Zumy Tutorial (cont’d)

James Lam Yi

August 19, 2015

This document provides an instruction to control multiple Zumies for navigation and network discovery in a ROS environment, and this document is based on completion of previous Zumy tutorial. We provide examples for using RQT robot steering\(^1\) and Man Joy Override\(^2\) package for driving multiple Zumies.

1 Setup

1.1 Auto Connect

Auto Connect is a ROS node in odroid machine package that allows the host automatically awakes Zumies as soon as they are connected to local network. This node detects new Zumy connections via Avahi\(^3\) displays lists of different status for detected Zumies, and tries to awake and reconnect these Zumies.

In this node, detected Zumies are divided into three lists of different status: online, alive and lost. On the status of online, Zumies are detected by the host via local network, but the host fails to receive ROS heart beat from them. On this status, the host tries to awake Zumies to alive status as soon as it receive ROS hear beat from the Zumy. On the status of alive, Zumies are connected to local network, and the host receives ROS hear beats from them. Zumies in this status have established a good ROS connection with the host. On the status of lost, Zumies are connected to local network once but currently lose connections. The host tries to search for these lost Zumies, and change their status to online as soon as they are reconnected to the host.

Following steps are setup and tests for launching Auto Connect:

1. Install dependencies

```
sudo apt-get install nmap python-nmap
```

2. Download the package to the src directory of a ROS workspace with

```
git clone https://github.com/jlamyi/odroid_machine.git
```

and run catkin_make from the workspace. If you have downloaded and installed Odroid machine, skip this step.

3. Launch Auto Connect with

```
roslaunch odroid_machine auto_connect.launch
```

If it is successfully launched, the terminal repeatly shows "Online Detector: scanning...".

If the host connects to a Zumy successfully, the wheels of the Zumy will move a little bit. Also, terminal should output "... done launching nodes", but it is probably hard to be seen since there are too many terminal outputs for launching Auto Connect.

\(^1\) http://wiki.ros.org/rqt_robot_steering
\(^2\) https://github.com/jlamyi/man_joy_override
\(^3\) http://www.avahi.org/
4. To confirm if a Zumy has a solid ROS connection with the host, Auto Connect provides lists for tracking Zumies status. Simply check out topic `/online_detector/alive_robots` for alive Zumies. If the host name of a Zumy is in the list of this topic, this Zumy has set up a solid ROS connection. Also, you can check out `/online_detector/online_robots` and `/online_detector/lost_robots` for Zumies on status of online and lost.

5. If it hangs with terminal output "waiting for connection...", no Zumy is detected by the host. In this case, it waits until the first Zumy is detected by the host. If it waits for too long for connections, relaunch Auto Connect.

6. If a Zumy fails to connect to the host, first, test its local network connection with

   ```
   ping "host name of the failed Zumy".local
   ```

   If terminal shows

   ```
   ping: unknown host "the host name of the failed Zumy"
   ```

   Restart the Zumy, and test its local network connection again with the command shown above.

   Also, rqt_graph should display Figure 1, which shows that Auto Connect is subscribing topic from Avahi for new connection detection, and subscribing Zumy heart beat to identify alive Zumies. Then it publishes lists of online, alive and lost status.

   ![Figure 1: ROS node graph of Auto Connect](http://wiki.ros.org/rqt_graph)

2  Drive

2.1  RQT robot steering

Similar to previous tutorial, RQT robot steering can be used to drive Zumies by publishing topics to them. Once a Zumy establishes a good ROS connection with the host, run RQT robot steering with

```
rosrun rqt_robot_steering rqt_robot_steering
```

Modify publish topic name with the Zumy you want to drive, and modify the namespace of this topic if you want to drive other Zumies that are connected to the host. For example, to drive Zumy "zumy3", modify your topic to

```
/zumy3/cmd_vel
```
2.2 Man Joy Override

Man Joy Override is a ROS package developed in Fearing Biomimetics Millisystem Laboratory to allow manual override of commanded velocities with joysticks. It contains a multiplexer so that the user can switch to control different Zumies with the same joystick (currently it supports up to four Zumies with one joystick).

Figure 3 shows the software architecture of Man Joy Override. It reads joystick input by subscribing topic Joy, and decides to which Zumy the host should send commands by subscribing topic /onlineDetector/alive_robot.

![Diagram of using Man Joy Override for driving Zumy](image)

Figure 3: Diagram of using Man Joy Override for driving Zumy

This package can be downloaded with this command

```bash
git clone https://github.com/jlamyi/man_joy_override.git
```
When the download is finished, compile the package from the workspace by running `catkin_make`. After it is compiled, first, plug in Logitech Rumble Gamepad F510 driver to the host (This type of driver was used for Zumy development in the lab). Then, launch Man Joy Override with the command shown below (This launch file includes Auto Connect and Joy node. This should be the only command typed in terminal, and you should not separately launch Auto Connect).

```bash
roslaunch man joy override zumy.launch
```

If you successfully launch Man Joy Override, `rqt_graph` should display the graph similar to Figure 4. Man Joy Override subscribes topic `joy` and `/online_detector/alive_robots` from Joy and Auto Connect. It publishes velocity commands to Zumies while Auto Connect subscribes heart beat from Zumies.

![ROS node graph of Man Joy Override](image)

Figure 4: ROS node graph of Man Joy Override

Figure 5 shows a picture of Logitech Rumble Gamepad F510 driver. In Man Joy Override, at most four Zumies can be controlled with one joystick, and the four colored buttons on the right map to four different Zumies. Start with the blue button, these buttons are clockwisely mapped to robot 0, 1, 2 and 3. The mapping relationship can be found in topic `/online_detector/alive_robot`. The first four Zumies on this list are mapped to robot 0, 1, 2 and 3.

To select a Zumy, you need to press the button mapped to the Zumy before driving it. The joystick on the right is the only joystick needed to drive a Zumy. Moving it up and down allows to drive a Zumy forward and backward, and moving it left and right allows to turn a Zumy left and right.

![Logitech Rumble Gamepad F510 driver](image)

Figure 5: Logitech Rumble Gamepad F510 driver