

Installationsanleitung
MORSE
mit Middleware ROS

Tobias Hecht
Universität Tübingen
Human-Computer Interaction Lehrstuhl

Vorwort zur Installation

Dieses Tutorial behandelt die Installation der Robotersimulationssoftware MORSE 1.2 mit Middleware ROS Indigo. Als Vorlage diente die Installationsanleitung von Florian Lier (Link siehe unten).

Voraussetzung für eine problemlose Installation ist Ubuntu/Xubuntu 14.04 mit vorhandenen Sudo-Rechten.

Anmerkungen

Einzelne Befehle sind durch Leerzeilen getrennt.

1 Ros Indigo

1.1 ROS Packages auf source.list setzen

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

1.2 ROS Keys herunterladen

```
wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.
key -O - | sudo apt-key add -
```

1.3 ROS Installation

```
sudo apt-get update
```

```
sudo apt-get install ros-indigo-desktop-full
```

Der zweite Befehl nimmt einige Minuten in Anspruch, weil die 'full' Version gewählt wurde. Das heißt, alle vorhanden ROS Indigo Bibliotheken werden heruntergeladen.

Nach Abschluss des Vorgangs sollte das Verzeichnis /opt/ros/ vorhanden sein.

1.4 Rosdep (Hilfstool für ROS) initialisieren

```
sudo rosdep init  
  
rosdep update
```

1.5 Environment Setup

ROS kommt mit einem eigenen Environment. Dazu müssen diverse Pfadvariablen gesetzt werden.

```
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
```

1.6 Test

Nun sollte eine funktionierende ROS-Installation vorhanden sein. Zum Testen starten wir zunächst den ROS-Hauptprozess.

```
bash  
  
roscore
```

```
student@mickey:~$ bash  
student@mickey:~$ roscore  
... logging to /home/student/.ros/log/086520fa-6fd6-11e4-a5fd-001999f7453e/roslaunch-mickey-3249.log  
Checking log directory for disk usage. This may take awhile.  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://mickey:53222/  
ros_comm version 1.11.9  
  
SUMMARY  
=====  
  
PARAMETERS  
* /rostdistro: indigo  
* /rosversion: 1.11.9  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [3261]  
ROS_MASTER_URI=http://mickey:11311/  
  
setting /run_id to 086520fa-6fd6-11e4-a5fd-001999f7453e  
process[rosout-1]: started with pid [3274]  
started core service [/rosout]  
█
```

Abbildung 1: Ausgabe Roscore

1.7 Rosinstall

Rosinstall ist ein commandline tool für ROS, um Packages zu installieren.

```
sudo apt-get install python-rosinstall
```

2 Dependencies

MORSE benötigt einige Hilfprogramme, u.a. Python und die 3D Renderengine Blender. Hierbei ist zu beachten, dass Blender und MORSE exakt dieselbe Pythonversion verwenden.

2.1 Zugriffsrechte

Wir benötigen Lese- und Schreibrechte für das komplette Verzeichnis /opt/ros. YOURUSERNAME ist der aktuelle Benutzername. Dieser kann mit dem Befehl „id -un“ ausgegeben werden.

```
sudo chgrp -R YOURUSERNAME /opt/ros/
```

```
sudo chmod -R g+w /opt/ros/
```

2.2 Python3.3.3 Installation

```
cd /opt/ros
```

```
mkdir dependencies && cd dependencies
```

```
mkdir python && cd python
```

```
wget http://sunsite.informatik.rwth-aachen.de/python/ftp/python  
/3.3.3/Python-3.3.3.tar.bz2
```

```
tar xvf Python-3.3.3.tar.bz2
```

```
cd Python-3.3.3
```

```
./configure --prefix=/opt/ros/indigo --without-pymalloc --enable  
-shared LDFLAGS="-Wl,-rpath /opt/ros/indigo/lib"
```

```
make install
```

2.3 Blender Installation

```
cd /opt/ros/dependencies
mkdir blender && cd blender
wget https://download.blender.org/release/Blender2.70/blender-2.70-linux-glibc211-x86_64.tar.bz2
tar xvzf blender-2.70-linux-glibc211-x86_64.tar.bz2
```

Ist die Installation erfolgreich verlaufen, kann Blender nun im Verzeichnis gestartet werden.

```
cd /opt/ros/dependencies/blender/blender-2.70-linux-glibc211-x86_64/
./blender
```

3 MORSE

3.1 MORSE Installation

```
mkdir /opt/ros/morse
cd /opt/ros/morse
mkdir source && cd source
git clone https://github.com/morse-simulator/morse.git -b 1.2_STABLE
cd morse
mkdir build && cd build
cmake -DCMAKE_INSTALL_PREFIX=/opt/ros/morse/ -
      DBUILD_CORE_SUPPORT=ON -DBUILD_ROS_SUPPORT=ON -
      DPYMORSE_SUPPORT=ON -DPYTHON_EXECUTABLE=/opt/ros/indigo/bin/
      python3.3 ..
make install
```

3.2 Pfadvariablen in Datei .bashrc anpassen

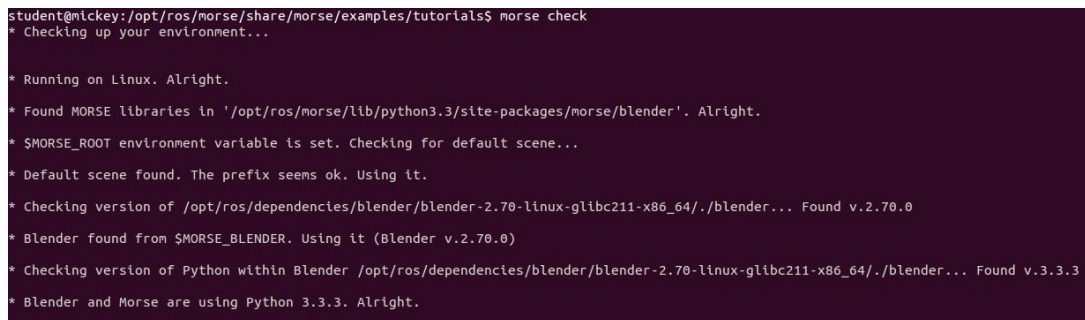
```
echo "export MORSE_ROOT=/opt/ros/morse" >> ~/.bashrc
echo "export PATH=/opt/ros/morse/bin:${PATH}" >> ~/.bashrc
echo "export MORSE_BLENDER=/opt/ros/dependencies/blender/blender
-2.70-linux-glibc211-x86_64/./blender" >> ~/.bashrc
echo "export PYTHONPATH=/opt/ros/morse/lib/python3.3/site-
packages:/opt/ros/indigo/lib/python3.3/site-packages/${
PYTHONPATH}" >> ~/.bashrc
```

3.3 Installation überprüfen

```
bash
```

```
morse check
```

Die Ausgabe sollte folgendermaßen aussehen:



```
student@mickey:/opt/ros/morse/share/morse/examples/tutorials$ morse check
* Checking up your environment...
* Running on Linux. Alright.
* Found MORSE libraries in '/opt/ros/morse/lib/python3.3/site-packages/morse/blender'. Alright.
* $MORSE_ROOT environment variable is set. Checking for default scene...
* Default scene found. The prefix seems ok. Using it.
* Checking version of /opt/ros/dependencies/blender/blender-2.70-linux-glibc211-x86_64/./blender... Found v.2.70.0
* Blender found from $MORSE_BLENDER. Using it (Blender v.2.70.0)
* Checking version of Python within Blender /opt/ros/dependencies/blender/blender-2.70-linux-glibc211-x86_64/./blender... Found v.3.3.3
* Blender and Morse are using Python 3.3.3. Alright.
```

Abbildung 2: Ausgabe Morse Check

4 Zusätzliche Dependencies

4.1 Distribute Installation

```
cd /opt/ros/dependencies
mkdir distribute
cd distribute
wget https://pypi.python.org/packages/source/d/distribute/
distribute-0.6.49.tar.gz
tar xvf distribute-0.6.49.tar.gz
cd distribute-0.6.49
python3.3 distribute_setup.py install
easy_install pip
```

4.2 PyYAML Installation

```
cd /opt/ros/dependencies/
mkdir pyyaml3.3
cd pyyaml3.3
wget http://pyyaml.org/download/pyyaml/PyYAML-3.11.tar.gz
tar xvf PyYAML-3.11.tar.gz
cd PyYAML-3.11
python3.3 setup.py install --prefix=/opt/ros/indigo/
```

4.3 rospkg/catkin Installation

4.3.1 rospkg

```
cd /opt/ros/dependencies/  
  
mkdir rospkg && cd rospkg  
  
git clone https://github.com/ros/rospkg.git  
  
cd rospkg && git checkout 1.0.20  
  
python3.3 setup.py install
```

4.3.2 catkinpkg

```
cd /opt/ros/dependencies  
  
mkdir catkin_pkg && cd catkin_pkg  
  
git clone https://github.com/ros-infrastructure/catkin_pkg.git  
  
cd catkin_pkg && git checkout 0.1.10  
  
python3.3 setup.py install
```

4.3.3 catkin

```
cd /opt/ros/dependencies  
  
mkdir catkin && cd catkin  
  
git clone https://github.com/ros/catkin.git  
  
cd catkin && git checkout 0.5.65  
  
python3.3 setup.py install
```


4.4 Pfadvariable anpassen

```
echo "export PYTHONPATH=/opt/ros/morse/lib/python3.3/  
sitepackages:/opt/ros/indigo/lib/python2.7/dist-packages/" >>  
~/.bashrc
```

5 Test

Nun sollte eine voll funktionsfähige Installation von MORSE mit ROS vorliegen. Mit folgenden Befehlen kann dies getestet werden:

```
bash  
  
roscore
```

Der Terminaltab, in dem roscore läuft, muss geöffnet bleiben, solange ROS genutzt werden soll.

Nun laden wir ein MORSE Testfile in unser MORSE Verzeichnis.

```
cd /opt/ros/morse/share/morse/examples/tutorials  
  
wget http://www.uni-tuebingen.de/fileadmin/Uni_Tuebingen/  
Fakultaeten/InfoKogni/WSI/Human-Computer-Interaction/Morse/  
morse_ros_test.py
```

Anschließend starten wir das Szenario

```
morse run morse_ros_test.py
```

Blender sollte ohne Fehlermeldungen starten und ihr solltet den Roboter mit den Pfeiltasten der Tastatur steuern können.

Nun überprüfen wir die Kommunikation zwischen MORSE und ROS. Dazu öffnen wir zunächst nochmals einen neuen Terminaltab. Anschließend werden folgende Befehle benutzt:

```
rostopic list  
  
rostopic echo /robot/pose
```

Als Ausgabe sehen wir die Positionsdaten des pose-Sensors, der dem Roboter hinzugefügt wurde.

```
pose:  
  position:  
    x: -2.20903921127  
    y: -3.95717978477  
    z: 0.853542923927  
  orientation:  
    x: -1.88557205547e-05  
    y: -4.34330031567e-05  
    z: -0.396550327539  
    w: 0.918012976646  
---  
header:  
  seq: 2312  
  stamp:  
    secs: 1415700083  
    nsecs: 682376623  
  frame_id: /map  
pose:  
  position:  
    x: -2.20903921127  
    y: -3.95717978477  
    z: 0.853542923927  
  orientation:  
    x: -1.88450721907e-05  
    y: -4.3438947614e-05  
    z: -0.396550327539  
    w: 0.918012976646  
---
```

Abbildung 3: Ausgabe Pose Sensor

6 Weitere Informationen

6.1 Mögliche Probleme

Blender und MORSE arbeiten mit unterschiedlichen Pythonversionen.
→ MORSE check liefert eine recht eindeutige Fehlermeldung.

Schreib- und Leserechte sind nicht in /opt/ros und allen Unterverzeichnissen vorhanden. → Rechte erneut für alle Unterverzeichnisse anpassen.

6.2 Weiterführende Tutorials

Tutorial von Florian Lier

<http://www.warp1337.com/content/morse-and-ros-detailed-installation-howto-ubuntu-precise>

Installationsanleitung ROS Indigo

<http://wiki.ros.org/indigo/Installation/Ubuntu>

Installationsanleitung MORSE

<http://www.openrobots.org/morse/doc/stable/user/installation.html>

Howto Ros + Ros Cheatsheet

<http://www.clearpathrobotics.com/blog/how-to-guide-ros-101/>

Morse-Tutorials

<https://www.openrobots.org/morse/doc/stable/tutorials.html>

Morse-Mailingliste

<https://sympa.laas.fr/sympa/subscribe/morse-users>