**USERNAME: openquad**

**PASSWORD: openquad**

# **Installing ubuntu mate on raspberry pi:**

1. Download ubuntu mate for raspberry pi from the official website. Size would be around 1.1 Gb.

<https://ubuntu-mate.org/raspberry-pi/>

1. Use SD card(16 GB preferable) formatter to format the SD card and then balena etcher or win32diskimager to write the downloaded image file to the SD card.
2. Using a keyboard monitor and mouse install the ubuntu mate on raspberry pi.
3. Now restart the network manager in the command line if you don’t see the Wi-Fi symbol or if you are unable to connect to Wi-Fi.

When it doesn't show wifi or doesn’t connect to wifi

> sudo service NetworkManager restart

If this doesn’t work then

> nohup nm-applet &

5) Now connect to the wifi you want to connect

Installing the wpasupplicant

> sudo apt-get install wireless-tools wpasupplicant

6) The write about the wifi you are going to use in the wpasupplicant file. Whenever you want to change the default Wi-Fi to which raspberry pi connects while booting you have to make changes to this file.

Opening the wpasupplicant file

> sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

Write the following in the file

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=ISO 3166-2:IN

network={

ssid="ARK"

psk="hellobitches"

key\_mgmt=WPA2-PSK

}

**Note:** ssid is the name of your network i.e. name of the Wi-Fi, psk is the password of the Wi-Fi and key\_mgmt is the type of encryption like WPA-PSK or something like that.

Once done copying to save the file press CTRL+X then press Y and then enter.

7) Installing xrdp for remote desktop connection and updating and upgrading

> sudo apt-get install xrdp

Also updating and upgrading

> sudo apt-get update

> sudo apt-get upgrade

**Lock problems:**

If there are lock errors while using sudo apt-get then execute the following commands

> sudo rm -rf /var/lib/apt/lists/lock

> sudo rm /var/lib/dpkg/lock

**After doing all these steps make sure to make a copy of the image using win32diskimager**

# **Installing ROS on ubuntu mate:**

Use this link as reference:

<https://www.intorobotics.com/how-to-install-ros-kinetic-on-raspberry-pi-3-ubuntu-mate/>

1. Follow steps 1 and 2 from the link and use close button. After you press the close button it will ask you to download. Don’t download.
2. > sudo -i

> echo deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main > /etc/apt/sources.list.d/ros-latest.list

> exit

1. > wget http://packages.ros.org/ros.key -O – | sudo apt-key add –

> sudo apt-key del 421C365BD9FF1F717815A3895523BAEEB01FA116

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

1. > sudo apt-get update

> sudo apt-get install -f (Run only if there are unmet dependencies)

1. > sudo apt-get install ros-kinetic-desktop-full
2. > sudo rosdep init

> rosdep update

Keep on updating if it shows an error

1. > echo source /opt/ros/kinetic/setup.bash >> ~/.bashrc

> source ~/.bashrc

1. > mkdir -p ~/catkin\_workspace/src

> cd catkin\_workspace/src

> catkin\_init\_workspace

> cd ~/catkin\_workspace/

> catkin\_make

1. > source ~/catkin\_workspace/devel/setup.bash

> echo source ~/catkin\_workspace/devel/setup.bash >> ~/.bashrc

1. To check installation

> export | grep ROS

You can also check via turtlesim simulator

1. Install git and usb cam node

> sudo apt install git

Go to source folder of catkin workspace

> git clone <https://github.com/ros-drivers/usb_cam.git>

> cd ..

> catkin\_make

> sudo apt-get install gedit

1. Find the IP address

> hostname -I

1. Make the following changes in bashrc file

> gedit ~/.bashrc

Make sure these two lines are present according to the requirement. Here RMI\_PROJECT is the name of the workspace. Generally it is catkin\_ws

source /opt/ros/kinetic/setup.bash

source ~/RMI\_PROJECT/devel/setup.bash

For local machine use add:

export ROS\_HOSTNAME=localhost

export ROS\_MASTER\_URI=http://localhost:11311

For multiple machine communication(through Wi-Fi)

export ROS\_IP=192.168.43.161

export ROS\_MASTER\_URI=http://192.168.43.161:11311

ROS\_IP is the local IP which can be found in step 12 in local computer

ROS\_MASTER\_URI is the IP of the master and should be common in both the machines. It is basically local IP of the master.

1. Modify the launch file of Usb cam node depending on the number of cameras

> cd ~/catkin\_ws/src/usb\_cam/launch

> gedit usb\_cam-test.launch

Replace the entire code in the launch file with the following and save

<launch>

<node name="usb\_cam1" pkg="usb\_cam" type="usb\_cam\_node" output="screen" >

<param name="video\_device" value="/dev/video1" />

<param name="image\_width" value="640" />

<param name="image\_height" value="480" />

<param name="pixel\_format" value="yuyv" />

<param name="camera\_frame\_id" value="usb\_cam" />

<param name="io\_method" value="mmap"/>

<remap from="image" to="/usb\_cam/image\_raw1"/>

<param name="autosize" value="true" />

</node>

<node name="usb\_cam0" pkg="usb\_cam" type="usb\_cam\_node" output="screen" >

<param name="video\_device" value="/dev/video0" />

<param name="image\_width" value="640" />

<param name="image\_height" value="480" />

<param name="pixel\_format" value="yuyv" />

<param name="camera\_frame\_id" value="usb\_cam" />

<param name="io\_method" value="mmap"/>

<remap from="image" to="/usb\_cam/image\_raw0"/>

<param name="autosize" value="true" />

</node>

</launch>

1. Run roscore on the master and run the following command on raspberry pi

> roslaunch usb\_cam usb\_cam-test.launch

1. Run rqt on the master

Go to plugins>visualization>imageview

Do it twice for 2 camera feed

Select the ros topic to view the output. Generally image\_raw/compressed

1. For compressing the image from master’s side

Go to plugins>configuration>dynamic

Select jpeg format and the appropriate topic. Always work with compressed topic.

# Using vnc with Linux

<http://mitchtech.net/vnc-setup-on-raspberry-pi-from-ubuntu/>