SCUTTLE ROBOT USING ROS

Running a SCUTTLE robot with ROS (Robot Operating System) in the real world can be a challenging but rewarding experience. ROS is an open-source software framework that provides a collection of libraries and tools for developing and controlling robots. It is widely used in robotics research, industrial automation, and other applications.

To run a SCUTTLE robot with ROS, you will need to install ROS on a computer or embedded device, and then connect the robot to the computer using a communication interface, such as USB or Ethernet. You will also need to install the necessary ROS packages and drivers for the SCUTTLE robot, which can be found on the ROS wiki or other online repositories.

Once the robot is connected to the computer and the ROS packages are installed, you can use the ROS tools and libraries to control the robot's movements and behaviour. This typically involves writing ROS nodes, which are programs that communicate with the robot's sensors and actuators, and with other nodes in the ROS system.

Some of the benefits of running a SCUTTLE robot with ROS include the ability to leverage the large and active ROS community, which provides a wealth of resources and support for robotics development. ROS also provides powerful tools for visualization, simulation, and debugging, which can help you test and refine your robot's behaviour before deploying it in the real world.

Install ROS on Raspbian OS

Install git

sudo apt-get install git

Install zsh

sudo apt-get install zsh

Make zsh default

chsh -s /bin/zsh

Install oh-my-zsh

sh -c "$(wget https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh -O -)"

Install autosuggestions.

git clone git://github.com/zsh-users/zsh-autosuggestions $ZSH\_CUSTOM/plugins/zsh-autosuggestions

Install highlighting

git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH\_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting

Update the plugins line in the .zshrdc file

plugins=(git zsh-autosuggestions zsh-syntax-highlighting)

Install ROS

Add the apt repository

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

add the key

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

Update

sudo apt update

Install dependencies

sudo apt-get install -y python-rosdep python-rosinstall-generator python-wstool python-rosinstall build-essential cmake

run rosdep

sudo rosdep init

rosdep update

make a catkin workspace

mkdir ~/ros\_catkin\_ws

cd ~/ros\_catkin\_ws

Use the ROS install generator to create the right install

rosinstall\_generator robot navigation slam\_gmapping laser\_pipeline perception --rosdistro noetic --deps --wet-only --tar > noetic-ros\_comm-wet.rosinstall

wstool init src noetic-ros\_comm-wet.rosinstall

rosdep install -y --from-paths src --ignore-src --rosdistro noetic -r --os=debian:buster

increase swap from 100Mb to 1Gb

sudo dphys-swapfile swapoff

sudoedit /etc/dphys-swapfile

sudo dphys-swapfile setup

sudo dphys-swapfile swapon

Build all the ROS things and install them in /opt/ros/noetic

sudo src/catkin/bin/catkin\_make\_isolated –install -DCMAKE\_BUILD\_TYPE=Release –install-space /opt/ros/noetic -j1 -DPYTHON\_EXECUTABLE=/usr/bin/python3

**install opencv**

Install dependencies – For images

sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

Install dependencies – For video

sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

sudo apt-get install libxvidcore-dev libx264-dev

More dependencies

sudo apt-get install libgtk2.0-dev

Optimization libraries

sudo apt-get install libatlas-base-dev gfortran

Install pip3

sudo apt-get install python3-pip

Install numpy

pip install numpy

The next step is to create a [ROS workspace](http://wiki.ros.org/catkin/workspaces), which is essentially a directory with some specific subdirectories. You can create this directory anywhere and it can be called anything. For this guide we'll assume that this directory is called scuttle\_ws and that it is created in the home directory for the pi user, i.e. ~/scuttle\_ws or `/home/pi/scuttle\_ws/.

mkdir -p <WORKSPACE\_DIRECTORY\_PATH>/src

cd <WORKSPACE\_DIRECTORY\_PATH>/src

You can do this by running the following command lines in a terminal. Make sure your current directory is the src directory.

git clone -b noetic https://github.com/scuttlerobot/scuttle\_bringup.git

git clone -b noetic https://github.com/scuttlerobot/scuttle\_description.git

git clone -b noetic https://github.com/scuttlerobot/scuttle\_driver.git

git clone -b noetic https://github.com/scuttlerobot/scuttle\_navigation.git

git clone -b noetic https://github.com/scuttlerobot/scuttle\_slam.git

git clone -b indigo-devel https://github.com/scuttlerobot/teleop\_twist\_joy.git

Once you have cloned the repositories you need to 'build' the workspace. This is done by calling 'catkin', the ROS build system.

catkin\_make

After running catkin\_make once you can source the workspace setup file. This allows the ros commands to find your packages. Depending on the terminal you are using you can source the .bash file or the .zsh file.

source devel/setup.bash or source devel/setup.zsh

In the end you should have a ROS workspace that looks like this

<WORKSPACE\_DIRECTORY>

build

devel

src

rplidar\_ros

scuttle\_bringup

scuttle\_description

scuttle\_driver

scuttle\_navigation

scuttle\_slam

teleop\_twist\_jo