sitl_ros_tutorial.git
chmod +x ./*.sh
```

It will authorize the execution of shell scripts.

> *For the users who see Linux for the first time:*
> The user directory is « /home/<user name>", and in our case "/home/apm".
> It can be shortened with the "~" character, e.g. "~/Documents" is identical to "/home/apm/Documents".

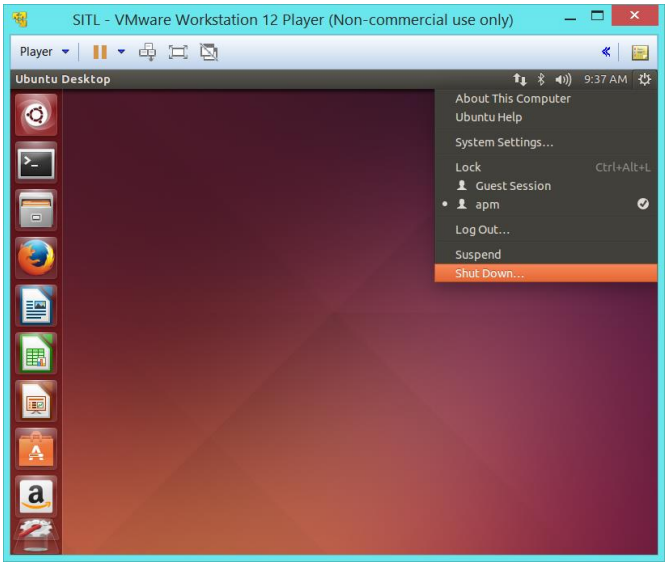
Then in the same terminal, type in :

```
./step_1_configure_ubuntu.sh
```

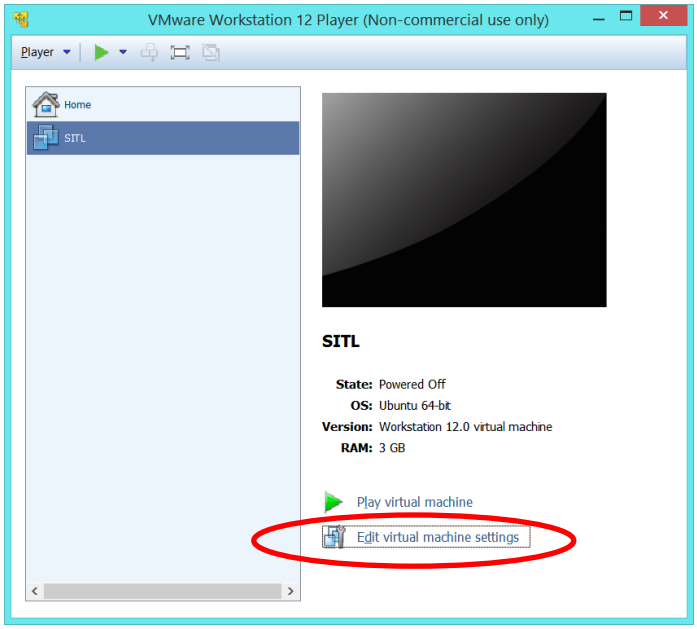
It will take 10 to 15 minutes.

Setup of shared folders

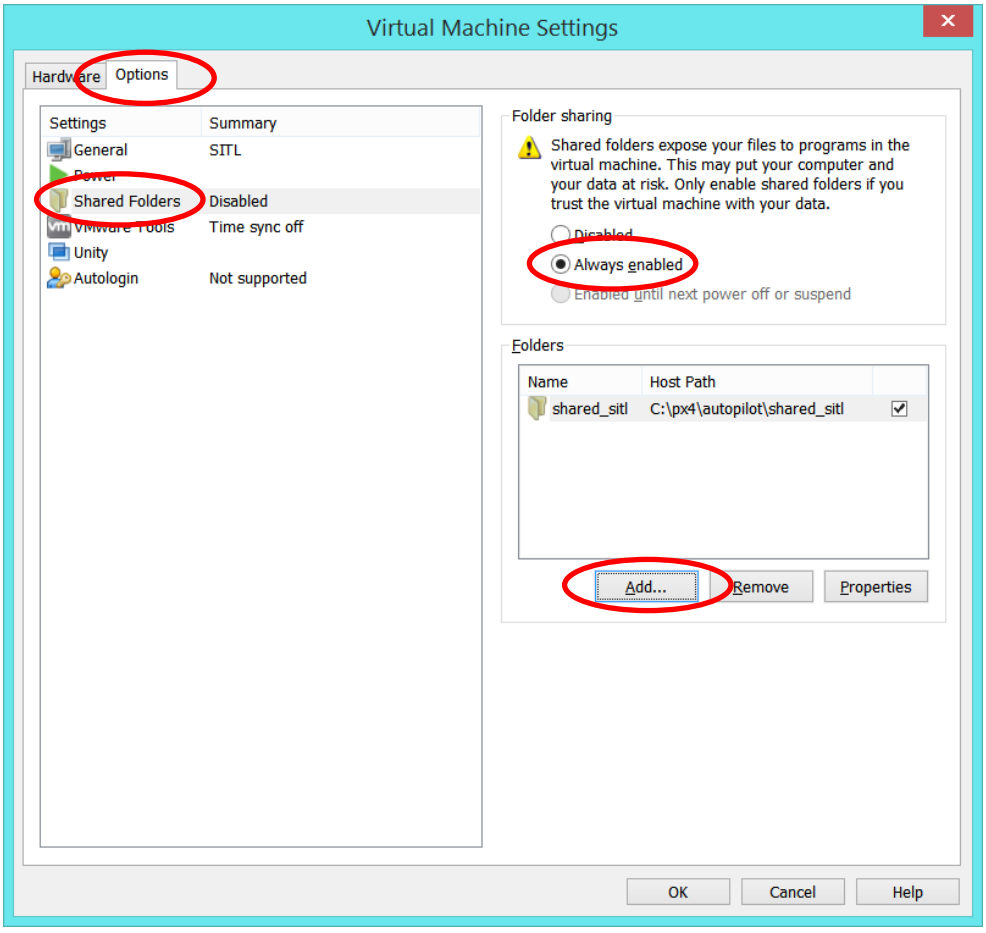
Shut down Ubuntu (cog icon on the top right corner) :



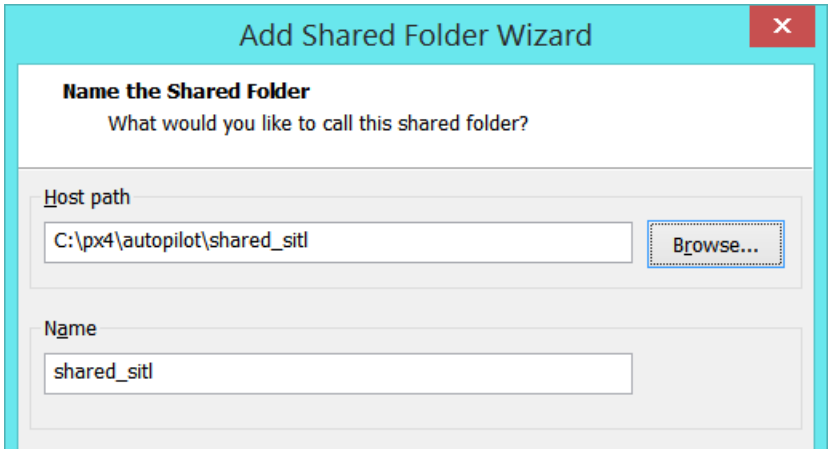
If it closed VMWare, restart it.
Click on « Edit virtual machine settings »:



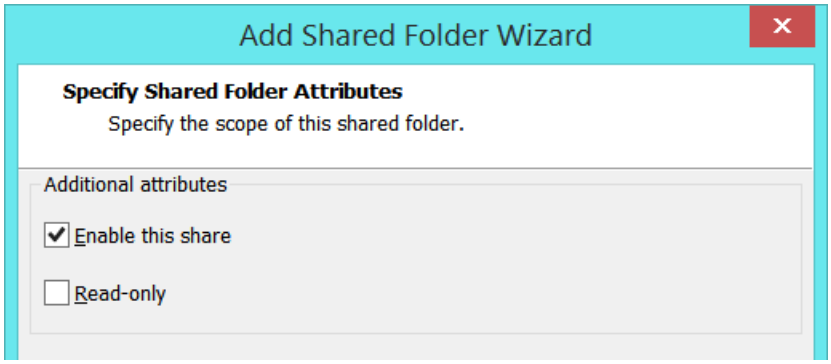
In the settings window, go to Options / Shared Folders, activate « Always enabled », and then add a new folder to share:



Select a folder that contains the ardupilot github project.
In my case, the ardupilot project was placed in *c:/px4/autopilot/shared_sitl/ardupilot*.

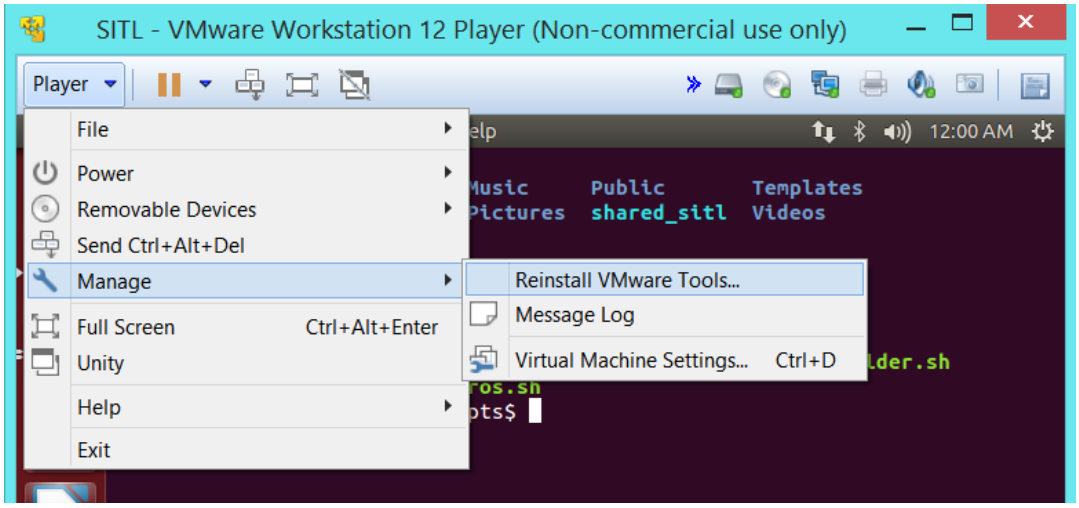


Click on *Next*

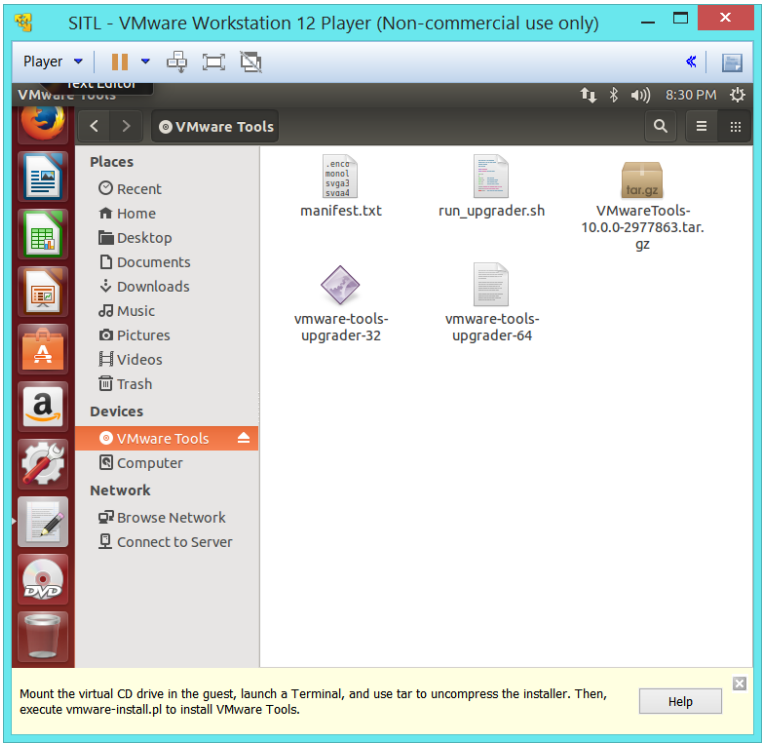


Restart the Virtual Machine (« Play »).
Log on the Ubuntu session.

Mount the CD with VMWare tools, through the menu bar: Player / Manage / ReInstall VMWare Tools



Usually Ubuntu opens up the CD directory. It is not useful, so you may close it:



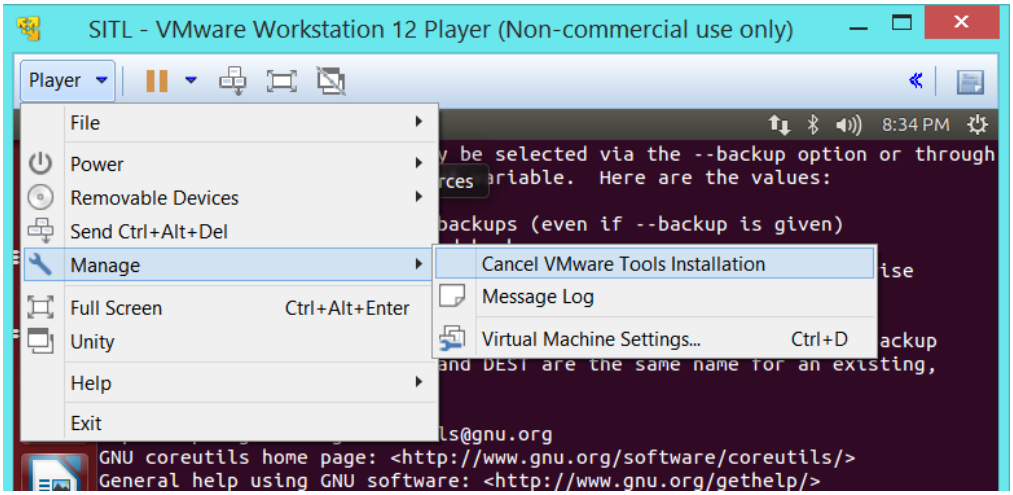
Open a new terminal and type in:

```
cd ~/Documents
./step_2_create_shared_folder.sh
```

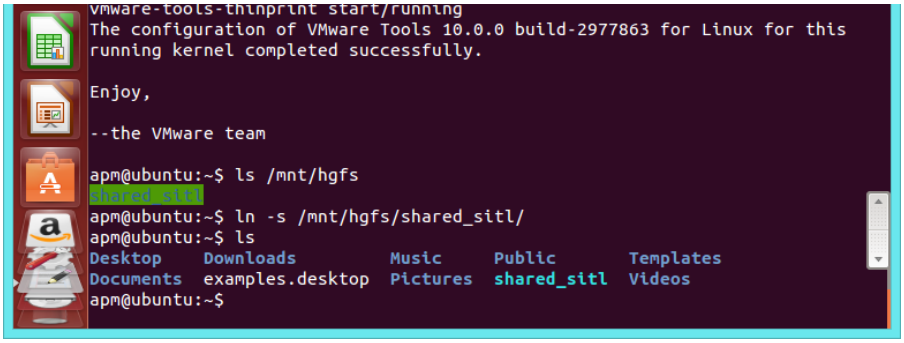
Accept all questions (paths, …) by pressing *Enter*.

> *Tip :*
> If you press *Enter* multiple times you can accept in advance the oncoming questions.

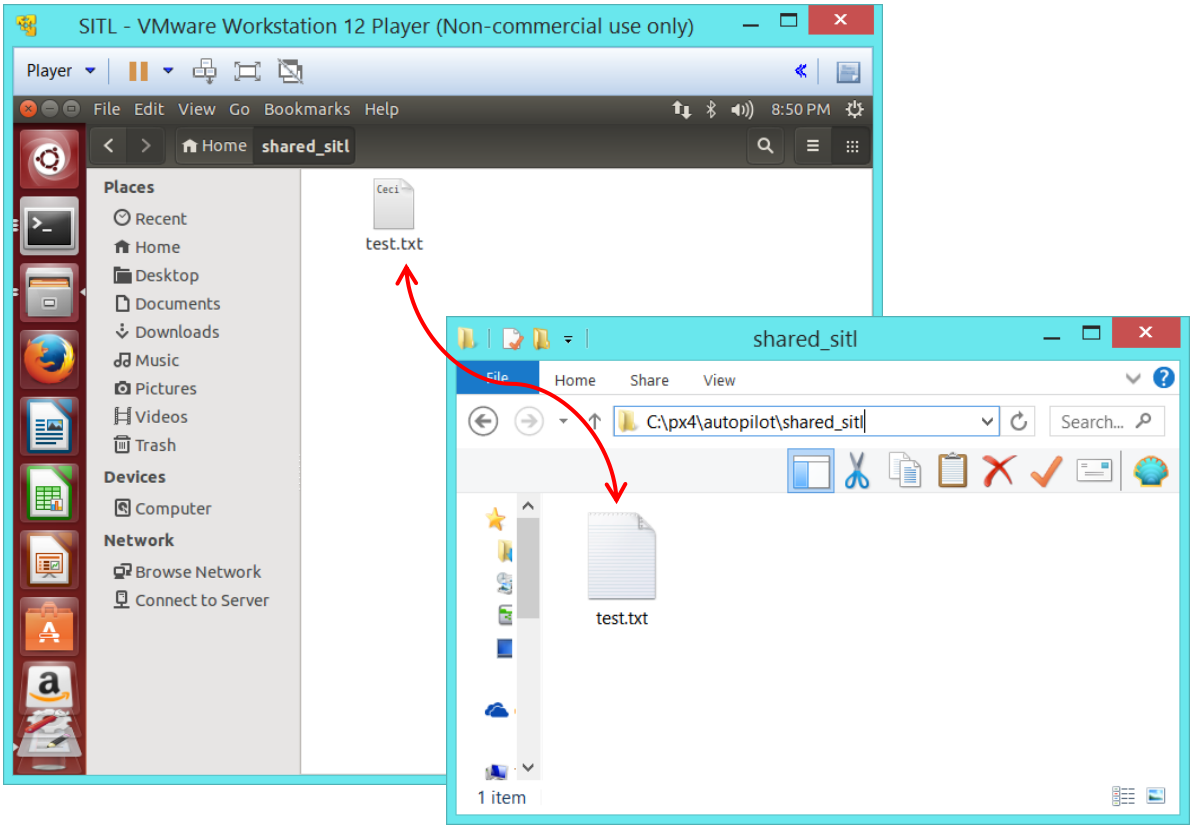
If the CD was not automatically unmounted, it is possible to do it manually on the menu bar:



The shared folder should now appear in */home/apm*.



Congratulations!
From now any file / folder placed in shared_sitl on Windows will be accessible from Ubuntu, and vice versa.


SITL - VMware Workstation 12 Player (Non-commercial use only)
Player
File Edit View Go Bookmarks Help
8:50 PM
Home
shared_sitl
Places
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Devices
Computer
Network
Browse Network
Connect to Server
test.txt
shared_sitl
File
Home
Share
View
C:\px4\autopilot\shared_sitl
Search...
test.txt
1 item

VII. Setting up the simulation

For the readers with a native Ubuntu installation who jumped directly to this section, you need to retrieve the following shell scripts:

- *step_3_install_arducopter.sh*
- *step_4_install_ros.sh*
- *step_5_create_ros_workspace.sh*

For this purpose, type in a terminal window the commands:

```
git clone https://github.com/AurelienRoy/ardupilot_sitl_ros_tutorial.git
chmod +x ./*.sh
```

It will authorize the execution of shell scripts.

1. Installation of Ardupilot

> *Note :*
> If your shared folder is not called *shared_sitl*, then you shall change the path name written in the script *step_3_install_arducopter.sh*.

Start the script *step_3_install_arducopter.sh*:

```
cd ~/Documents
./step_3_install_arducopter.sh
```

This shell script downloads Alex Buyval's version of Ardupilot, and the latest version of JSBSim, and places them in */home/apm/shared_sitl*.

> *Note :*
> Once the github branch of Alex Buyval will be integrated in Ardupilot's trunk, you will be able to switch the github project path in the script *step_3_install_arducopter.sh*.

Afterwards, log off and back on the Ubuntu session to apply the changes to the PATH variable.

2. Start the normal SITL

It is advised to run at least once the regular SITL, to make sure everything is properly setup up to this point.

On a terminal, from any directory (e.g. */home/apm*), type in:

```
./sim_vehicle.sh -v ArduCopter -j 2 -console --map
```

> *Note :*
> Thanks to the PATH variable, the system is able to find the *sim_vehicle.sh* script from anywhere.

Note :
Sometimes not launching *sim_vehicle.sh* with the « -v ArduCopter » result in a connection error, “system sleeping”.

It should open up a Mavproxy window and an APM console.

Note :
For more information on the regular SITL, and to make the UAV take off, go to : http://dev.ardupilot.com/wiki/setting-up-sitl-on-linux/

3. Installation of ROS

Execute the shell script *step_4_install_ros.sh* by typing:

```
cd ~/Documents
./step_4_install_ros.sh
```

It takes 10 to 15 minutes.

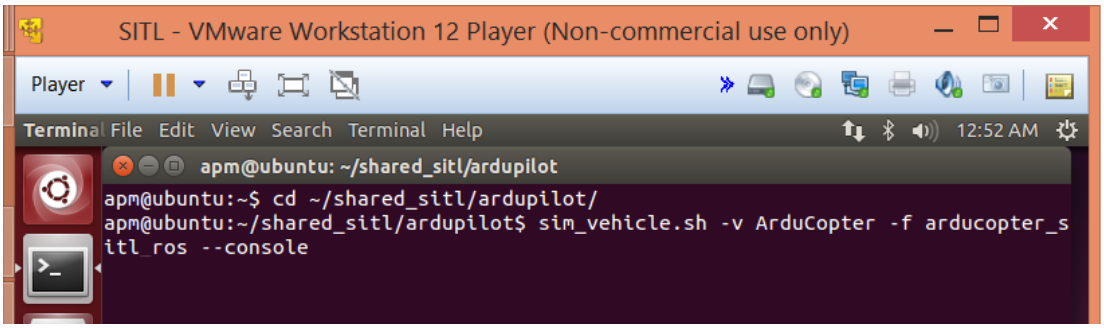
Afterwards, start the script *step_5_create_ros_workspace.sh*:

```
cd ~/Documents
./step_5_create_ros_workspace.sh
```

VIII. Launch a Gazebo simulation

Now call the Ardupilot simulation launch script:

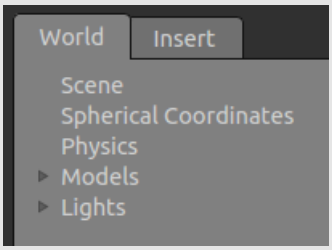
```
cd ~/shared_sitl/ardupilot
sim_vehicle.sh -v ArduCopter -f arducopter_sitl_ros --console
```



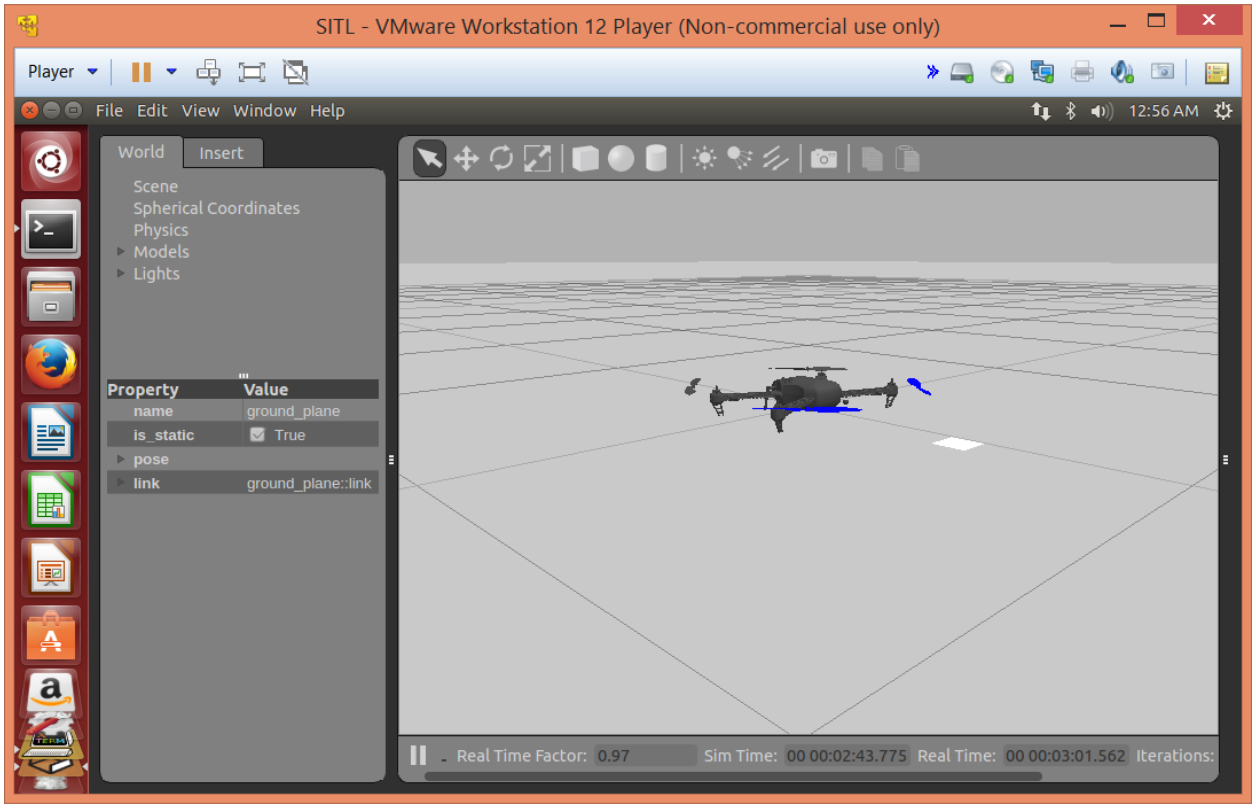
Note :
Without the argument « -v ArduCopter », the system throws back a connection refused error, process sleeping.

Note :
The work of Alex Buyval has not yet been reintegrated on the trunk, so make sure to call the *sim_vehicle.sh* modified by Alex (from his github).

Tip :
Gazebo takes some time to load the 3D display, and meanwhile just presents a black screen. You can already check that it loaded the models successfully by checking the world tree :



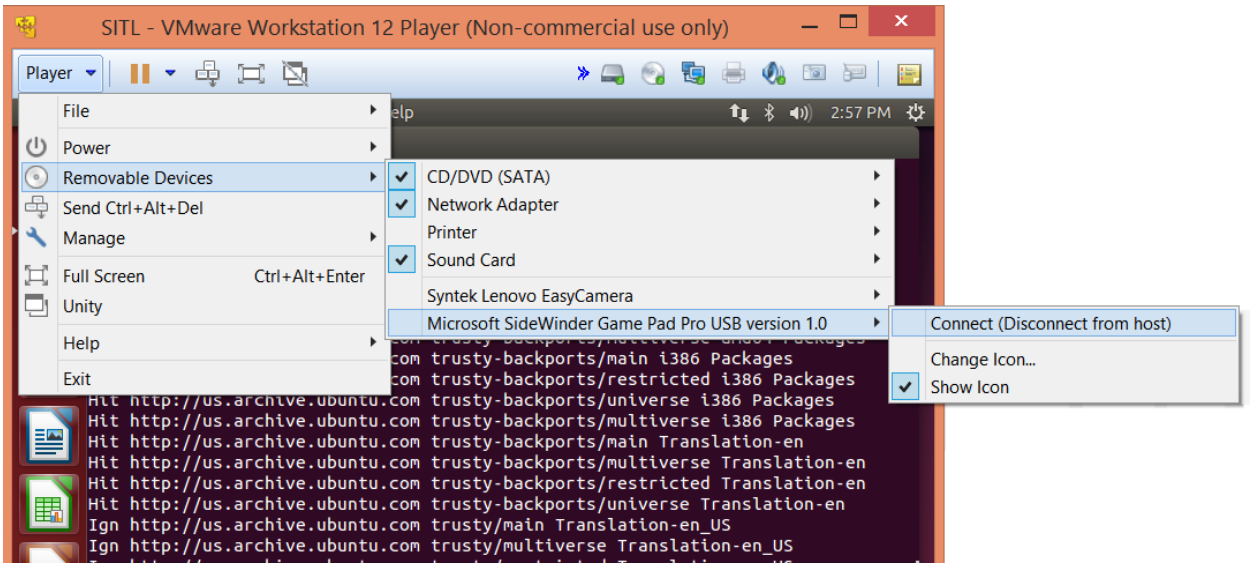
If everything works, it should open up Gazebo.


SITL - VMware Workstation 12 Player (Non-commercial use only)
Player
File Edit View Window Help
12:56 AM
World
Insert
Scene
Spherical Coordinates
Physics
Models
Lights
Property
Value
name
ground_plane
is_static
True
pose
link
ground_plane::link
Real Time Factor: 0.97
Sim Time: 00 00:02:43.775
Real Time: 00 00:03:01.562
Iterations:

IX. Install a Joystick

Check that Windows recognizes the plugged in Joystick.

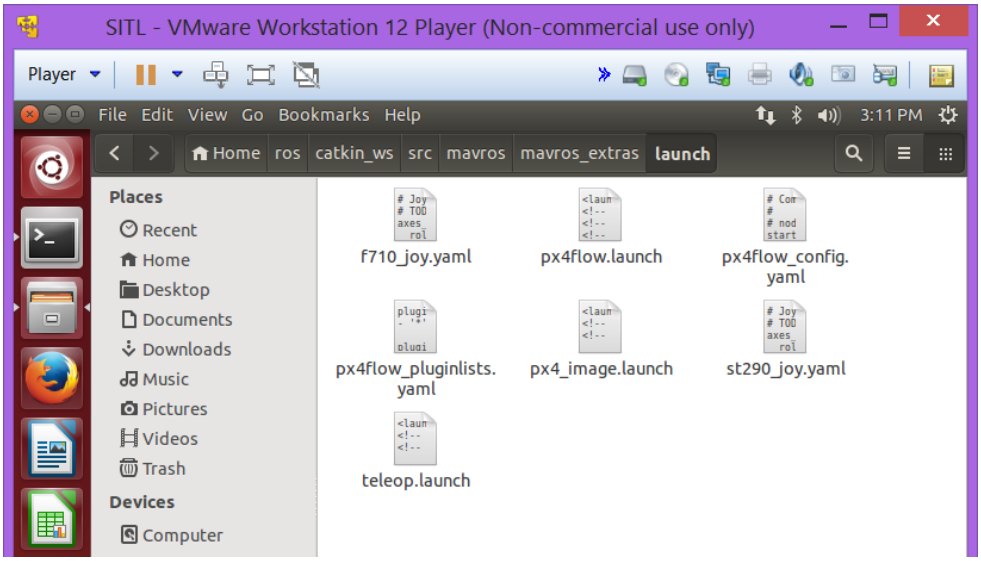
Then in VMware, assign the joystick to the virtual machine (it will disappear from Windows):



To configure the joystick on Ubuntu, you can follow the answers on:
http://askubuntu.com/questions/32031/how-do-i-configure-a-joystick-or-gamepad
https://pixhawk.org/dev/ros/sitl#joystick_configuration

Go to the directory:

```
cd ~/ros/catkin_ws/src/mavros/mavros_extras/launch
```



Create a *.yaml* file for your joystick model, on the example of « st290_joy.yaml » or « f710_joy.yaml ».
Then open up *teleop.launch* to call your new *.yaml* file.