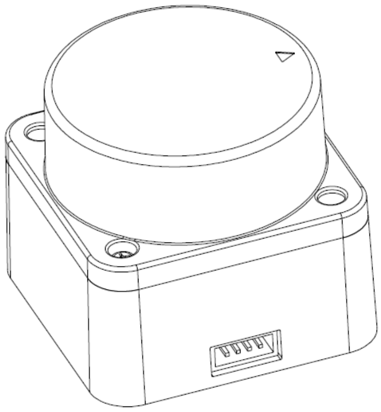
LD06 LiDAR

Principle of DTOF

Ultimate small size , low cost

High reliability , long working life



Development manual v1.0

Contents

sales@ldrobot.com

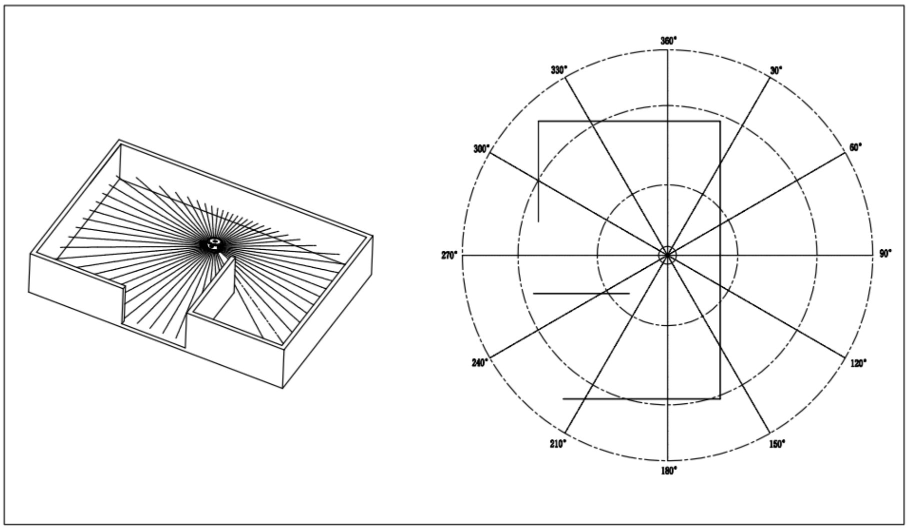
1. Product Description

LD06 is mainly composed of laser ranging core, wireless transmission unit, wireless communication unit, angle measuring unit, motor driving unit and mechanical housing.

The LD06 ranging core adopts DTOF technology to measure 4500 times each second. When it works, LD06 emits the infrared laser forward, the laser is reflected to the single photon receiving unit after encountering the target object. Thus, we get both time of laser emitting and receiving, the gap between them is time of flight. With the light speed, we can calculate the distance.

After receiving distance data, LD06 will combine them with angel value getting from angle measurement unit to comprise the points cloud data, then transmitting the points cloud data to external interface via wireless communication. Meanwhile the external interface provides PWM to allow motor driving unit to drive the motor. After the external control unit gets the rotational speed , it will reach to specified speed through PID algorithm closed-loop control to ensure LD06 work stably.

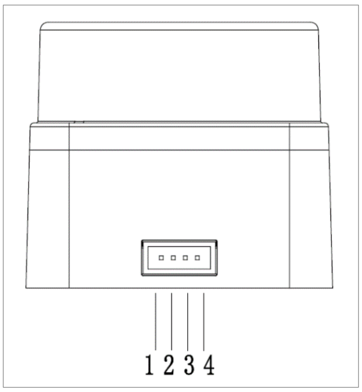
Below please find the environment scanning diagram formed by LD06 point cloud data:



sales@ldrobot.com

2. Communication Interface

LD06 uses ZH1.5T-4P 1.5mm connectors to connect with external system to implement power supply and data receiving. Below please find the definition and parameter requirements for specific interfaces:



Number	Signal Name	Type	Description	Min	Typical	Max
1	Tx	Output	Radar data output	0V	3.3V	3.5 V
2	PWM	Input	Motor control signal	0V	-	3.3 V
3	GND	Power Supply	Power negative	-	0V	-
4	P5V	Power Supply	Power positive	4.5V	5V	5.5 V

LD06 is equipped with a stepless speed adjusting mode motor driver which can control the start, stop and speed of the motor via the PWM signal in the interface. Due to the individual differences of each motor, the actual speed may be different when the duty rate is set as typical value. To accurately control the motor speed, it is needed to according to the speed information in the receiving data to control in closed-loop .

The LD06 takes use of standard asynchronous serial port (UART) to transmit data in one way, the transmission parameters are shown as below table:

Baud Rate	Data Bits	Stop Bits	Parity Check Bit	Flow Control
230400	8 Bits	1	No	No

3. Communication Protocol

3.1. Data Packet Format

LD06 adopts one-way communication, it begins to send measuring data packet once

sales@ldrobot.com

working stably, without any instruction. The format of the data packet is as below:

Start Character	Data Length	Radar Speed		Start Angle		Data	End Angle		Timestamp		CRC check
54H	1 Byte	LSB	MSB	LSB	MSB	……	LSB	MSB	LSB	MSB	1 Byte

- starting character: Length 1 Byte, fixed value 0x54, means the beginning of data packet;
- Data Length: Length 1 Byte, the first three digits reserved, the last five digits represent the number of measured points in a packet, currently fixed value 12;
- Radar speed: Length 2 Byte, in degrees per second;
- Start angle: Length: 2 Byte; unit: 0.01 degree;
- Data: A measurement data length is 3 bytes, please refer to next section for detailed explanation;
- End Angle: Length: 2 Byte; unit: 0.01 degree;
- Timestamp: Length 2 Bytes in ms, recount if reaching to MAX 30000;
- CRC check: Checksum of all previous data;

Please refer to the data structure as following:

```
#define ANGLE_PER_FRAME 12
#define HEADER 0x54
typedef struct  __attribute__((packed))
{
    uint8_t           header;
    uint8_t           ver_len;
    uint16_t          speed;
    uint16_t          start_angle;
    LidarPointStructDef point[POINT_PER_PACK];
    uint16_t          end_angle;
    uint16_t          timestamp;
    uint8_t           crc8;
}LiDARFrameTypeDef;
```


sales@ldrobot.com

The calculation method of CRC check is as following:

```
static const uint8_t CrcTable[256] =
{
    0x00, 0x4d, 0x9a, 0xd7, 0x79, 0x34, 0xe3,
    0xae, 0xf2, 0xbf, 0x68, 0x25, 0x8b, 0xc6, 0x11, 0x5c, 0xa9, 0xe4, 0x33,
    0x7e, 0xd0, 0x9d, 0x4a, 0x07, 0x5b, 0x16, 0xc1, 0x8c, 0x22, 0x6f, 0xb8,
    0xf5, 0x1f, 0x52, 0x85, 0xc8, 0x66, 0x2b, 0xfc, 0xb1, 0xed, 0xa0, 0x77,
    0x3a, 0x94, 0xd9, 0x0e, 0x43, 0xb6, 0xfb, 0x2c, 0x61, 0xcf, 0x82, 0x55,
    0x18, 0x44, 0x09, 0xde, 0x93, 0x3d, 0x70, 0xa7, 0xea, 0x3e, 0x73, 0xa4,
    0xe9, 0x47, 0x0a, 0xdd, 0x90, 0xcc, 0x81, 0x56, 0x1b, 0xb5, 0xf8, 0x2f,
    0x62, 0x97, 0xda, 0x0d, 0x40, 0xee, 0xa3, 0x74, 0x39, 0x65, 0x28, 0xff,
    0xb2, 0x1c, 0x51, 0x86, 0xcb, 0x21, 0x6c, 0xbb, 0xf6, 0x58, 0x15, 0xc2,
    0x8f, 0xd3, 0x9e, 0x49, 0x04, 0xaa, 0xe7, 0x30, 0x7d, 0x88, 0xc5, 0x12,
    0x5f, 0xf1, 0xbc, 0x6b, 0x26, 0x7a, 0x37, 0xe0, 0xad, 0x03, 0x4e, 0x99,
    0xd4, 0x7c, 0x31, 0xe6, 0xab, 0x05, 0x48, 0x9f, 0xd2, 0x8e, 0xc3, 0x14,
    0x59, 0xf7, 0xba, 0x6d, 0x20, 0xd5, 0x98, 0x4f, 0x02, 0xac, 0xe1, 0x36,
    0x7b, 0x27, 0x6a, 0xbd, 0xf0, 0x5e, 0x13, 0xc4, 0x89, 0x63, 0x2e, 0xf9,
    0xb4, 0x1a, 0x57, 0x80, 0xcd, 0x91, 0xdc, 0x0b, 0x46, 0xe8, 0xa5, 0x72,
    0x3f, 0xca, 0x87, 0x50, 0x1d, 0xb3, 0xfe, 0x29, 0x64, 0x38, 0x75, 0xa2,
    0xef, 0x41, 0x0c, 0xdb, 0x96, 0x42, 0x0f, 0xd8, 0x95, 0x3b, 0x76, 0xa1,
    0xec, 0xb0, 0xfd, 0x2a, 0x67, 0xc9, 0x84, 0x53, 0x1e, 0xeb, 0xa6, 0x71,
    0x3c, 0x92, 0xdf, 0x08, 0x45, 0x19, 0x54, 0x83, 0xce, 0x60, 0x2d, 0xfa,
    0xb7, 0x5d, 0x10, 0xc7, 0x8a, 0x24, 0x69, 0xbe, 0xf3, 0xaf, 0xe2, 0x35,
    0x78, 0xd6, 0x9b, 0x4c, 0x01, 0xf4, 0xb9, 0x6e, 0x23, 0x8d, 0xc0, 0x17,
    0x5a, 0x06, 0x4b, 0x9c, 0xd1, 0x7f, 0x32, 0xe5, 0xa8
};
uint8_t CalCRC8(uint8_t *p, uint8_t len)
{
    uint8_t crc = 0;
    uint16_t i;

    for (i = 0; i < len; i++)
    {
        crc = CrcTable[(crc ^ *p++) & 0xff];
    }

    return crc;
}
```

3.2. Measurement Data Analysis

Each measurement data point is consists of a distance value of 2 bytes and a confidence of 1 byte, shown as below:

start character	data length	Radar speed		Start angle		data	end angle		Timestamp		CRCcheck
54H	2CH	LSB	MSB	LSB	MSB	……	LSB	MSB	LSB	MSB	1Byte

measurement point 1			measurement point 2			…	measurement point n		
Distance Value		Confidence	Distance Value		Confidence		Distance Value		Confidence
LSB	MSB	1 Byte	LSB	MSB	1 Byte	…	LSB	MSB	1 Byte

The distance value is in mm. Confidence reflects the intensity of light reflection, the higher the intensity, the greater the confidence is; The lower the intensity, the smaller the confidence is. For white objects within 6m, the typical confidence is around 200.

The Angle value of each point is obtained by linear interpolation of the starting angle and the ending angle. The calculation method of the angle is as following:

step = (end_angle – start_angle)/(len – 1);

angle = start_angle + step*i;

Len is the length of the packet, and the i value range is [0, len].

sales@ldrobot.com

3.3. Reference Example

For example we receive some data as below:

54 2C 68 08 AB 7E E0 00 E4 DC 00 E2 D9 00 E5 D5 00 E3 D3 00 E4 D0 00 E9 CD 00 E4 CA 00 E2 C7 00 E9 C5 00 E5 C2 00 E5 C0 00 E5 BE 82 3A 1A 50

The interpretation is as follows:

start character	data length	Radar speed		Start angle		data	end angle		timestamp		CRC check
54H	2CH	68H	08H	AB	7E	……	BE	82	3A	1A	50

0868H = 2152°/s

7EABH=32427 Be 324.27°

82BEH= 33470 Be 334.7°

measurement point 1			measurement point 2			···	measurement point 12		
Distance value		Confidence	Distance value		Confidence		Distance value		Confidence
E0	00	E4	DC	00	E2	···	B0	00	EA

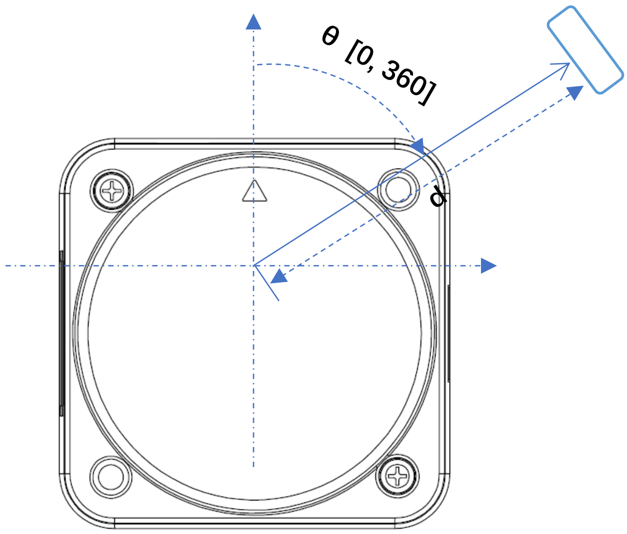
00E0H= 224 mm
Confidence 228

00DCH=220 mm
Confidence 226

00B0H=176 mm
Confidence 234

4. Coordinate System

LD06 uses the left handed coordinate system, the rotation center is the coordinate origin, the front of the sensor is defined as the zero direction, and the rotation Angle increases along the clockwise direction. Please refer to below photo:



sales@ldrobot.com

5. ROS SDK Instructions

ROS (Robot Operating System, abbreviated as "ROS") is an open sourced meta-operating system for robots. It provides the services that an operating system should have, including hardware abstraction, underlying device control, implementation of common functions, inter-process messaging, and package management. It also provides the necessary tools and library functions to get, compile, write, and run code across computers.Please refer to the ROS website for installation steps for each version.*http://wiki.ros.org/kinetic/Installation*

This manual USES the ubuntu16.04 system and the ROS version installed is kinetic.

5.1. Set Access

First, connect the radar to our transfer module (CP2102 serial port transfer module) and connect the module to the computer. Then, open a terminal under Ubuntu and type ls /dev/ttyUSB* to see if the serial port device is connected. If a serial port device is detected, sudo chmod 777/dev/ttyUSB * is used to give it the highest permissions, that is, to the file owner, the group, and other users read, write, and execute permissions.

```
pi@pi:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
pi@pi:~$ sudo chmod 777 /dev/ttyUSB*
pi@pi:~$
```

5.2. Compile the sdk

Access to the sdk_ld_sllidar_ros folder and use catkin_make to compile the source file shown as below.

sales@ldrobot.com

```
pi@pi: ~/sdk_ld_sllidar_ros
pi@pi:~$ cd sdk_ld_sllidar_ros/
pi@pi:~/sdk_ld_sllidar_ros$ catkin_make
Base path: /home/pi/sdk_ld_sllidar_ros
Source space: /home/pi/sdk_ld_sllidar_ros/src
Build space: /home/pi/sdk_ld_sllidar_ros/build
Devel space: /home/pi/sdk_ld_sllidar_ros/devel
Install space: /home/pi/sdk_ld_sllidar_ros/install
Creating symlink "/home/pi/sdk_ld_sllidar_ros/src/CMakeLists.txt" pointing to "/
opt/ros/kinetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/pi/sdk_ld_sllidar_ros/src -DCATKIN_DEVEL_PREF
IX=/home/pi/sdk_ld_sllidar_ros/devel -DCMAKE_INSTALL_PREFIX=/home/pi/sdk_ld_slli
dar_ros/install -G Unix Makefiles" in "/home/pi/sdk_ld_sllidar_ros/build"
####
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
```

After successful compilation, it will show the following interface:

```
Scanning dependencies of target ldlidar
[ 16%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/main.cpp.o
[ 33%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/transform.cpp.o
[ 50%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/slbf.cpp.o
[ 66%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/cmd_interface_linux.cpp.o
[ 83%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/lipkg.cpp.o
[100%] Linking CXX executable /home/pi/sdk_ld_sllidar_ros/devel/lib/ldlidar/ldlidar
[100%] Built target ldlidar
```

5.3. Program Running

After completing the compilation, add the compiled file to the environment variable, and the command is source devel/ sep.bash. This command is to temporarily add environment variable to the terminal, which means that if you open a new terminal, you also need to enter the sdk_ld_sllidar_ros path to execute the command to add environment variable. After adding the environment variable, the roslaunch command finds the ros package and the launch file and runs roslaunch ldlidar LD06.launch.

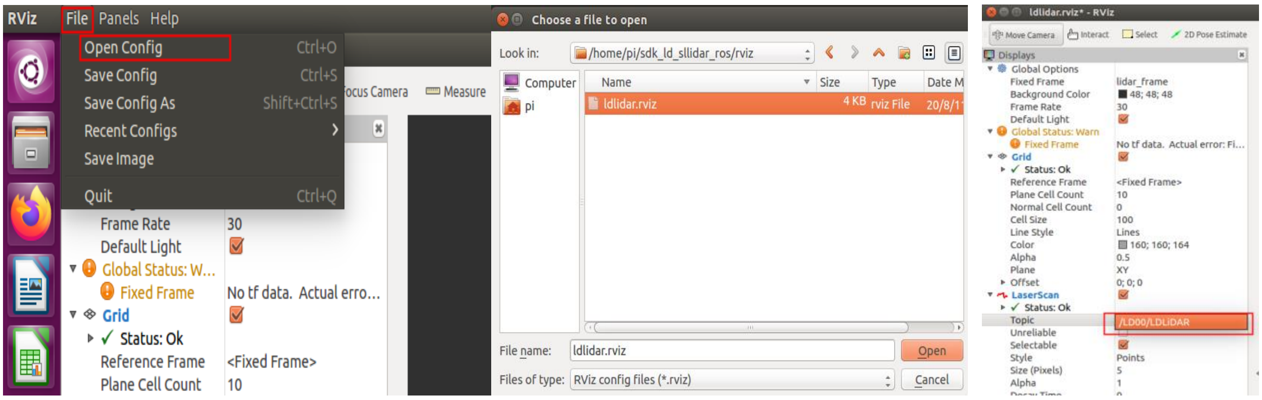
sales@ldrobot.com

```
pi@pi:~/sdk_ld_sllidar_ros$ source devel/setup.bash
pi@pi:~/sdk_ld_sllidar_ros$ roslaunch ldlidar ld00.launch
... logging to /home/pi/.ros/log/0fad1b9c-de11-11ea-b9bb-000c29fc5655/roslaunch-pi-3191.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://pi:42991/

SUMMARY
========

PARAMETERS
 * /product: LD00
 * /rosdistro: kinetic
 * /rosversion: 1.12.14

NODES
  /
    LD00 (ldlidar/ldlidar)

auto-starting new master
process[master]: started with pid [3201]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 0fad1b9c-de11-11ea-b9bb-000c29fc5655
process[rosout-1]: started with pid [3214]
started core service [/rosout]
process[LD00-2]: started with pid [3222]
/dev/ttyUSB0    CP2102 USB to UART Bridge Controller
FOUND LiDAR_LD00
```

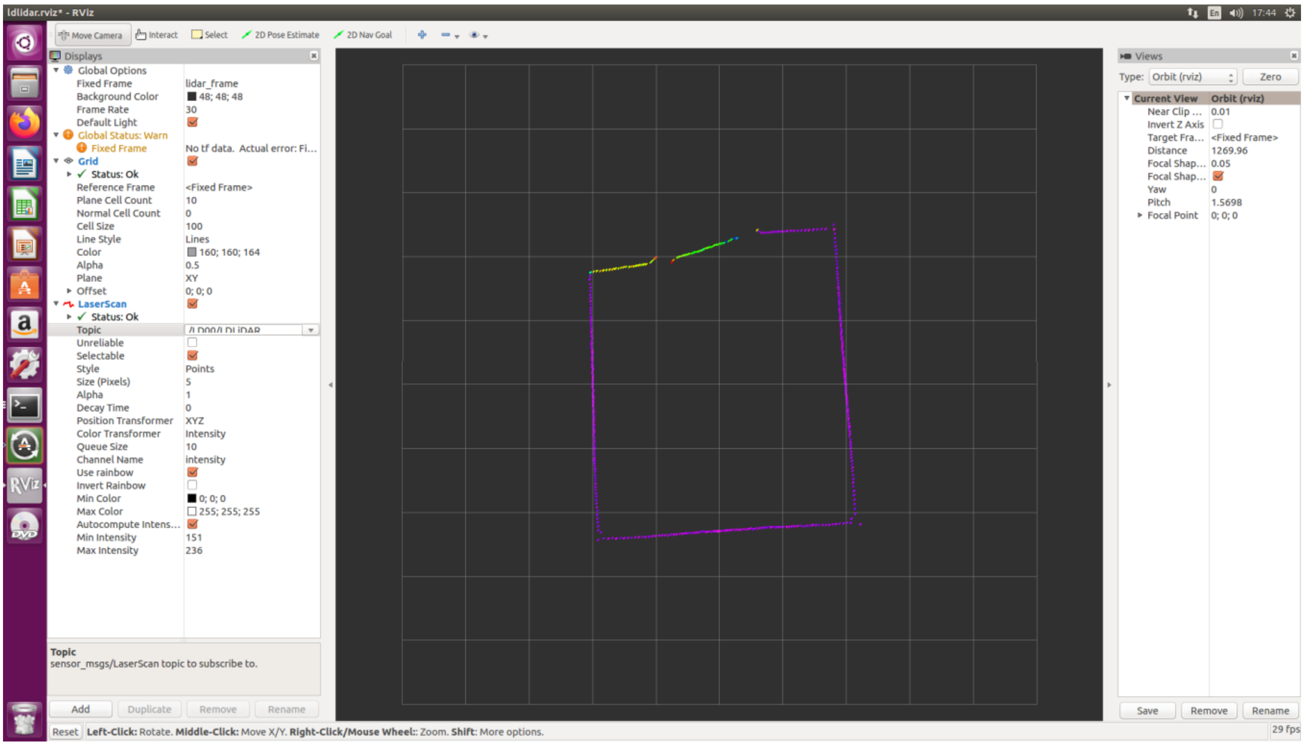
After booting successfully, you will see the information as above circled in red. The program output FOUND LiDAR_LD06 means that the LD06 device is recognized and the ROS node of LD06 is successfully started.

5.4. RVIZ display

rviz is the next common 3D visual tool for ROS, where radar data can be displayed.After the successful run of roslaunch, open the new terminal, enter rosrun rviz rviz, click file->open->Config, then select sdk_ld_sllidar_rosrviz/ldlidar.rviz file, open the configuration file ldlidar.rviz, and click on the LaserScan Topic to select /LD06/LDLiDAR.

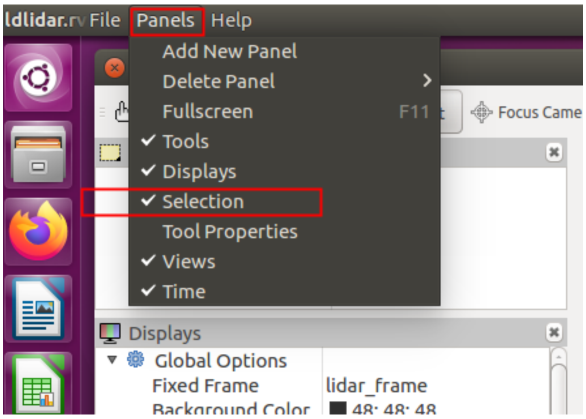


The radar diagram can be displayed normally on Rivz.

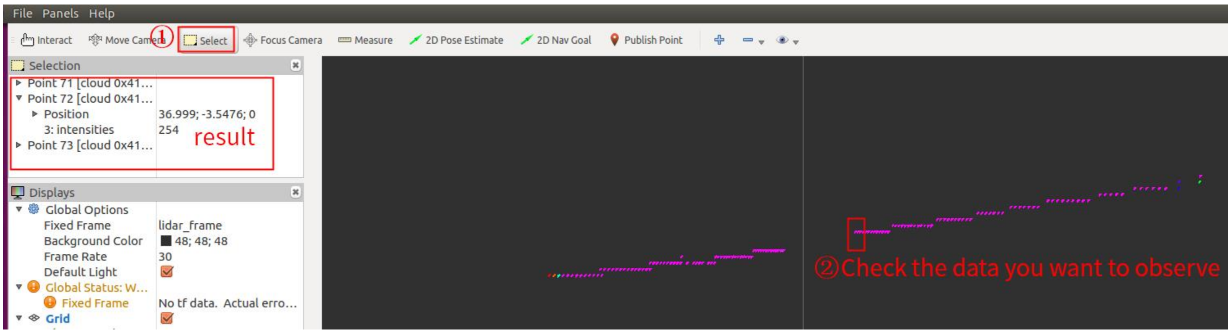


If you need to view the data at a particular point in the radar chart, you can click Panel->Selection in turn.

sales@ldrobot.com



Then click Select and select the data you want to observe, which displays the Position and intensities information for the current point in the top-left window.



sales@ldrobot.com

6. Revision History

version	revision date	Revision contents
1.0	2020-09-01	Initial creation

sales@ldrobot.com