

YDLIDAR X2 USER MANUAL



CONTENTS

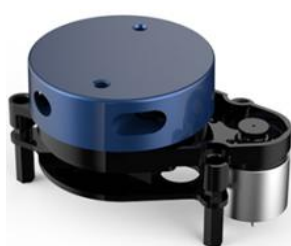
1	YDLIDAR X2 LIDAR DEVELOPMENT KIT	1
1.1	Development Kit	1
2	USAGE UNDER WINDOWS	1
2.1	Device Connection	1
2.2	Driver Installation	3
2.3	Evaluation Software Usage	4
2.3.1	Start Scanning	5
2.3.2	Data Storage	6
2.3.3	Display Mean and Standard Deviation	6
2.3.4	Play and Record	6
2.3.5	Debug	7
2.3.6	Filter	8
3	LINUX ROS OPERATION	8
3.1	Device Connection	8
3.2	Compile and Install YDlidar-SDK	8
3.3	ROS Driver Installation	9
3.4	Run the ydlidar_ros_driver	9
3.5	RVIZ View Scan Results	10
3.6	Modify Scan Angle	10
4	CAUTION	12
4.1	Ambient Temperature	12
4.2	Ambient Lighting	12
4.3	Power Supply	12
5	REVISE	13

1 YDLIDAR X2 LIDAR DEVELOPMENT KIT

The development kit of YDLIDAR X2 lidar (hereinafter referred to as X2) is an accessory tool provided for performance evaluation and early development of the X2. Through the X2 development kit, and with the evaluation software, users can observe point cloud data scanned by X2 on your environment or development on the SDK.

1.1 Development Kit

The X2 development kit has the following components:



X2 Lidar



USB Type-C Cable



USB Adapter Board

FIG 1 YDLIDAR X2 DEVELOPMENT KIT

CHART 1 YDLIDAR X2 LIDAR DEVELOPMENT KIT DESCRIPTION

Item	Qty.	Description
X2 Lidar	1	Standard version of the X2 lidar. The X2 has an integrated motor drive for despin control and motor control
USB Type-C Cable	1	Use with USB adapter board to connect X2 and PC. USB cable is both a power supply cable and a data cable
USB Adapter Board	1	Realize the function of USB to UART, which is convenient for the fast interconnection between X2 and PC. Serial port DTR signal to control X2 motor despin control, a Micro USB power interface (PWR) for auxiliary power supply is provided

Note: USB Adapter board has two USB TYPE C interface: USB_DATA, USB_PWR.

USB_DATA: Data powered interface. In most cases, this interface can be used to meet power and communication requirements.

USB_PWR: Auxiliary power supply interface. The USB interface of some development platforms has weak current drive capability. At this time, auxiliary power supply can be used.

2 USAGE UNDER WINDOWS

2.1 Device Connection

When evaluating and developing X2 under windows, X2 and PC needed to be connected. The specific process is as follows:

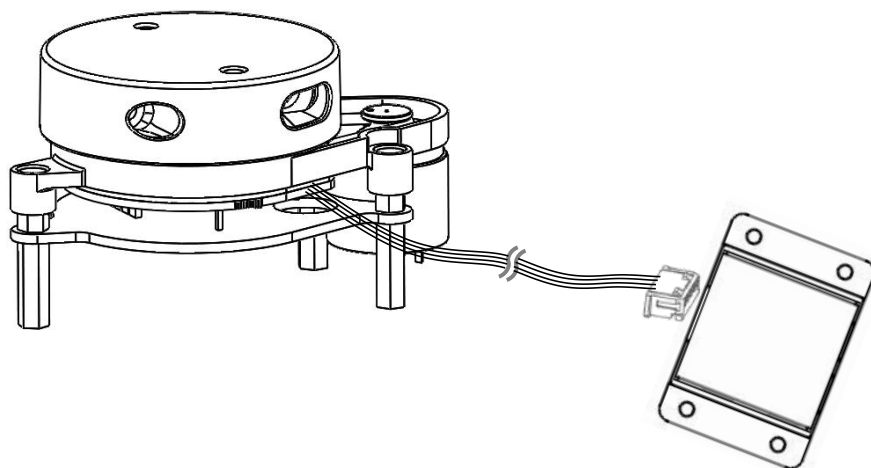


FIG 2 YDLIDAR X2 DEVICE CONNECTION STEP 1

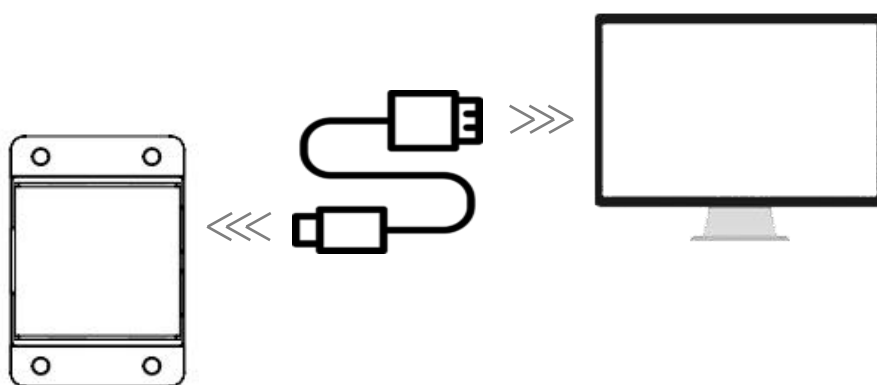


FIG 3 YDLIDAR X2 DEVICE CONNECTION STEP 2

Connect the adapter board with X2 first, then connect the USB cable to the USB port of the adapter board and the PC. Note that the Type-C interface of the USB cable is connected to the USB_DATA of the USB adapter board, and the X2 enters the idle mode after being powered on. The motor does not rotate.

The drive current of USB interface of some development platforms or PC is not sufficient. X2 need to be connected to the auxiliary power supply of +5V, otherwise the lidar will be abnormal.

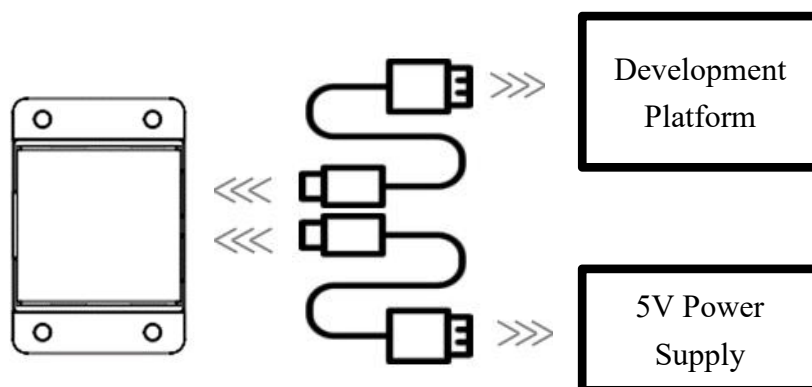


FIG 4 YDLIDAR X2 AUXILIARY POWER SUPPLY

2.2 Driver Installation

To evaluate and develop the X2 under Windows, the serial port driver of the USB adapter board. The USB adapter board need to be installed of this kit adopts CP2102 chip to realize serial port (UART) to USB signal conversion. Its driver can be downloaded from our official website:

<https://www.ydlidar.com/download/category/lidar-sensor>

After unzip the driver package, run the CP2102's Windows driver installation file (exe file under CP210x_VCP_Windows). Please select the 32-bit version (x86) or 64-bit version (x64) installation program according to the version of the windows operating system.

x64	2013/10/25 11:39	文件夹	
x86	2013/10/25 11:39	文件夹	
CP210xVCPInstaller_x64.exe	2013/10/25 11:39	应用程序	1,026 KB
CP210xVCPInstaller_x86.exe	2013/10/25 11:39	应用程序	901 KB
dpinst.xml	2013/10/25 11:39	XML 文档	12 KB
ReleaseNotes.txt	2013/10/25 11:39	文本文档	10 KB
SLAB_License_Agreement_VCP_Windo...	2013/10/25 11:39	文本文档	9 KB
slabvcp.cat	2013/10/25 11:39	安全目录	12 KB
slabvcp.inf	2013/10/25 11:39	安装信息	5 KB

FIG 5 YDLIDAR X2 DRIVER VERSION SELECTION

Double-click the exe file and follow the prompts to install it.



FIG 6 YDLIDAR X2 DRIVER INSTALLING

After the installation is complete, right-click on [My Computer] and select [Properties]. On the open [System] screen, select [Device Manager] from the left menu to access the [Device Manager].

Expand [Port] to see the serial port name corresponding to the identified USB adapter, that is, the driver installation is successful. The following figure shows COM3. (Note that the port must be checked in case of X2 and PC interconnection).

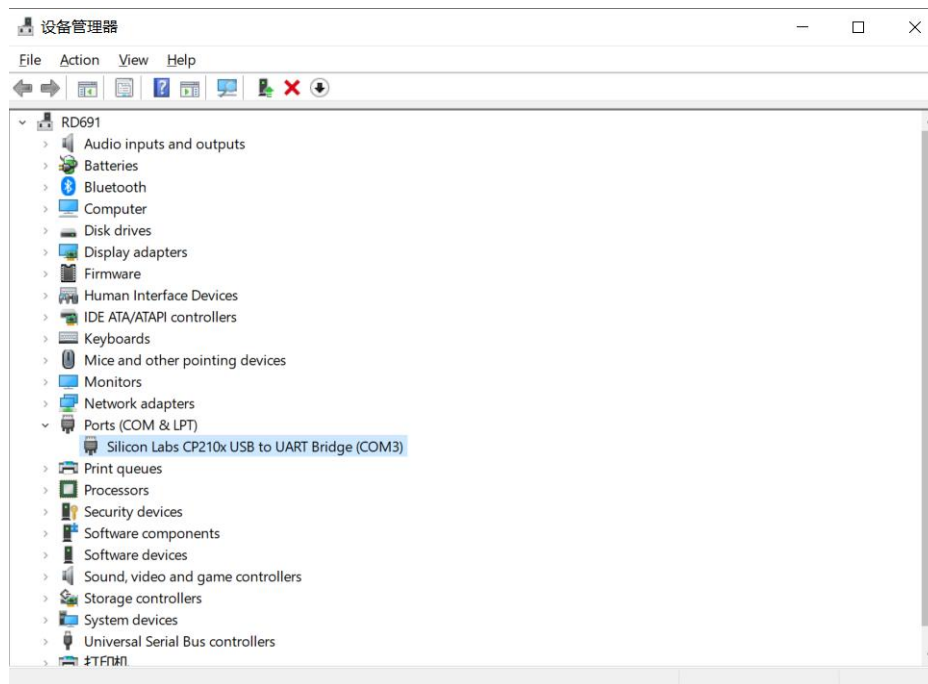


FIG 7 YDLIDAR X2 DRIVER INSTALLATION CHECK

2.3 Evaluation Software Usage

YDLIDAR provides Point Cloud Viewer, a point cloud data LidarClient software EaiLidarTest for X2 real-time scanning. Users can intuitively observe the X2 scanning effect chart. GDL real-time point cloud data and real-time scanning frequency are provided on YDLIDAR. At the same time, save the scanned data offline to an external file for further analysis. LidarClient software download link:

<https://www.ydlidar.com/download/category/lidar-sensor>

Before using the YDLIDAR software, make sure that the X2 USB adapter board serial port driver is installed successfully, and interconnect the X2 with the USB port of the PC. Run the evaluation software: EaiLidarTest.exe, select the corresponding serial port number and model number. Meanwhile, users could choose language on the top right corner.

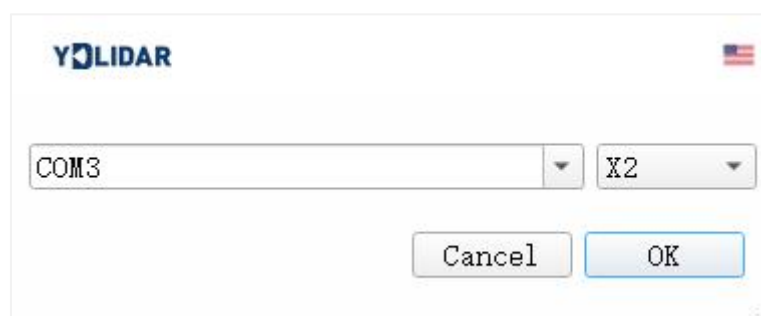


FIG 8 YDLIDAR X2 EVALUATION SOFTWARE

If the connection is correct, you will see the following screen:

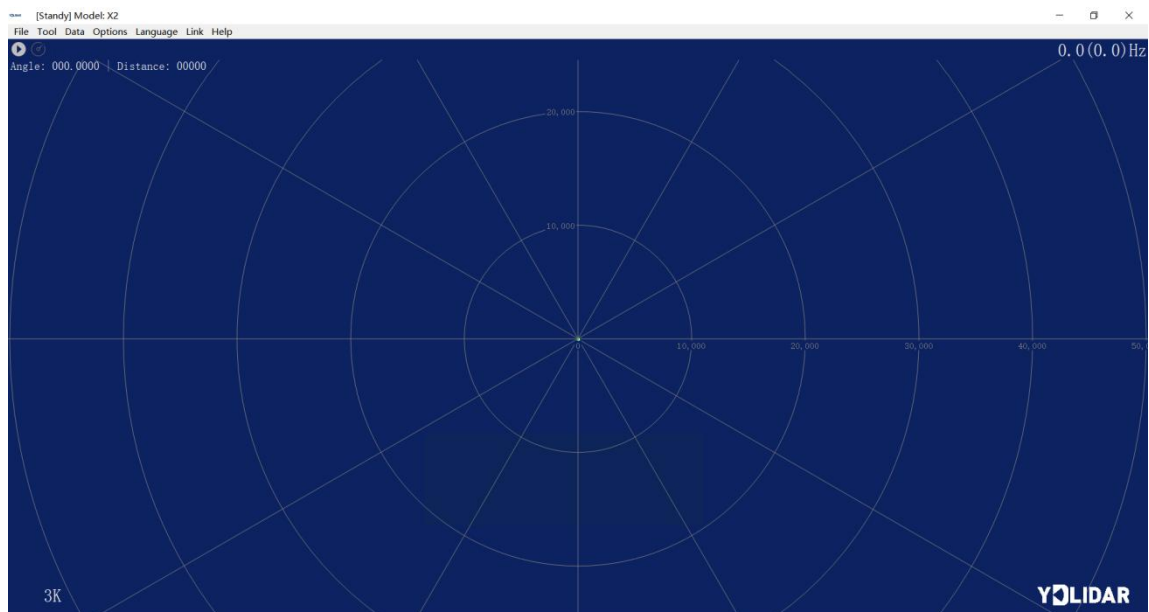




FIG 9 POINTCLOUD VIEWER INTERFACE

2.3.1 Start Scanning

Click  to start scanning and display the environment point cloud, the upper left corner displays the angle & distance information of the red line position (unit: mm). Click  to stop it. as shown below:

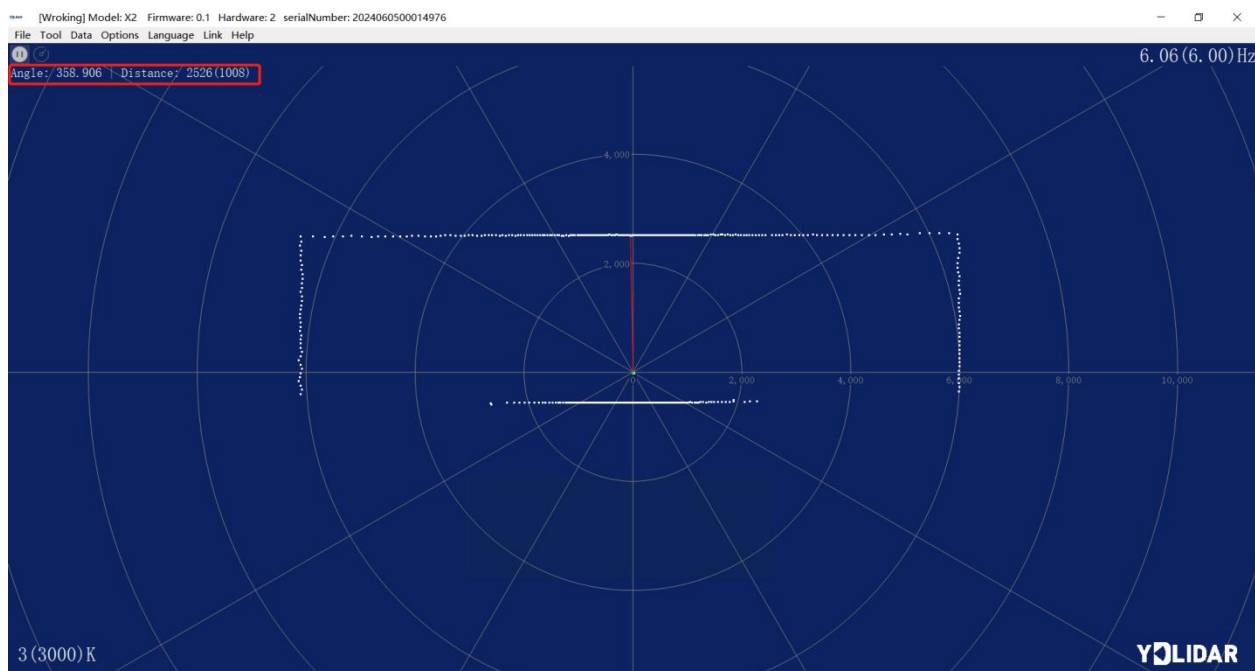


FIG 10 LIDAR SCANNING POINT CLOUD DISPLAY

2.3.2 Data Storage

During lidar scanning, click [File] in the main menu, select [Export to Excel], and save point cloud data according to the prompts. Then the system will save the point cloud information scanned in a circle in Excel format.



FIG 11 SAVE DATA

2.3.3 Display Mean and Standard Deviation

Click [Tools] in the main menu, then select [Mean And STD] - [View].



FIG 12 YDLIDAR G4 DISPLAY MEAN AND STANDARD DEVIATION

Choose one according to needs, move the mouse to the test position, right-click the pop-up menu, and select [Lock Mouse Tracking].

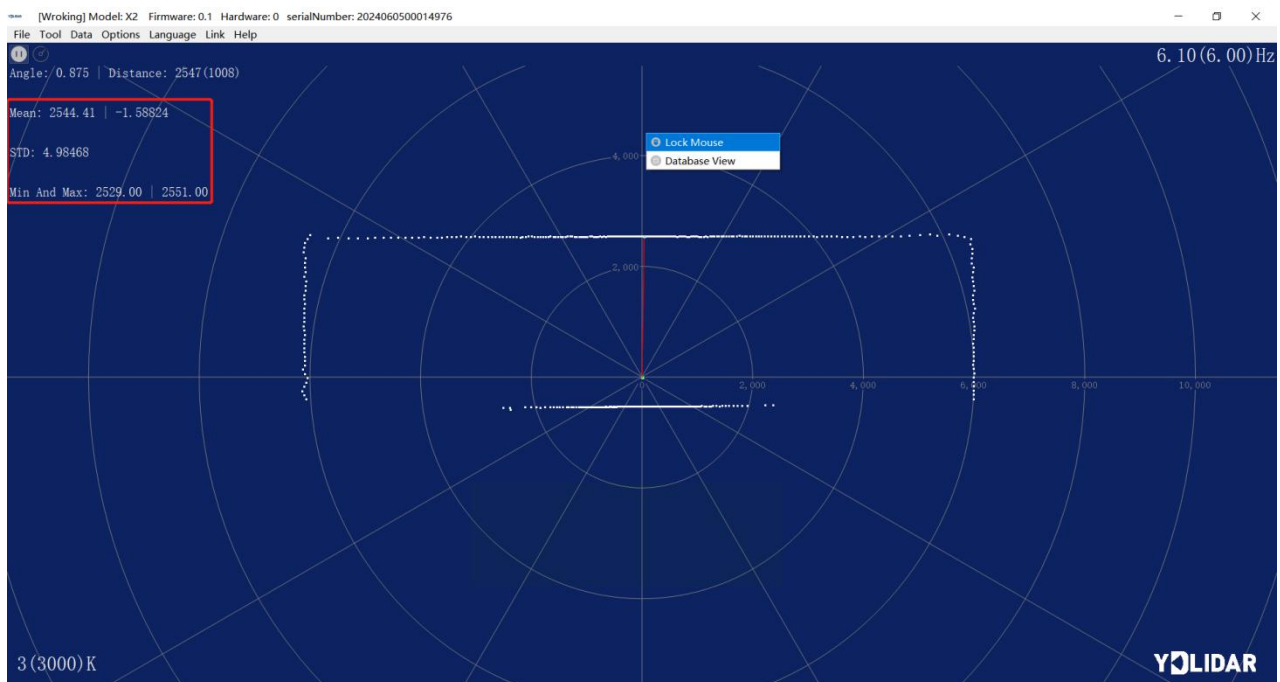


FIG 13 LOCK MOUSE TRACKING

2.3.4 Play and Record

Click [Tools] in the main menu, then select [Record and Play].

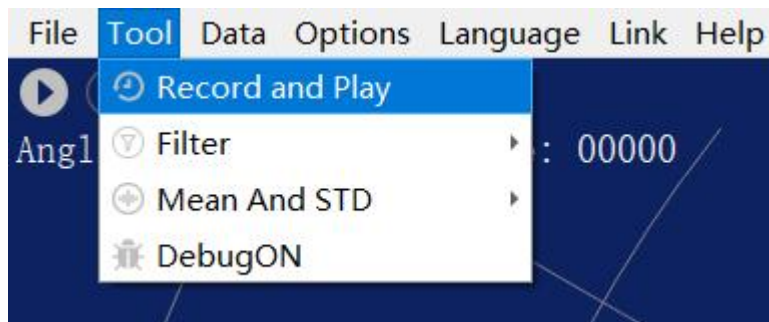



FIG 14 RECORD AND PLAY

The main window is displayed  as follows:

To record lidar data, click  to start recording, and click  to stop recording.

In non-scanning mode, click  to start play.

The play process is as follows:

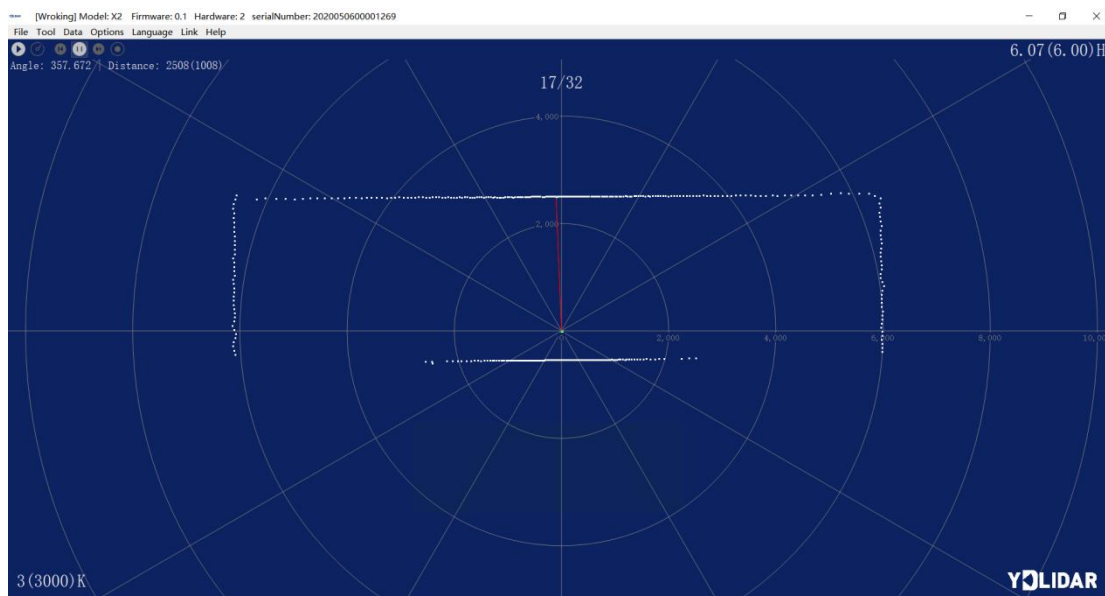


FIG 15 PLAY PROCESS

2.3.5 Debug

Click [Tools] in the main menu, and then select [DebugON] to output the raw lidar data to the "viewer_log.txt" and "viewer_log_err.txt" files.

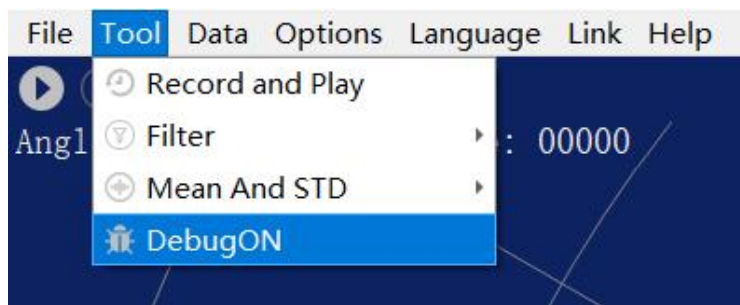


FIG 16 START DEBUGGING

2.3.6 Filter

Click [Tools] in the main menu, and then select [Filter] to add lidar data filtering algorithm.

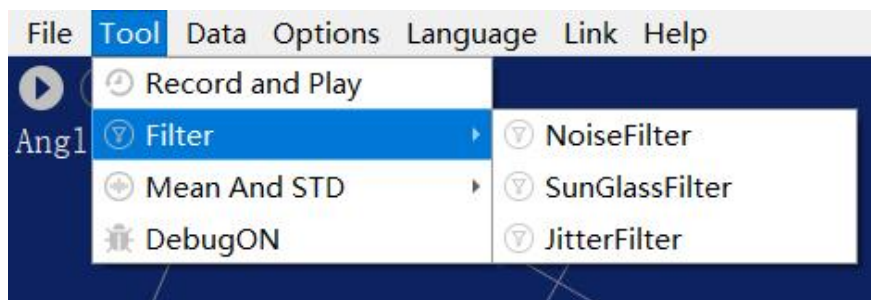


FIG 17 FILTER SETTING

Note: For more functions of EaiLidarTest, please click [Help], select [More Information], and learn more about how to use it.

3 LINUX ROS OPERATION

There are many Linux versions, this article only uses Ubuntu 18.04, Melodic version ROS as an example.

SDK driver address:

<https://github.com/YDLIDAR/YDLidar-SDK>

ROS driver address:

https://github.com/YDLIDAR/ydlidar_ros_driver

3.1 Device Connection

Under Linux, the X2 and PC interconnect processes are consistent with those under Windows. See [Device Connection under Window](#).

3.2 Compile and Install YDLidar-SDK

YDLIDAR_ROS_DRIVER depends on the YDLIDAR-SDK library. If never installed the YDLidar-SDK library before, or it has expired, it is necessary to install the YDLidar-SDK library. If you have the latest version of ydlidar-SDK, please skip this step, then go to the next step.

```
$ git clone https://github.com/YDLIDAR/YDLidar-SDK.git
$ cd YDLidar-SDK/build
$ cmake ..
$ make
$ sudo make install
```

3.3 ROS Driver Installation

- 1) Cloning GitHub's YDLidar ROS Driver Package:

```
$ git clone https://github.com/YDLIDAR/ydlidar_ros_driver.git
ydlidar_ws/src/ydlidar_ros_driver
```

- 2) Build the Ydlidar_ROS_Driver software package:

```
$ cd ydlidar_ws
$ catkin_make
```

- 3) Package environment Settings:

```
$ source ./devel/setup.sh
```

Note: Add permanent workspace environment variables. It will be very convenient if ROS environment variables are automatically added to your bash session every time when start a new shell:

```
$ echo "source ~/ydlidar_ws/devel/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

- 4) Verify that the package path is set, echo the ROS_PACKAGE_PATH variable.

```
$ echo $ROS_PACKAGE_PATH
```

It can see something like this: /home/tony/ydlidar_ws/src:/opt/ros/melodic/share Create

- 5) Serial Port Alias [Optional]

```
$ chmod 0777 src/ydlidar_ros_driver/startup/*
$ sudo sh src/ydlidar_ros_driver/startup/initenv.sh
```

Note: After completing the previous operation, re-insert the LIDAR again.

3.4 Run the ydlidar_ros_driver

Run ydlidar_ros_driver with startup file, as shown below:

```
$ roslaunch ydlidar_ros_driver X2.launch
```

3.5 RVIZ View Scan Results

Run the launch file and open rviz to view the X2 scan results, as shown in the figure below:

```
$ roslaunch ydlidar_ros_driver lidar_view.launch
```

Note: Take G4 as an example by default, if use other types of lidar, need to change lidar.launch in lidar_view.launch file to the corresponding *.launch file. (If X2 lidar is used, it needs to be changed to X2.launch)

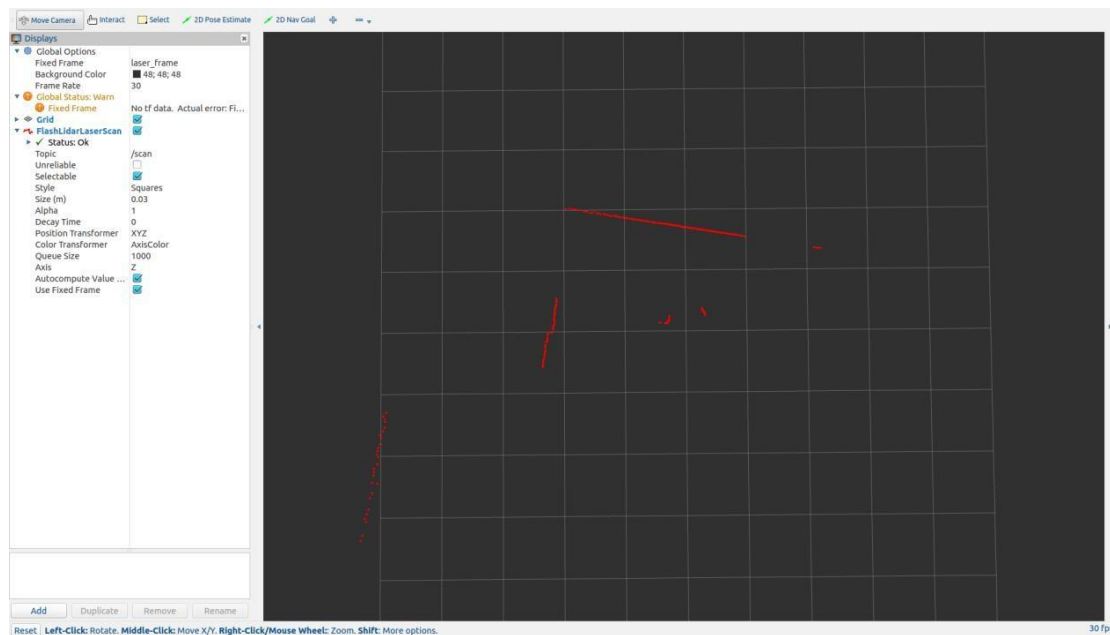


FIG 18 YDLIDAR X2 RVIZ

3.6 Modify Scan Angle

The scanning data seen by running the launch file is displayed by default with 360-degree data. To modify the display range, you need to modify the configuration parameters in the launch file. The specific operation is as follows:

- 1) Switch to the directory where the corresponding [launch file] is located, edit the file, and its content is as shown in the figure:

```
$ vim X2.launch
```

```
<launch>
<node name="ydlidar_lidar_publisher" pkg="ydlidar_ros_driver" type="ydlidar_ros_driver_node" output="screen" respawn="false" >
  <!-- string property -->
  <param name="port" type="string" value="/dev/ydlidar"/>
  <param name="frame_id" type="string" value="laser_frame"/>
  <param name="ignore_array" type="string" value="" />

  <!-- int property -->
  <param name="baudrate" type="int" value="512000"/>
  <!-- 0:TYPE_TOF, 1:TYPE_TRIANGLE, 2:TYPE_TOF_NET -->
  <param name="lidar_type" type="int" value="0"/>
  <!-- 0:YDLIDAR_TYPE_SERIAL, 1:YDLIDAR_TYPE_TCP -->
  <param name="device_type" type="int" value="0"/>
  <param name="sample_rate" type="int" value="20"/>
  <param name="abnormal_check_count" type="int" value="4"/>

  <!-- bool property -->
  <param name="resolution_fixed" type="bool" value="true"/>
  <param name="auto_reconnect" type="bool" value="true"/>
  <param name="reversion" type="bool" value="true"/>
  <param name="inverted" type="bool" value="true"/>
  <param name="isSingleChannel" type="bool" value="false"/>
  <param name="intensity" type="bool" value="false"/>
  <param name="support_motor_dtr" type="bool" value="false"/>
  <param name="invalid_range_is_inf" type="bool" value="false"/>

  <!-- float property -->
  <param name="angle_min" type="double" value="-180" />
  <param name="angle_max" type="double" value="180" />
  <param name="range_min" type="double" value="0.01" />
  <param name="range_max" type="double" value="50.0" />
  <param name="frequency" type="double" value="10.0" />
</node>
<node pkg="tf" type="static_transform_publisher" name="base_link_to_laser4"
  args="0.0 0.0 0.2 0.0 0.0 0.0 /base_footprint /laser_frame 40" />
</launch>
```

FIG 19 X2.LAUNCH FILE

Note: For more information about the file contents, please refer to

https://github.com/YDLIDAR/ydlidar_ros_driver#configure-ydlidar_ros_driver-internal-parameter

- 2) The X2 lidar coordinates follow the right-hand rule within ROS, with an angle range of $[-180, 180]$. "angle_min" is the start angle, and "angle_max" is the endangle. The specific scope needs to be modified according to actual use.

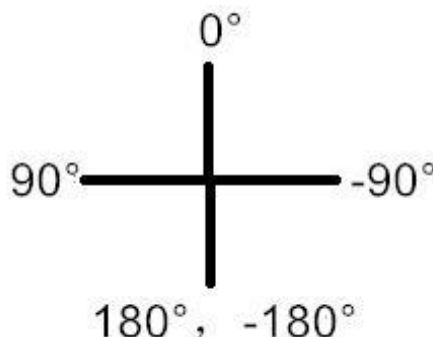


FIG 20 YDLIDAR X2 COORDINATES DEFINITION

4 CAUTION

4.1 Ambient Temperature

When the working environment temperature of X2 is overhigh or overlow, it will affect the accuracy of the distance measuring system. It may also damage the structure of the scanning system and reduce the service life of the X2 lidar. Avoid using in high temperature (>50 degrees celsius) and low temperature (<0 degrees celsius) conditions.

4.2 Ambient Lighting

The ideal working environment of X2 is indoor, and the indoor environment light (including no light) will not affect the work of X2. However, please avoid using strong light sources (such as high-power lasers) to directly illuminate the X2 vision system.

If you need to use it outdoors, please avoid the X2's vision system directly facing the sun, which may cause permanent damage to the vision system's photosensitive chip, which will invalidate the range finding.

The X2 standard version will cause interference in the distance measurement under the conditions of strong sunlight reflection outdoors, please pay attention to it.

4.3 Power Supply

During the development process, since the drive current of the USB interface of each platform or the USB interface of the computer may be too low to drive the X2, the external power supply of the +5V to the X2 needs to be provided through the USB_PWR interface of the USB interface board. It is not recommended for use mobile phone power bank because the voltage ripple of power bank is too large.

5 REVISE

Date	Version	Content
2017-12-05	1.0	Composing a first draft
2018-01-22	1.1	Added auxiliary power connection method, document description, configuration description, and power supply requirements
2018-04-03	1.2	Adapt to PointCloudViewer2.0 client
2021-08-02	1.3	Adapt ClientViewer, update SDK, ROS tutorial
2025-06-21	1.4	Revised LidarClient&UART driver address