# Requirements

1 x Raspberry pi 3 B

1 x OKO Lİdar

# Ubuntu Mate Installation

1. We need to install an operating system to raspberry pi 3 beta.

* Equipment list

1 X Raspberry pi 3 beta

1 X LCD monitor

1 X mouse and keyboard

1 X microSD card

2. Plug microSD into the computer

3. Download the Ubuntu mate 16.04 version for the raspberry pi from the given website

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

4. You can use dd command to write the image file to the microSD but we prefer ddrescue

* $ sudo apt-get install gddrescue xz-utils
* $ cd ~/Downloads
* $ unxz ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz
* $ sudo ddrescue -D --force ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img /dev/<your\_microSD\_path>

You can check your microSD path by running the command below

* $ lsblk

5. Plug your microSD into the Raspberry pi 3 , start your operating system make the installation.

# Ros Installation

* You can visit the webpage below to see more information.

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

1. Setup your sources.list

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

2. Setup your keys

* $ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116

3. Update your Debian package index

* $ sudo apt-get update

4. Install the Desktop-Full Install repository by using the command given below.

* $ sudo apt-get install ros-kinetic-desktop-full

5. Inıtialize rosdep

* $ sudo rosdep init
* $ rosdep update

6. Add ros environment to your bash session

* $ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
* $ source ~/.bashrc

7. Install additional dependencies for building ros packages

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

# Setup the Workspace

1. Clone the repository from the given link,

* $ git clone git@github.com:samialperen/oko\_slam.git

2. Go into the ros\_ws directory

* $ cd ros\_ws

3. Run catkin\_make command to make your ros packages

* $ catkin\_make

# How Serial Communication Package Works

## Kamu Robotu Lidar Bridge

We need a communication method between the Lidar and the Ros environment. kamu\_robotu\_lidarbridge is implemented to convert serial data into laser\_scan messages. We have used a cross-platform library to interface with serial ports.

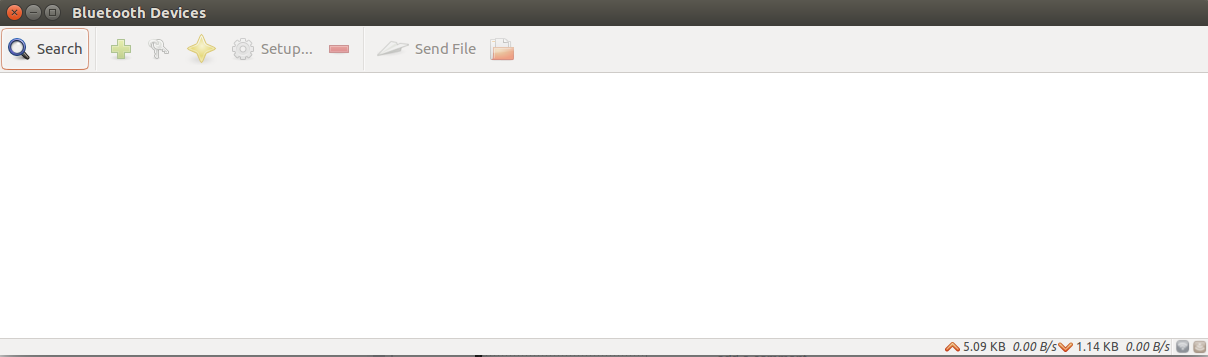
1. Install serial library which is developed by William Woodall(Remember that is has been already installed into the oko\_slam repository.)

* $ git clone <https://github.com/wjwwood/serial.git>
* $ make
* $ make test
* $ make doc
* $ make install

2. Install bluetooth manager.

* $ sudo apt-get install blueman

3. Open the Bluetooth Manager from the Applications you will see a program as given below



4. Pair the bluetooth module of the OKO Lidar with your raspberry pi

* Click to the search button and find the bluetooth module with the parameters given below

Name : OKO

Password : 5216

Baud Rate : 9600

Stop Bit : 2

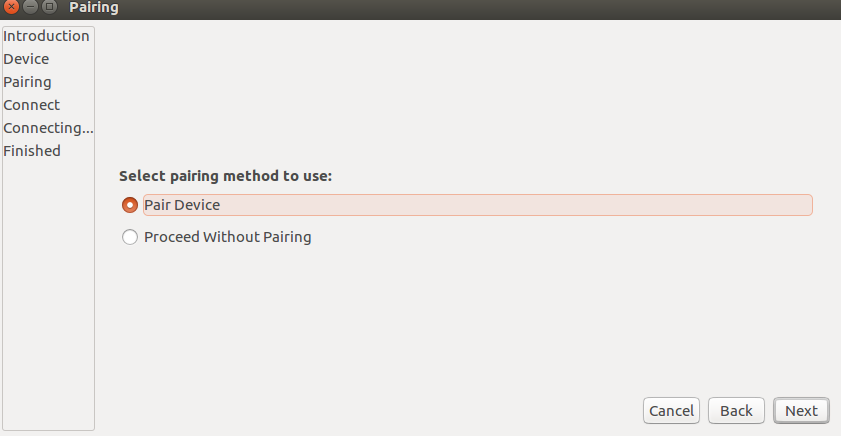
Pairing : None

Address : 21:13:1F:A4

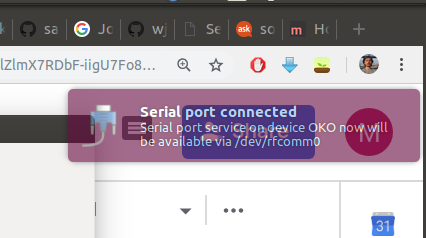
Version : 3.0-20170601

Role : Slave

* Click to the setup button and you will see a figure given below, then click to the next button



* Enter the password given above
* Choose the serial port option
* Then you will see a notification given below which indicates the serial port path as /dev/rfcomm0

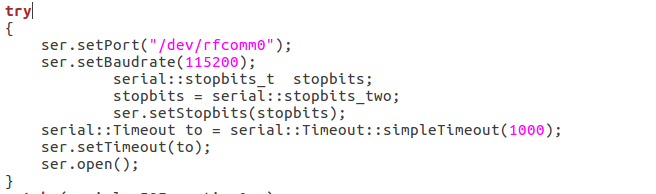


5. Modify the port parameters in the kamu\_robotu\_lidarbridge node

* $ cd <your\_download\_path>/oko\_slam/ros\_ws/src/kamu\_robotu/kamu\_robotu\_lidarbridge/src
* $ gedit laser\_scan\_publisher

Change the parameters given below then go into the ros\_ws directory and make a catkin\_make command

* $ catkin\_make



6. Run the rosnode to obtain laser\_scan data

* $ rosrun kamu\_robotu\_lidarbridge laser\_scan\_publisher

## Test with Bag Files

1. Add your ros\_ws as a ros environment

* $ cd <your\_download\_path>/oko\_slam/ros\_ws
* $ source devel/setup.bash

2. Open a new terminal and start our launch file fo the hector slam

* $ roslaunch hector\_slam\_launch oko\_hector\_launcher.launch

3. Go to bagfile folder

* $ cd bagfiles

4. Start to publish stored laser\_scan data

* $ rosbag play easy\_map.bag --clock