



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

AUTHORS: CHRISTIAN EIGENSEER
WALDEMAR DROZD
PASCAL BERGER
JASON CABEZUELA
28.09.2019

Speechbot Henry *- a step by step tutorial*

Step 1: VirtualBox (Only for Windows-systems)

Download and install a Virtualbox on your system.
We preferably used „VM-Ware Workstation Player“ which you can download for free at:

<https://www.vmware.com/de/products/workstation-player.html>

Step 2: Ubuntu

Download the right Ubuntu version (depends on your host system) at:

<https://ubuntu.com/download/desktop>

Note: keep in mind that the last released version of Ubuntu for 32-Bit systems is Ubuntu 16.04.1 LTS.

This project was developed under the "Bionic Beaver" version of Ubuntu.

Once you have set up your virtual environment you are ready to install ROS!

Step 3: ROS – Robotic Operating System

The following is based on the tutorial at the official wiki for ROS. You can either follow this instruction or navigate to the official website where you can find helpful tutorials as well to get in touch with ROS:

<http://wiki.ros.org/ROS/Installation>

To get started you need to decide on which ROS version you want to work. We have been using „ROS Melodic Morenia“.

1) Setup your sources.list

First of all we need to set up your host system to accept software from packages.ros.org. For doing that, open up a terminal and copy paste:

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

If there occurs a problem try to download through an alternate mirror at:

<http://wiki.ros.org/ROS/Installation/UbuntuMirrors>

II) Set up your keys

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

If you experience issues connecting to the keyserver, you can try substituting `hkp://pgp.mit.edu:80` or `hkp://keyserver.ubuntu.com:80` in the previous command.

III) Installation

First, make sure your Debian package index is up-to-date:

```
$ sudo apt update
```

To install the full version of ROS for your Desktop-Computer enter:

```
$ sudo apt install ros-melodic-desktop-full
```

Note: Replace „melodic“ by your distro-name. Also keep in mind that the installation will take some time. Depending on your internet speed up to one hour.

You can also download and install reduced versions of ROS (at the official wiki).

IV) Initialize rosdep

Before you can use ROS, you will need to initialize rosdep. rosdep enables you to easily install system dependencies for source you want to compile and is required to run some core components in ROS.

```
$ sudo rosdep init  
rosdep update
```

V) Environment setup

It is convenient if the ROS environment variables are automatically added to your bash session every time a new shell is launched:

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

VI) Dependencies for building packages

Up to now you have installed what you need to run the core ROS packages. To create and manage your own ROS workspaces, there are various tools and requirements that are distributed separately.

To install this tool and other dependencies for building ROS packages, run:

```
$ sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

Step 4: Catkin Workspace

Catkin is the official build system of ROS and the successor to the original ROS build system, rosbuilt. Catkin combines CMake macros and Python scripts to provide some functionality on top of CMake's normal workflow.

You can get more detailed information at:

http://wiki.ros.org/catkin/conceptual_overview

For the official installation guide, follow:

https://wiki.ros.org/catkin#Installing_catkin

Otherwise just open up a terminal and enter:

```
$ sudo apt-get install ros-melodic-catkin
```

Beforehand we need to source your environment like this:

```
$ source /opt/ros/melodic/setup.bash
```

Note: you need to replace „melodic“ by your distro-name.

To create and build a catkin workspace you need to enter:

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
```

The catkin_make command is a convenience tool for working with catkin workspaces. Running it the first time in your workspace, it will create a CMakeLists.txt link in your 'src' folder. Additionally, if you look in your current directory you should now have a 'build' and 'devel' folder. Inside the 'devel' folder you can see that there are now several setup.*sh files. Sourcing any of these files will overlay this workspace on top of your environment. To understand more about this see the general catkin documentation.

Before continuing, source your new setup.*sh file:

```
$ source devel/setup.bash
```

To make sure your workspace is properly overlaid by the setup script, make sure ROS_PACKAGE_PATH environment variable includes the directory you're in.

```
$ echo $ROS_PACKAGE_PATH  
/home/youruser/catkin_ws/src:/opt/ros/kinetic/share
```

Want to learn more about how to use the workspace? Follow:

http://wiki.ros.org/catkin/Tutorials/using_a_workspace

Step 5: Python, pip, rospy, programy, pygame, PyAudio, SpeechRecognition

Before continuing we need to get „pip“ for Python, rospy (Python client library for ROS), the Chat-Bot engine Program-Y, pygame (makes it able to play wave-files), SpeechRecognition, and PyAudio (Using your microphone to record audio).

I) Python

To install Python for ROS enter:

```
$ sudo apt-get install python3-dev
```

II) pip

Start by updating the package list using the following command:

```
$ sudo apt update
```

To install pip for Python 3, enter:

```
$ sudo apt install python-pip
```

The command above will also install all the dependencies required for building Python modules.

After the installation is complete you can verify the installation by checking the pip version:

```
$ pip --version
```

To prevent any errors about a „YAML“ file you can install „PyYAML“ with Python3 support:

```
$ sudo pip3 install pyyaml
```

III) rospy

You already installed rospy with your ROS- Distro.

If you need some help at writing your own code in Python or C++, we recommend the following site to get an idea of how to start:

<http://www.theconstructsim.com/difference-run-code-ros-python-cpp/>

IV) programy

For installing the Chat-Bot dependency "Program-Y" simply use the following command:

```
$ sudo pip install programy
```

If you run into any problems, please visit the official instructions at:

<https://github.com/keiffster/program-y/wiki/Install>

V) Pygame

You can either visit the official site at:

<https://www.pygame.org/wiki/GettingStarted>

or just follow these steps to install pygame for Python.

```
$ pip install pygame
```

To see if it works you can run one of the included examples like this:

```
$ python -m pygame.examples.aliens
```

VI) SpeechRecognition

You can either visit the official site at:

<https://pypi.org/project/SpeechRecognition/>

or just follow these steps to install SpeechRecognition for Python.

```
$ pip install SpeechRecognition
```

VII) PyAudio

Make sure you install PyAudio on Debian-derived Linux distributions using APT:

```
$ sudo apt-get install python-pyaudio python3-pyaudio
```

If the version in the repository is too old, you can install the latest release with:

```
$ sudo apt-get install portaudio19-dev python-all-dev python3-all-dev && sudo  
pip install pyaudio
```

Step 6.: How to start the speechbot Henry

After you have followed the instructions above, you should be ready to put our speechbot Henry into operation. You need to download or clone the git as a prerequisite:

https://github.com/stcheige/InMoov_speech

I) Start the MaryTTS-Server

You can download MaryTTS at:

<http://mary.dfki.de/download/index.html>

Make sure to download version 5.2. Select the "Runtime package". Now unpack the .zip file on your hard drive. After unpacking open the folder and navigate to the bin folder. Make sure the marytts-server file is executable. Open a terminal in this directory and type in:

```
$ ./marytts.server
```

The server should now start. If you get an error message, make sure the Java Runtime Environment (JRE) is properly installed.

II) roscore

roscore is a collection of nodes and programs that are prerequisites of a ROS-based system. You must have a **roscore** running in order for ROS nodes to communicate. To do so you need to open a terminal and enter:

```
$ roscore
```

III) Create a catkin workspace

as mentioned above. Copy the “in_moov_speech” folder from the downloaded „InMoov_speech-master”.ZIP and the contents of “inmoov_controls” into the “src.” folder of your catkin workspace. Make sure the two scripts in the scripts folder are both executable as follows:

```
$ chmod +x txt2speech.py
```

```
$ chmod +x speech2txt.py
```

Now enter:

```
$ catkin_make
```

And wait until everything is set up.

IV) Source your new setup.*sh files

```
$ source devel/setup.bash
```

V) Start the Scripts

Open a new terminal, navigate to your catkin workspace and execute the scripts each in an own terminal by typing:

```
$ rosrun in_moov_speech txt2speech.py
```

```
$ rosrun in_moov_speech speech2txt.py
```

You can start the gesture-component as follows:

```
$ rosrun inmoov_controls InMoovNodeJointStateSubNPub
```

To start the Chat-Bot module open up a new terminal, change the directory to the "scripts" folder of the module and type:

```
$ ./henry-ros.sh
```

To change the configuration of the bot, simply edit the "henry.ros.yaml" config-file in "bot_Henry/config/xnix/" and restart it.

Done!