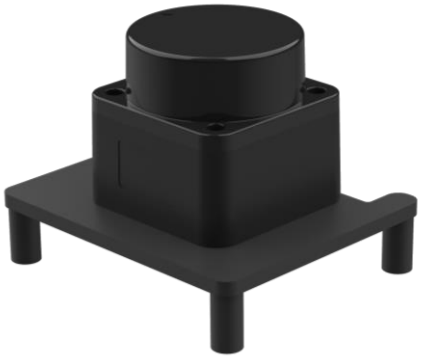
# SHENZHEN LDROBOT CO., LTD.

# DTOF LiDAR LD06 SPECIFICATION

| | | |
|---|---|---|
| Product Name | : | DTOF LiDAR_LD06 |
| Description | : | DTOF COAXIAL BRUSHLESS LiDAR with Raspberry pi SBC (based on Raspbian OS 32-bit kernel version 5.4) |
| Date | : | 2021-12-17 |
| File No | : | LD-LD06-DS-REV_2.5_EN |



乐动机器人
MOVE SMARTER

# 目录

# 1. DEVELOPMENT KIT

The development kit of DTOF LiDAR_LD06 is an accessory tool (includes bracket & DTOF module & Uart cable & Assembly screws) provided for robotic device development or performance evaluation of sensor products, and for the educational purpose use of robotic device motion control and algorithm study, Users need to purchase a RPI SBC ( Raspberry PI3 A+/B+, Raspberry PI3 B, Raspberry PI 4B) to pair with DTOF module for use/development.

(a) TOFLIDAR_LD06 (b) Uart cable (c) raspberry pi 4B /PI 3B/PI3 A+/Pi3 B+

FIG 1 TOFLIDAR_LD06 DEVELOPMENT KIT

CHART 1 TOFLIDAR_LD06 DEVELOPMENT KIT DESCRIPTION

| Item | Qty | Description |
|---|---|---|
| TOFLiDAR_LD06 | 1 | Detection product for space detection and obstacle recognition |
| Uart cable | 1 | Use for connection between the DTOF and Raspberry pi 4B for power and data transfer |
| Raspberry pi 4B/3B/3B+/3A+ | 1 | As a computing tool for the TOF lidar data analysis and visualization into to display device |

# 2. INSTALL RASPBIAN OS ON SD CARD

## STEP1: Download a Raspbian OS

To install Raspbian OS on a SD Card you will need to download a Raspbian OS firstly. User may download the Rasbian OS directly from the official website of raspberry foundation, Ldrobot TOF lidar user manual is based on the version of Raspberry Pi OS (32-bit) with desktop and recommended software as highlighted in the figure 2. *https://www.raspberrypi.org/downloads/raspberry-pi-os/*

*or* *https://www.raspberrypi.com/software/operating-systems/*

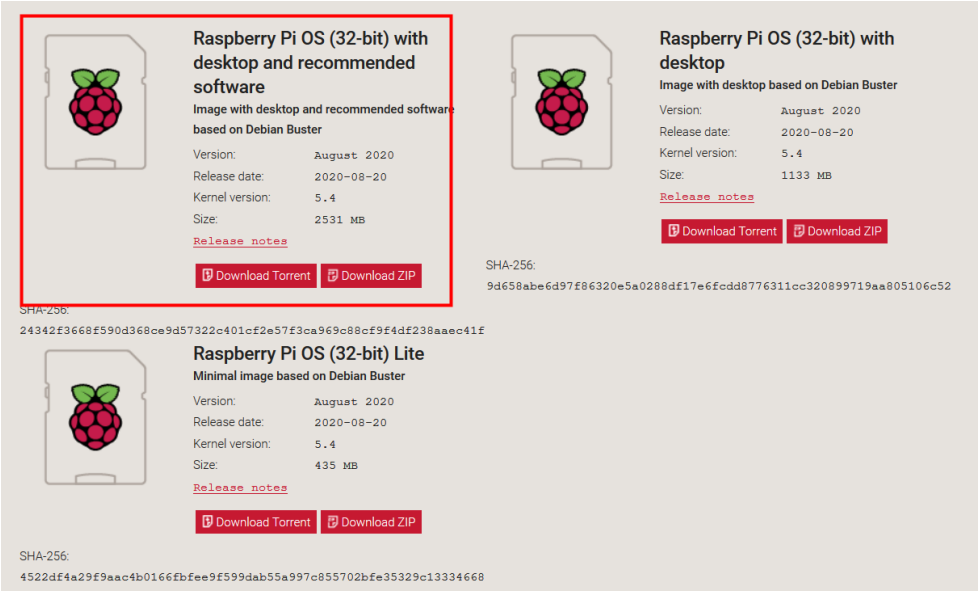



FIG 2. RASPBERRY OFFICIAL WEBSITE

Raspbian os download site of history version：

- Raspbian： *https://downloads.raspberrypi.org/raspbian/images/*
- Raspbian full： *https://downloads.raspberrypi.org/raspbian_full/images/*
- Raspbian lite： *https://downloads.raspberrypi.org/raspbian_lite/images/*

## STEP2: Flash Raspbian OS into SD card

After downloading a Raspbian OS, you need to install win32diskimager as the tool to flash Raspbian OS into SD card. After the image file has been flashed into the SD card sucessfully ，SD card will automatically display a boot partition.

*https://sourceforge.net/projects/win32diskimager/*

# 3. INSTALL ROS MELODIC ON RASPBIAN OS

Powering up the Raspberry Pi. And then insert the Micro SD card into the Pi SD-cage . Connects the Mini-HDMI cable to your display ,connect mouse and keyboard . Plug in the power cable to turn on the Raspberry Pi. Then modify the source file of Rasbian OS.

Installation reference tutorial is

*http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Melodic%20on%20the%20Raspberry%20Pi*

- If you are a Chinese user, please modify as follows:

$ sudo nano /etc/apt/sources.list

Block the existing modification as:

deb http://mirrors.ustc.edu.cn/raspbian/raspbian/ buster main contrib non-free rpi

$ sudo nano /etc/apt/sources.list.d/raspi.list

Block the existing modification as:

deb https://mirrors.tuna.tsinghua.edu.cn/raspberrypi/ buster main ui

- If you are a user outside of Chinese, please modify as follows:

$ sudo nano /etc/apt/sources.list

Block the existing modification as:

Install any source from the following URL: *https://www.raspbian.org/RaspbianMirrors*

Example:

[1]

deb http://mirror.nus.edu.sg/raspbian/raspbian/ buster main contrib non-free rpi

[2]

deb http://ftp.jaist.ac.jp/raspbian/ buster main contrib non-free rpi

[3]

deb http://mirror.ox.ac.uk/sites/archive.raspbian.org/archive/raspbian/ buster main contrib non-free rpi

[4]

deb http://mirrors.ocf.berkeley.edu/raspbian/raspbian/ buster main contrib non-free rpi

[5]

deb http://reflection.oss.ou.edu/raspbian/raspbian/ buster main contrib non-free rpi

[6]

deb http://mirror.liquidtelecom.com/raspbian/raspbian/ buster main contrib non-free rpi

[7]

deb http://mirrordirector.raspbian.org/raspbian/ buster main contrib non-free rpi

[8]

deb https://archive.raspbian.org/raspbian/ buster main contrib non-free rpi

[9]

deb https://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main contrib non-free rpi

[10]

deb http://mirrors.aliyun.com/raspbian/raspbian/ buster main contrib non-free rpi

[11]

deb http://ftp.cse.yzu.edu.tw/Linux/raspbian/raspbian/ buster main contrib non-free rpi

## STEP1: Install Dependencies and Download ROS source packages

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

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

$ sudo apt-get update

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

Then initialize rosdep and update it

```
$ sudo rosdep init
$ rosdep update
```

**STEP2(OPTIONAL): Solve the ERROR:**

ERROR :cannot download default sources list from: https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/sources.list.d/20-default.list Website may be down.

$ cd ~

$ git clone https://github.com/ros/rosdistro.git

if Cloning into 'rosdistro'...

fatal: unable to access 'https://github.com/ros/rosdistro.git/': Failed to connect to github.com port 443: Connection timed out

You can go to this link to download the source package(rosdistro-master.zip) of the master branch. this link is https://github.com/ros/rosdistro

$ mkdir ~/rosdistro

$ unzip rosdistro-master.zip -d ~/rosdistro

$ mv ~/rosdistro/rosdistro-master/* ~/rosdistro/

$ rm -rf ~/rosdistro/rosdistro-master

When the file is successfully cloned or downloaded, proceed as follows:

$ cd ~

***Step1:***

$ sudo nano ~/rosdistro/rosdep/sources.list.d/20-default.list

Modify the file as follows:

Note that ‘pi’ is the user name of the system, you can replace it with your own system user name

# os-specific listings first

#yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml osx

yaml file:///home/pi/rosdistro/rosdep/osx-homebrew.yaml osx

# generic

#yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml

#yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml

#yaml https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml

#gbpdistro https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml fuerte

yaml file:///home/pi/rosdistro/rosdep/base.yaml

yaml file:///home/pi/rosdistro/rosdep/python.yaml

yaml file:///home/pi/rosdistro/rosdep/ruby.yaml

gbpdistro file:///home/pi/rosdistro/releases/fuerte.yaml fuerte

***Step2:***

$ sudo nano /usr/lib/python2.7/dist-packages/rosdep2/sources_list.py

Modify the file as follows:

# default file to download with 'init' command in order to bootstrap

# rosdep

#DEFAULT_SOURCES_LIST_URL = 'https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/sources.list.d/20-default.list'

DEFAULT_SOURCES_LIST_URL = 'file:///home/pi/rosdistro/rosdep/sources.list.d/20-default.list'

# seconds to wait before aborting download of rosdep data

DOWNLOAD_TIMEOUT = 15.0

***Step3:***

$ sudo nano /usr/lib/python2.7/dist-packages/rosdep2/rep3.py

Modify the file as follows:

```
# location of targets file for processing gbpdistro files

#REP3_TARGETS_URL                                                    =
'https://raw.githubusercontent.com/ros/rosdistro/master/releases/targets.yaml'

REP3_TARGETS_URL = 'file:///home/pi/rosdistro/releases/targets.yaml'

# seconds to wait before aborting download of gbpdistro data

DOWNLOAD_TIMEOUT = 15.0
```

***Step4:***

$ sudo nano /usr/lib/python2.7/dist-packages/rosdistro/__init__.py

Modify the file as follows:

```
# same version as in:

# - setup.py

# - stdeb.cfg

__version__ = '0.8.3'


# index information


#DEFAULT_INDEX_URL                                                    =
'https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml'

DEFAULT_INDEX_URL = 'file:///home/pi/rosdistro/index-v4.yaml'
```

Then reinitialize rosdep and update it

```
$ sudo rosdep init

$ rosdep update
```

**STEP3: Install Melodic Desktop**

You need to create a dedicated catkin workspace for building ROS and move to that directory.

```
$ mkdir ~/ros_catkin_ws

$ cd ~/ros_catkin_ws

$ rosinstall_generator desktop --rosdistro melodic --deps --wet-only --tar > melodic-desktop-wet.rosinstall

$ wstool init -j8 src melodic-desktop-wet.rosinstall
```

The command will take a few minutes to download all of the core ROS packages into the src folder. If wstool init fails or is interrupted, you can resume the download by running (network access may affect the download of ROS source code, because if you encounter download failures, please try multiple times):

```
$ wstool update -j 4 -t src
```

**STEP4: Fix the Issues**

Let's install the compatible version of Assimp (Open Asset Import Library) to fix collada_urdf dependency problem.

```
$ mkdir -p ~/ros_catkin_ws/external_src

$ cd ~/ros_catkin_ws/external_src

$ wget http://sourceforge.net/projects/assimp/files/assimp-3.1/assimp-3.1.1_no_test_models.zip/download -O assimp-3.1.1_no_test_models.zip

$ unzip assimp-3.1.1_no_test_models.zip

$ cd assimp-3.1.1

$ cmake .
```

```
$ make
$ sudo make install
```

The user need to  install OGRE for rviz ,too

```
$ sudo apt-get install libogre-1.9-dev
```

The next step is to use the rosdep tool for installing all the rest of the dependencies:

```
$ cd ~/ros_catkin_ws
$ rosdep install -y --from-paths src --ignore-src --rosdistro melodic -r --os=debian:buster
```

**STEP5: Build and Source the Installation**

Once it has completed downloading the packages and resolving the dependencies you are ready to build the catkin packages. If you're using raspberry Pi 3, you need to increase the swap size first. If you are using raspberry Pi 4,you can skip this step.

```
$ sudo mkdir /swap
$ cd /swap
$ sudo dd if=/dev/zero of=swapfile bs=1024 count=2000000
$ sudo mkswap swapfile
$ sudo swapon swapfile
```

Then build.

```
$ cd ~/ros_catkin_ws
$ sudo ./src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/melodic -j2
```

Now ROS Melodic should be installed on your Raspberry Pi 4. We will source

the new installation with following command:

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

Try launching roscore to check if everything was successful.

```
$ roscore
```

## 4. START SERIAL ttyS0

```
$ sudo raspi-config
```

Open the system configuration interface as shown in the figure below, and select the **interface options**.

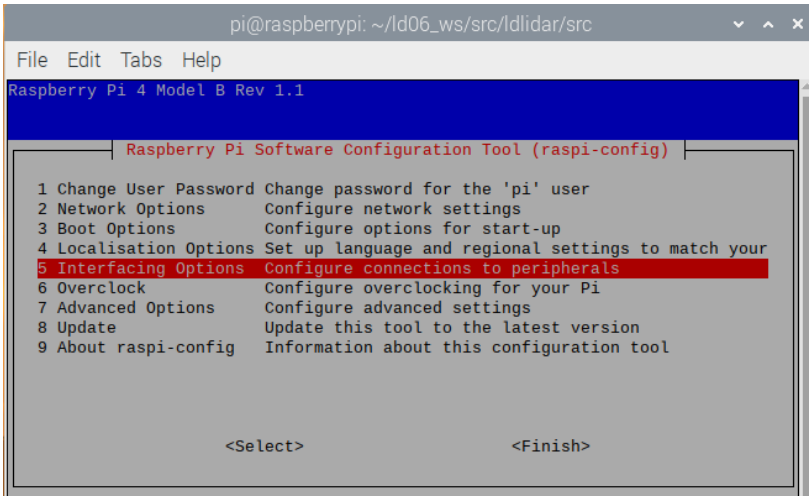



FIG 3. RASPBERRY SYSTEM CONFIGURATION I

Then select **P6 serial**.

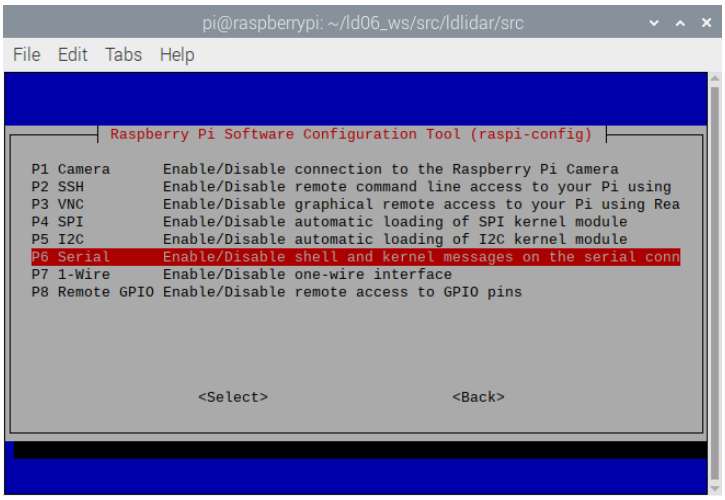



FIG 4. RASPBERRY SYSTEM CONFIGURATION II

Then click **Yes**.

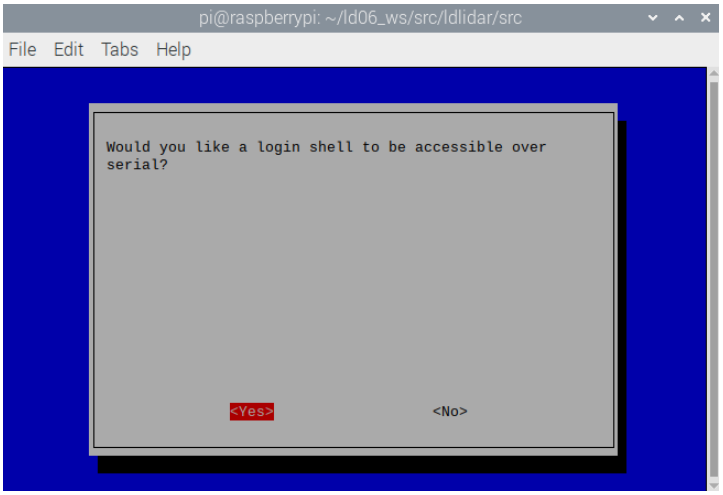



FIG 5. RASPBERRY SYSTEM CONFIGURATION III

Then Save and exit. Reboot raspberryPi. Please do not disconnect power during reboot. Otherwise, the serial configuration may not take effect.

```
$ reboot
```

Check if the serial port is open. View serial port mapping relationship

```
$ ls -l /dev
```

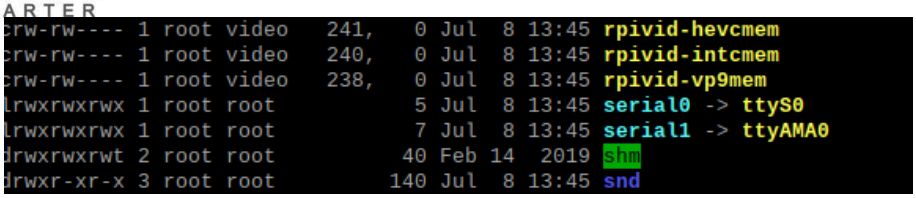



FIG 6. RASPBERRY SERIAL PORT MAPPING

Serial0 is the serial port corresponding to the GPIO pin. If you see serial0 connected to ttys0, the serial port configuration is successful.

## 5. INSTALL LIDAR ROS PACKAGE

**STEP1: Device connection**

Connect Lidar and Raspberry Pi 4B as shown in the figure below. 5v connect 5v Power, GND connect Ground, Motor PWM connect BCM18(pwm0),Lidar Uart TX connect BCM15(RXD).

Our company's Lidar support internal speed control. If you want external speed control, you need to download an install the WiringPi library yourself, and configure the BCM18 pin as PWM0, output a 24KHz PWM signal, and implement PID speed control yourself.

WiringPi library installation method:

Then install wiringPi,WiringPi has updated to 2.52 for the Raspberry Pi 4B.

```
$ cd /tmp
$ wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
```

After wiringPi being updated ,you can check with the latest version update

```
gpio -v
```

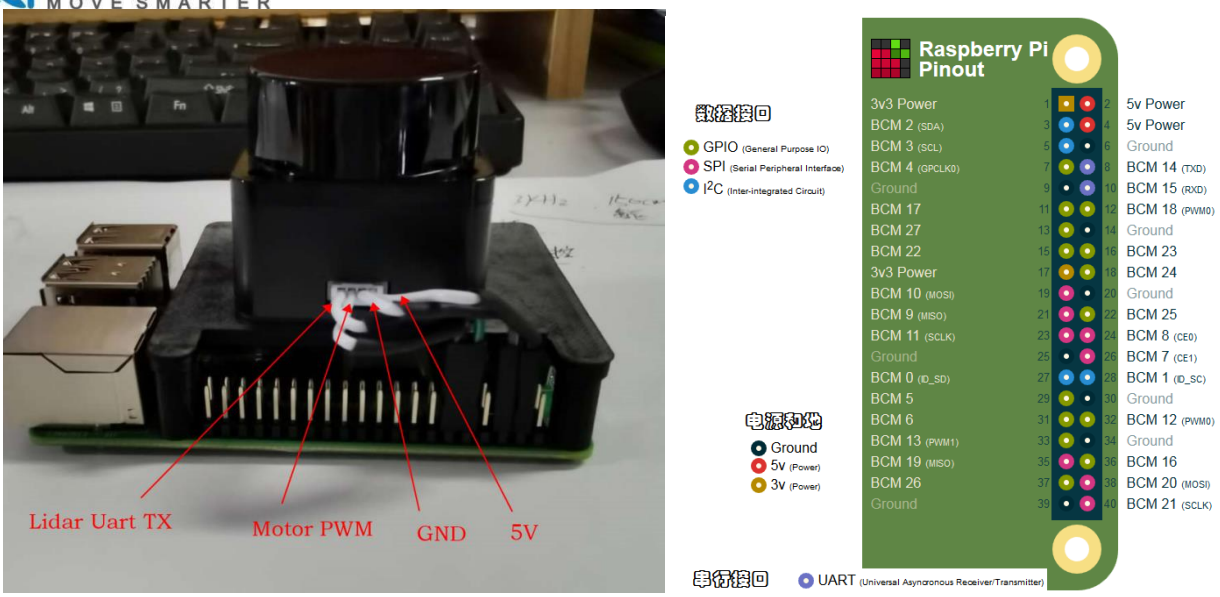



FIG 7. CONNECTION BETWEEN LIDAR AND RASPBERRY PI 4B

The users need to have the -x permission of raspbian kernel system. After connecting TOF LiDAR_LD06 with raspberry pi 4B/3B/3B+/3A+.

```
$ sudo chmod 777 /dev/ttyS0
```

**STEP2: ROS DTOF_LD06 Driver Compile**

The ROS driver package has been uploaded into the sdk_ld06_raspberry_ros/ directory. You just open the sdk_ld06_raspberry_ros workspace and compile.

```
$ cd ~

$ git clone https://github.com/ldrobotSensorTeam/sdk_ld06_raspberry_ros.git

if  Cloning into '' sdk_ld06_raspberry_ros...

fatal: unable to access: Failed to connect to github.com port 443: Connection timed out. You can go to this link to download the source package of the main branch. this link is https://github.com/ldrobotSensorTeam/sdk_ld06_raspberry_ros

The download name is sdk_ld06_raspberry_ros-main.zip of source package is success, then:

$ mkdir ~/sdk_ld06_raspberry_ros

$ unzip sdk_ld06_raspbery_ros-main.zip -d ~/sdk_ld06_raspberry_ros
```

```
$ cp ~/sdk_ld06_raspberry_ros/sdk_ld06_raspbery_ros-main/*  ~/sdk_ld06_raspberry_ros/ -a

$ rm -rf  ~/sdk_ld06_raspberry_ros/sdk_ld06_raspbery_ros-main/
```

When the file is successfully cloned or downloaded, proceed as follows:

```
$ cd  ~/sdk_ld06_raspberry_ros

$ catkin_make

$ source devel/setup.bash

$ roslaunch ldlidar ld06.launch
```

**STEP3: RVIZ results**

After running the launch file, you need to open a new terminal. Then run rviz to view the scan results, as shown in the following figure:

```
$ rosrun rviz rviz
```

You need to click the open config button. Then select the ldlidar.rviz file.

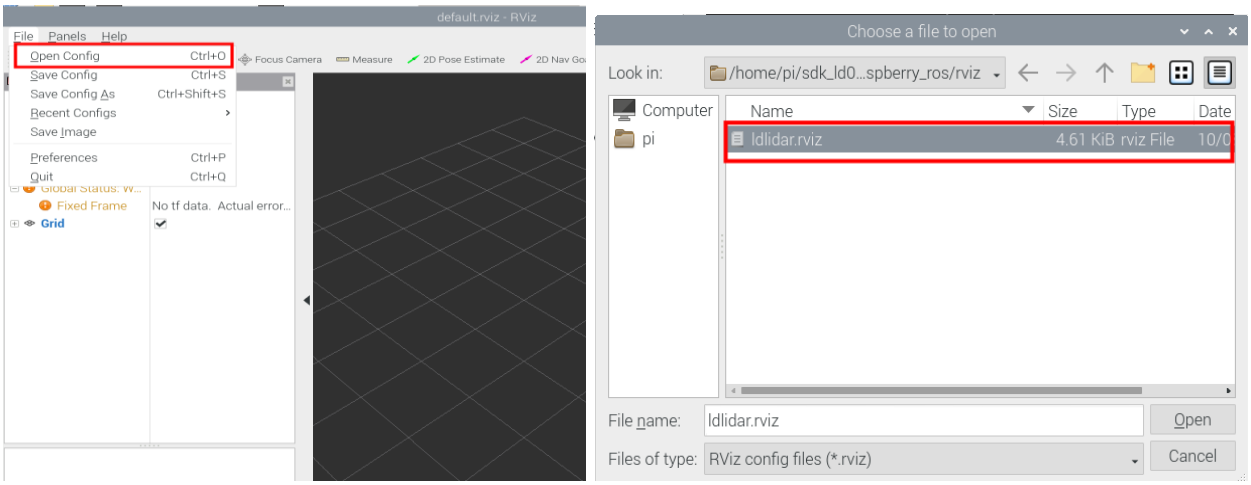



FIG 8 RVIZ CONFIGURATION

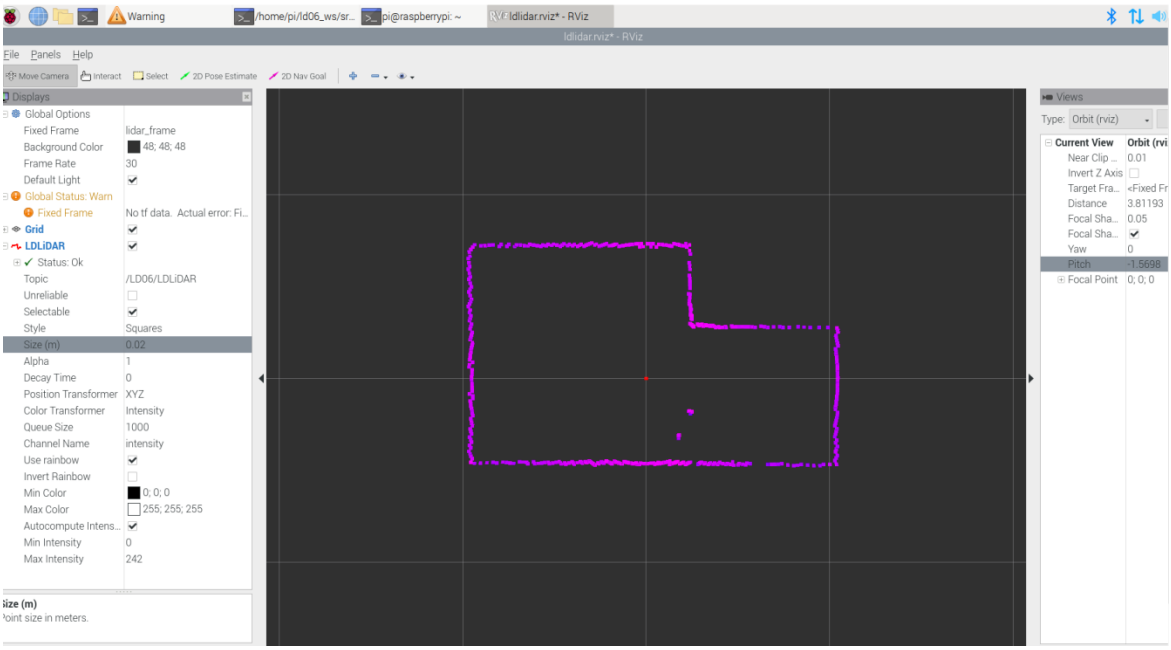



FIG 9 TOFLiDAR_LD06 RVIZ

## 6. USE CAUTION

***ALARM: Please connect the TOFLiDAR_LD06 before you power up the raspberry.***

- **Temperature**

When the working environment temperature of TOFLiDAR_LD06 is too high or too low, it will affect the accuracy of the distance measuring system. It may also damage the structure of the scanning system and reduce the life of the TOFLiDAR_LD06. Avoid use in high temperature (>40 degrees Celsius) and low temperature (<0 degrees Celsius) conditions.

- **Ambient lighting**

The ideal working environment for the Lidar is indoor, indoor lighting (including no light) will not affect it work. Don’t using a strong light source (such as a high-power laser) to directly illuminate the lidar's vision system.

If you need to use it outdoors, please avoid that the its vision system is directly facing the sun. This may cause permanent damage to the vision system's sensor chip,

thus invalidating the distance measurement.

Please note that the Lidar standard version is subject to interference in outdoor strong sunlight reflection environments.

- **Power demand**

  For development ,both external adaptor or independent power bank works , but need to ensue 5V and 200MA current power input, for external adaptor solution, the Raspberry Pi SBC adaptor is the preference choose.