# Setting Up and Running the Kobuki Robot

# One Time Setup

## The Laptop / Controlling Computer / Ubuntu

The Kobuki robot runs ROS Kinetic. ROS Kinetic, in turn, requires Ubuntu 16.04. There is a ROS node called kobuki_node that is used to control the robot so you will need to select a computer to run ROS and the kobuki_node. If the computer chosen has low power or low memory, it is recommended to use Ubuntu 16.04 Server Edition since this doesn't have a desktop and requires less resources to run. If the computer chosen has enough resources, the desktop is nice to have so that the ROS GUI tools can be used. However, only the command line will be used so a desktop is not necessary. NOTE: This guide assumes that the controlling computer has WIFI capabilities. Ubuntu has good documentation on how to install the OS.

The installation of Ubuntu 16.04 is generic but afterward, two configuration changes are needed to make it easier to run the controlling computer on the robot without wires. The first change was to disable networking on start up. Without this, there is a 5 minute delay during the boot process while the computer looks for a network connection. The command to disable networking is as follows:

```
sudo systemctl disable networking
```

The second configuration change was to disable suspend when the lid is closed. This is so the laptop will run with the lid closed while on the robot. To do this, follow this procedure:

- Edit the file **/etc/systemd/logind.conf** (need to use sudo)
- Copy the following two lines from lower in the file to the top just under [Login]
  - **`#HandleLidSwitch=suspend`**
  - **`#LidSwitchIgnoreInhibited=yes`**
- Change the two copied lines to the following (note that the # is removed):
  - **`HandleLidSwitch=ignore`**
  - **`LidSwitchIgnoreInhibited=no`**

The laptop/controlling computer is set up as a wifi hotspot so that another computer can connect to it directly without having to connect network wires to the robot. Later you will see how to start the hotspot which is set up each time the system is started.

## The Laptop / Controlling Computer / ROS

ROS was installed onto the controlling computer following the ROS installation tutorial. No special changes were needed.

## Kobuki Software

The kobuki_node and other kobuki software were installed following the kobuki / ROS installation tutorial. No other special changes were needed.

# To Run the Robot

1. Boot the controlling computer and login.
2. At the command prompt of the computer, enter the command to start the Wifi hotspot. It takes about 15 - 20 seconds but will say Device 'wlan0' successfully activated… NOTE: you will need to determine the name the system gives the WIFI connector. This name was wlan0 on the Dell Mini Laptop used for initial testing.
   - **`nmcli d wifi hotspot ifname wlan0 ssid kobuki password mocap123`**
3. On a separate laptop or desktop to be used for remote connection, connect to the kobuki WIFI network.
4. On remote computer, start a terminal (or PuTTY shell) and connect to the controlling computer with the following command and enter the laptop password:
   1. **`ssh <userID>@<IP of controlling computer>`**
5. Repeat the previous step to gain three more shells into the controlling computer.
6. In the first shell, run **`roscore`**
7. Connect the USB cable between the controlling computer and the robot and power up robot.
8. In the second shell, start kobuki node software with:

   **`1. source ~/ros_ws/devel/setup.bash`**

   **`2. roslaunch kobuki_node minimal.launch –screen`**

   3. It takes about 30 seconds so wait for rising tone sequence on robot and for the shell to say Kobuki : initialized (for the technically curious, kobuki_node will create /dev/ttyUSB0 and /dev/kobuki devices to talk over)

## Drive the Kobuki

The Kobuki robot represents the control system of the CANSAT. Therefore, in general, the controlling computer should be able to receive commands and drive the robot. During the first stage of the project, several test programs were implemented to control the robot.

- **ros_ws/src/drive_kobuki/scripts/drive_kobuki.py** – This is a ROS script that drives the robot through a canned path. The script sends twist commands to the kobuki_node. Forward linear velocity is always the same. Each time the left or right function is called, it increments the angular velocity by an increment. The stop function zeros both forward and angular velocity. A

shutdown callback and a bumper event callback are implemented to make sure the robot stops if it hits something. To execute this program execute **rosrun drive_kobuki drive_kobuki.py**.

- **ros_ws/src/drive_kobuki/scripts/semicircle1.py** – This is ROS script that causes the robot to move in a canned path of a semicircle through 180 degrees with a radius of about 6 feet. It also prints out odometry information and writes that information to a file. To execute this program execute **rosrun drive_kobuki semicircle1.py**.
- **ros_ws/src/drive_kobuki/scripts/drive_kobuki2.py** – This is a ROS script that implements a ROS listener to subscribe to the /chatter topic and receive text commands from another ROS node. In this case, the publishing node was running on an Arduino micro controller connected to the controlling computer via USB cable. The listener was implemented in a node named drive_kobuki2.py and the Arduino node was implemented in arduino_node.ino. These two can be used as examples to implement future features.
- **arduino_node.ino** – This Arduino sketch implements a ROS node that publishes the text words, “forward”, “right”, “left”, and “stop” to the rosserial topic chatter. Initially this node was implemented to follow the same canned patch as drive_kobuki.py. However, the timing on the steps was not very good and it didn’t produce a smooth path. At the end of the semester, code was added to receive I2C commands from the command module to implement the left, right, and straight commands. Some limited testing was done but the timing issues were not completely worked out.

To drive the Kobuki using the Arduino to control it, follow this procedure:

9. In the third shell, start the node that will implement the listener and drive the robot:

   **1. source ~/ros_ws/devel/setup.bash**

   **2. rosrun drive_kobuki drive_kobuki2.py**

10. In a fourth shell to the Dell Mini, make sure you know which serial port (USB) the Arduino is on then (could be /dev/ttyACM0 or /dev/ttyACM1) start rosserial which forwards the text coming in from the serial port to the topic and the listener with:

    **1. rosrun rosserial_python serial_node.py /dev/ttyACM0**

11. Plug the Arduino up to the controlling computer with a USB cable. It's software will automatically start running.

## Other Misc. Notes

To run teleop mode, in the third shell to the Dell Mini, type the command:

**roslaunch kobuki_keyop safe_keyop.launch –screen**

To check amount of battery left on the Dell Mini:

**upower -i *org/freedesktop/Upower/devices/battery_BAT1***