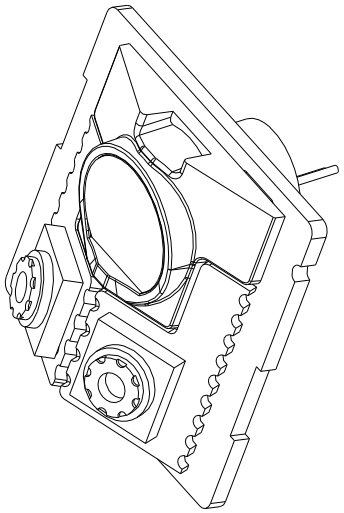
乐动机器人
MOVE SMARTER

LD07 LiDAR

Principle of structured light

Small size , low cost , long working life



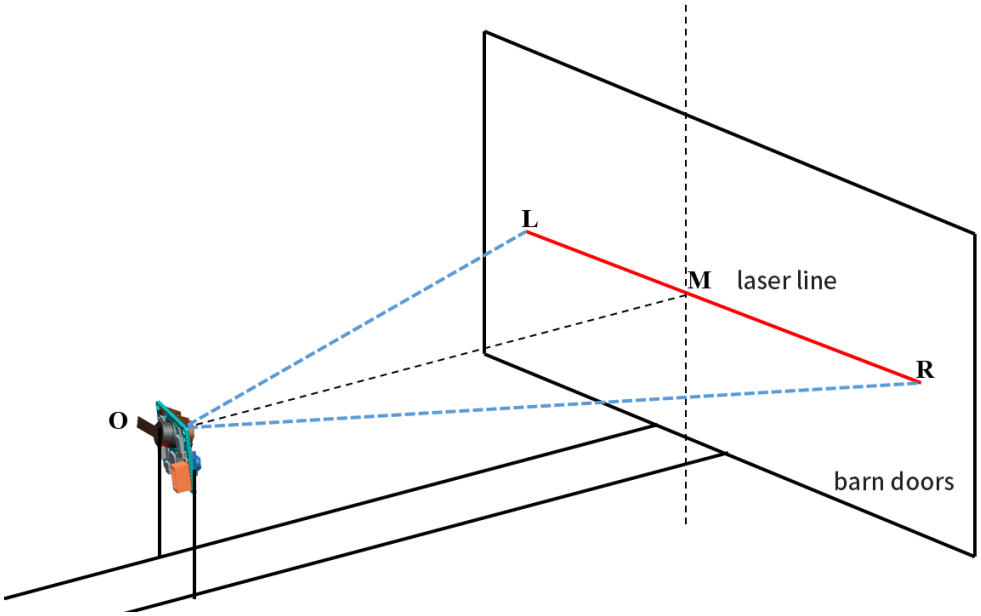
Development manual v1.0

Contents

sales@ldrobot.com

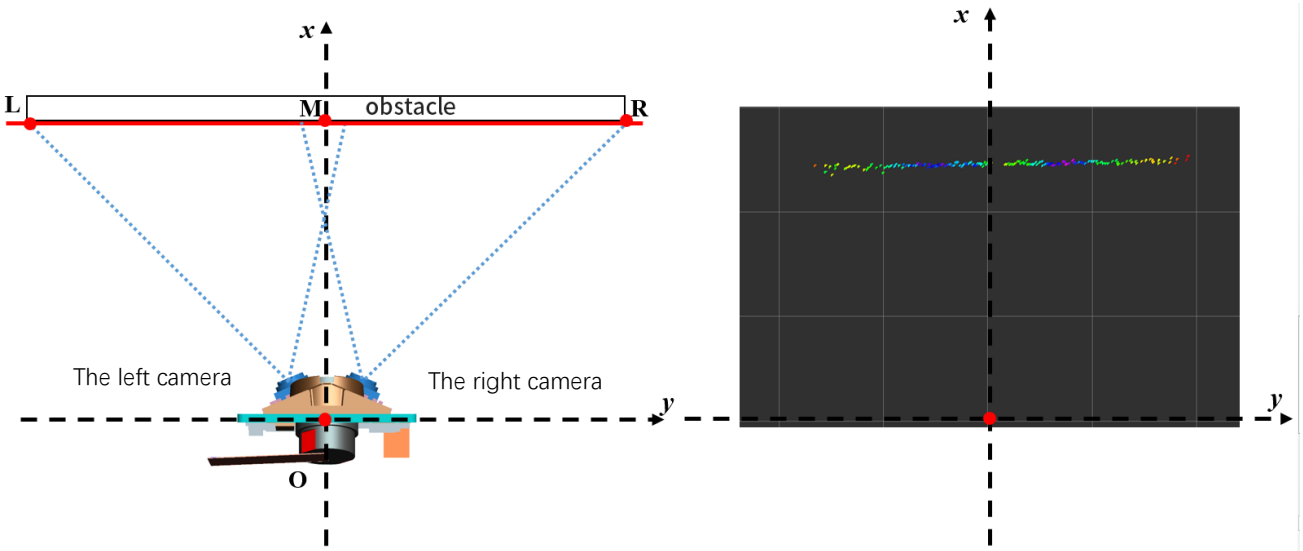
1. Measuring principle

The LD07 is a short-range solid-state radar with a range of 25-300mm. It is mainly composed of a linear laser and cameras. After the laser is sending out by the linear laser device,it is captured by the camera. Based on the fixed structure of the laser and the camera, combining with triangulation measuring principle, we can calculate the distance between the object and LD07. Then according to the camera fixed parameters, we can get the angle value of the measured object in the radar coordinate system. Thus, we obtain the complete measurement data of target object.



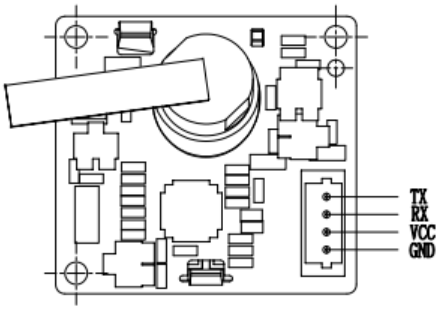
Below please find the LD07 measuring schematics. Point O is the origin of coordinate, and the rang of blue dashed lines is the distance the radar can actually measure.

sales@ldrobot.com



2. Communication Interface

LD07 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	GND	Power Supply	GND	-	0	-
2	VCC	Power Supply	Power Supply	3.1V	3.3V	3.63V
3	RX	Input	UART RX	0V	2.8V	3.63V
4	TX	Output	UART TX	0V	2.8V	3.63V

The LD07 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
921600	8 Bits	1	No	No

sales@ldrobot.com

3. Communication Protocol

After power on, LD07 is in standby state and needs to get specific instructions to enter the working state. The format of packet sent and received by LD07 serial port is shown in the following table:

Start Character				Device address	cmdcode	Packet offset address		Data length field		Data fields	Check code
AAH	AAH	AAH	AAH	Device address	cmdcode	LSB	MSB	LSB	MSB	DATA	CS

- Start Character: 0xAAAAAAAA, Identifies the beginning of the packet in four bytes;
- Device address: The last three digits are available, A0-A2 bits are available, other bits are reserved, and each device occupies one bit in the address byte. Currently at most it supports to set three equipment cascades.

reserved	reserved	reserved	reserved	reserved	A2	A1	A0

- Command codes: Common command codes are as following:

Macro	value	remark
PACK_GET_DISTANCE	0x02	Fetch distance data
PACK_STOP	0x0F	Stop distance data transmission
PACK_GET_COE	0x12	Acquisition of correction parameters
PACK_CONFIG_ADDRESS	0x16	The address configuration includes the number of devices
PACK_ACK	0x10	answerback code

- Packet offset address: When the data is too long, it can be broken into chunks and sent separately, which represent the offset address of the chunk in the entire package;
- Data length field: Represents the length of the data segment, ranging from 0 to 800;

sales@ldrobot.com

- Data fields：Transmit data information;
- Check code：The checksum of the data after the data protocol header is removed;

Start character				Device address	Cmdcode	Packet offset address		Data length field		Data fields	Check code
AAH	AAH	AAH	AAH	Device address	cmdcode	LSB	MSB	LSB	MSB	DATA	CS

4 byte　　6 byte

The following is the calculation process of the check code. Data is the first address of the packet and the data protocol header is removed. Therefore, the first address of the check is data+4, and data_len is the value obtained from the data length field, so the data length of the check is data_len +6.

```
uint8_t checksum = CalCheckSum(data + 4, 6+ data_len);

uint8_t CalCheckSum(const uint8_t *data, uint16_t len)
{
    uint8_t checksum = 0;
    std::size_t i = 0;
    for (i = 0; i < len; i++)
    {
        checksum += *data++;
    }
    return checksum;
}
```

3.1. Configure address command

If only one LD07 device is connected, there is no need to send the configuration address command, which defaults to 0x01. If multiple LD07 devices are cascaded, you need to send the configure address instruction, which is shown below：

Device address：device_address = 0x00

sales@ldrobot.com

cmdcode：PACK_CONFIG_ADDRESS= 0x16

Offset address：chunk_offset = 0x00

Data length：data_len = 0x0000

Refer to the test instruction sent：

Start character				Device address	Cmd code	Offset address		Data length		CS
AAH	AAH	AAH	AAH	00	16H	00	00	00	00	16H

Reference to receive:

Only one device is connected： AA AA AA AA 01 16 00 00 00 00 17

Connect two devices： AA AA AA AA 03 16 00 00 00 00 19

Connect three devices： AA AA AA AA 07 16 00 00 00 00 1D

3.2. Gets the correction parameter command

Before obtaining the distance data, you need to get the parameters for calibration. Each LD07 device has different calibration parameters.

Device address：device_address is the number of devices acquired.

Reserved	Reserved	Reserved	Reserved	Reserved	A2	A1	A0

001 Only get No.1 device
010 Only get No.2 device
100 Only get No.3 device
111 Get devices 1, 2, and 3

Command code：PACK_GET_COE = 0x12

Offset：chunk_offset = 0x00

Data length：data_len = 0x0000

Referring to the test instruction sent：

sales@ldrobot.com

Start character				Device address	Cmd code	Offset address		Data length		CS
AAH	AAH	AAH	AAH	01H	12H	00	00	00	00	13H

Referring to receiving:

Start character				Device address	Cmdcode	Offset address		Data length		coe_k[0]				coe_k[1]				coe_b[0]				coe_b[1]				Points		CS
AA	AA	AA	AA	01	12	00	00	12	00	7B	00	00	00	79	00	00	00	81	13	00	00	84	15	00	00	50	00	BA

0000007BH=123 00000079H=121 00001381H=4993 00001584H=5508 0050H= 80

Converted to a coefficient used in distance calculations:

```
double k0 = (double)coe_k[0]/10000 = 0.0123;
double k1 = (double)coe_k[1]/10000 = 0.0121;
double b0 = (double)coe_b[0]/10000 = 0.4993;
double b1 = (double)coe_b[1]/10000 = 0.5508;
```

K0 and B0 are the distance calculation parameters of left camera; K1 and B1 are the distance calculation parameters of right camera; 80 is the number of distance data points measured by a single camera, indicating totally 160 data points in the two cameras.

3.3. Get the distance data command

Device address: device_address (same as above)

Command code: PACK_GET_DISTANCE = 0x02

sales@ldrobot.com

Offset address：chunk_offset = 0x00

Data length：data_len = 0x0000

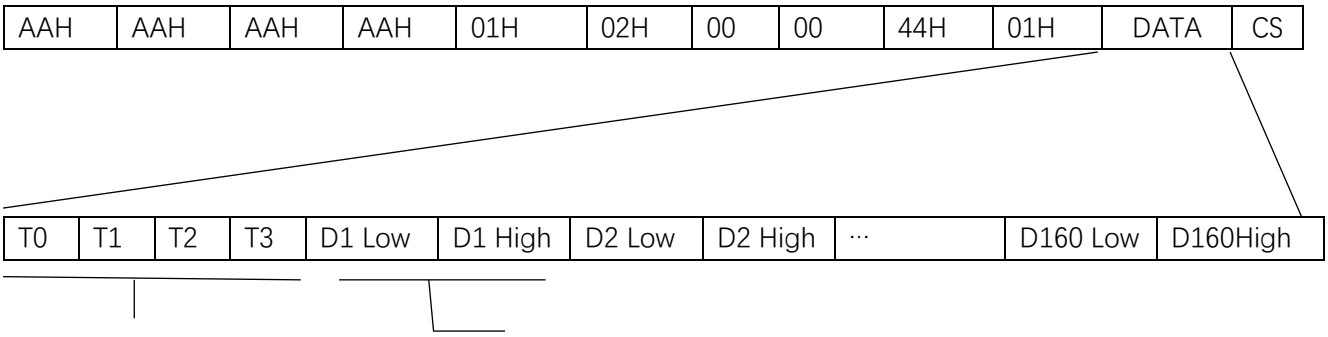
Start character				Device address	Cmdc ode	Offset address		Data length		CS
AAH	AAH	AAH	AAH	07H	02H	00	00	00	00	09H

Reference to receive：

AA AA AA AA 01 02 00 00 44 01 + Data（Length 0x144） + Checksum

AA AA AA AA 02 02 00 00 44 01 + Data（Length 0x144） + Checksum

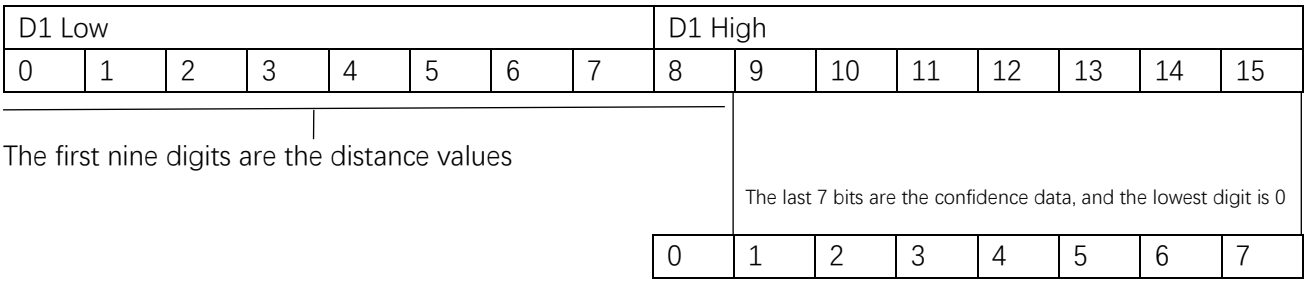
AA AA AA AA 04 02 00 00 44 01 + Data（Length 0x144） + Checksum



0xT3T2T1T0 Constitutes 32-bit timestamp information

D1-D80 represents right camera data，D81-D160represents the left camera data.

The distance information data Uint16_t has the following specific meanings：



sales@ldrobot.com

3.4. Stop distance send command

After getting the distance data, LD07 will send the distance data continuously and will not stop before power off. If the user wants to stop receiving distance data by halfway, the stopping instruction is needed to be sent.

Device address：device_address（same as above）

Cmdcode：PACK_STOP = 0x0F

Offset address：chunk_offset = 0x00

Data length：data_len = 0x0000

Refer to the test instruction sent：

Start character				Device address	Cmd code	Offset address		Data length		CS
AAH	AAH	AAH	AAH	01H	0FH	00	00	00	00	10H

Reference to receive：

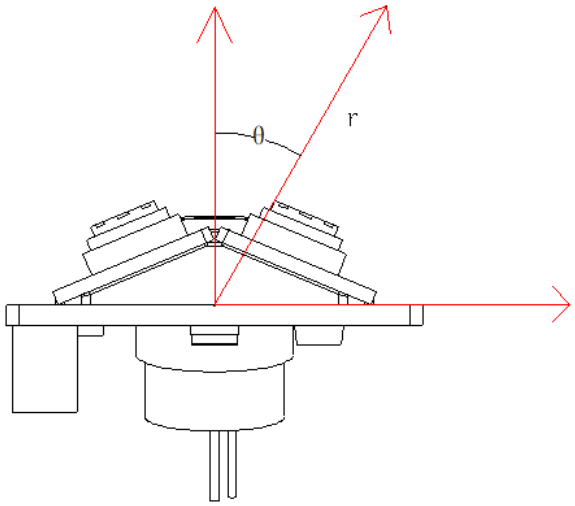
AAH	AAH	AAH	AAH	01H	10H	00	00	02H	00	0FH	01H	22H

After this command send , LD07 will enter standby mode. Laser will stop emit , receive unit will enter sleep mode.

sales@ldrobot.com

4. Coordinate conversion

The coordinate system of LD07 is shown as below picture. The laser exit direction is taken as the front side, the projection of the laser center on the PCB plane is taken as the origin of coordinates, and the polar coordinate system is established with the normal line of the PCB plane in 0 degree. The angle increases gradually going with clockwise direction.



The conversion of the original data transmitted by radar into the coordinate system above requires a series of calculations. The conversion function is as following（Refer to the SDK for details）：

```
int data_amount = pack->data.size() - 4 ;
int n = 0;
for(uint i=0;i<data_amount;i+=2,n++)
{
    data.index = n;
    data.confidence = n;
    uint16_t value = *(uint16_t *)(pack->data.data() + i+4);
    data.confidence = (value>>9)<<1;
    value &= 0x1ff;
    int center_dis = value;
    data.distance= center_dis;
    if(data.distance > 0)
    {
```


sales@ldrobot.com

```
            angTransform(data.distance,n,&data.angle,&data.distance);
        }
    }

void angTransform(uint16_t dist, int n, double *dstTheta, uint16_t *dstDist)
{
    const double k0 = (double)config->coe[0] / 10000;
    const double k1 = (double)config->coe[1] / 10000;
    const double b0 = (double)config->coe[2] / 10000;
    const double b1 = (double)config->coe[3] / 10000;
    const double pi = 3.14159265;
    double pixelU = n, Dist, theta, tempTheta, tempDist, tempX, tempY;

    if (pixelU > 80)
    {
        pixelU = pixelU - 80;
        pixelU = 80 - pixelU;
        if (b0 > 1)         {
            tempTheta = k0 * pixelU - b0;
        }
        else
        {
            tempTheta = atan(k0 * pixelU - b0) * 180 / pi;
        }
        tempDist = (dist - 1.22) / cos((22.5 - (tempTheta)) * pi / 180);
        tempTheta = tempTheta * pi / 180;
        tempX = cos(22.5 * pi / 180) * tempDist * cos(tempTheta) + sin(22.5 * pi / 180) * (tempDist *
sin(tempTheta));
        tempY = -sin(22.5 * pi / 180) * tempDist * cos(tempTheta) + cos(22.5 * pi / 180) * (tempDist *
sin(tempTheta));
        tempX = tempX + 1.22;
        tempY = tempY - 5.315;
        Dist = sqrt(tempX * tempX + tempY * tempY);
        theta = atan(tempY / tempX) * 180 / pi;
    }
    else
    {
        pixelU = 80 - pixelU;
        if (b1 > 1)
        {
            tempTheta = k1 * pixelU - b1;
        }
        else
```


sales@ldrobot.com

```
        {
            tempTheta = atan(k1 * pixelU - b1) * 180 / pi;
        }
        tempDist = (dist - 1.22) / cos((22.5 + (tempTheta)) * pi / 180);
        tempTheta = tempTheta * pi / 180;
        tempX = cos(-22.5 * pi / 180) * tempDist * cos(tempTheta) + sin(-22.5 * pi / 180) * (tempDist * sin(tempTheta));
        tempY = -sin(-22.5 * pi / 180) * tempDist * cos(tempTheta) + cos(-22.5 * pi / 180) * (tempDist * sin(tempTheta));
        tempX = tempX + 1.22;
        tempY = tempY + 5.315;
        Dist = sqrt(tempX * tempX + tempY * tempY);
        theta = atan(tempY / tempX) * 180 / pi;
    }
    if (theta < 0)
    {
        theta += 360;
    }
    *dstTheta = theta;
    *dstDist = Dist;
}
```


sales@ldrobot.com

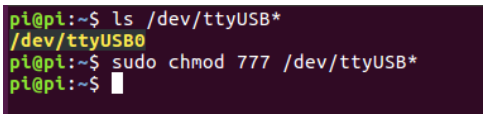
5. ROS SDK Instruction

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.

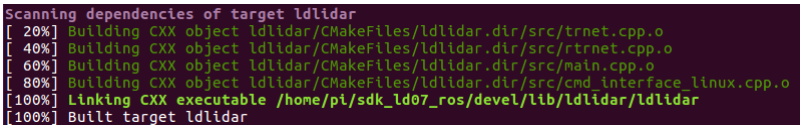


5.2. Compile the sdk

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

sales@ldrobot.com

After successful compilation, it will show the following interface:



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 LD07.launch.

sales@ldrobot.com

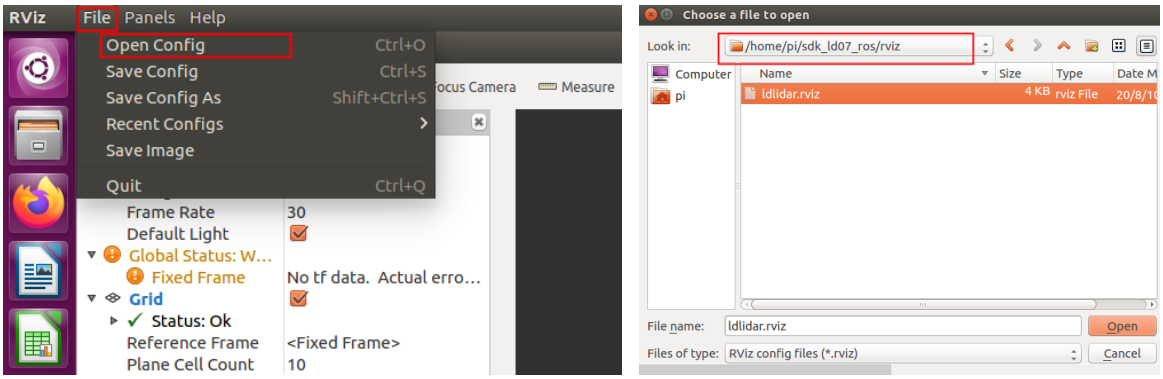
```
pi@pi:~/sdk_ld07_ros$ source devel/setup.bash
pi@pi:~/sdk_ld07_ros$ roslaunch ldlidar ld07.launch
... logging to /home/pi/.ros/log/94aaabde-ddf3-11ea-b307-000c29fc5655/roslaunch-pi-2958.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://pi:42129/

SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.14

NODES
  /
    ld07 (ldlidar/ldlidar)

auto-starting new master
process[master]: started with pid [2969]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 94aaabde-ddf3-11ea-b307-000c29fc5655
process[rosout-1]: started with pid [2982]
started core service [/rosout]
process[ld07-2]: started with pid [2985]
/dev/ttyUSB0    CP2102 USB to UART Bridge Controller
FOUND LiDAR_LD07
get param successfull!!
k0=131  k1=120  b0=5482  b1=3361
Picture pixels: 192*80
get param successfull!!
SEND PACK_GET_DISTANCE CMD
get param successfull!!
k0=131  k1=120  b0=5482  b1=3361
Picture pixels: 192*80
```

After booting successfully, you will see the information as above circled in red. The program output FOUND LiDAR_LD07 means that the LD07 device is recognized and the ROS node of LD07 is successfully started.

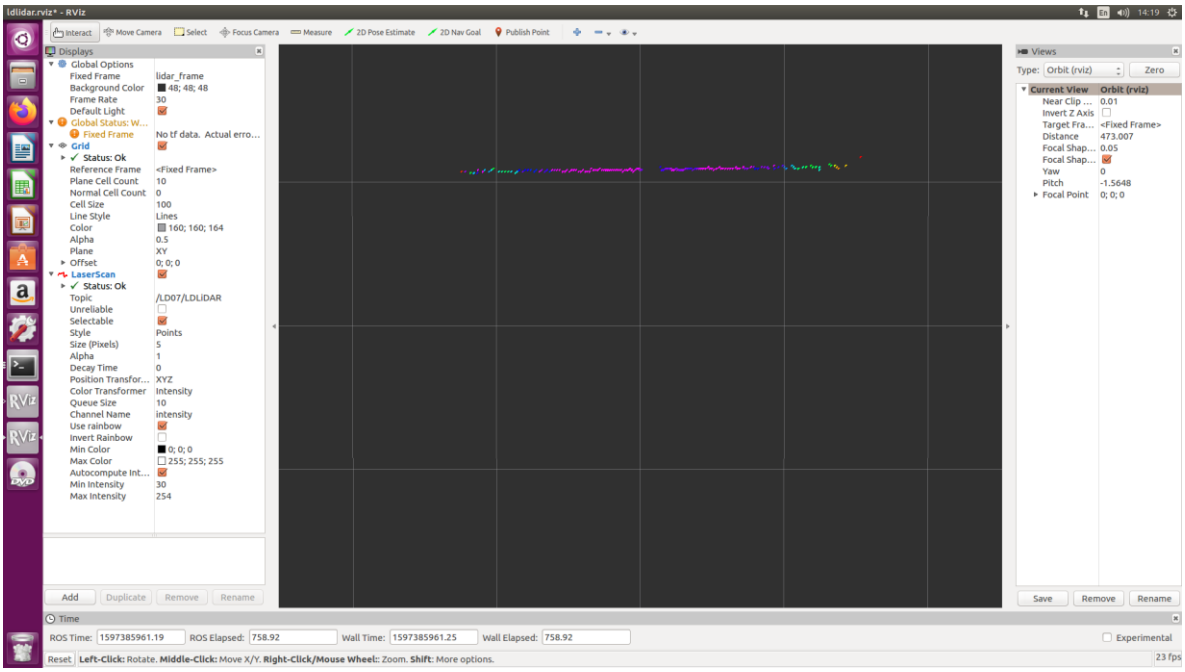
5.4. RVIZ display

rviz is the next common 3D visualization 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 /LD07/LDLiDAR.

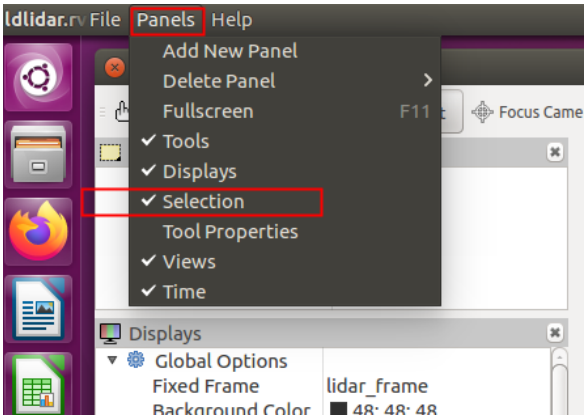
sales@ldrobot.com



The radar map of LD07 is displayed normally on Rivz.

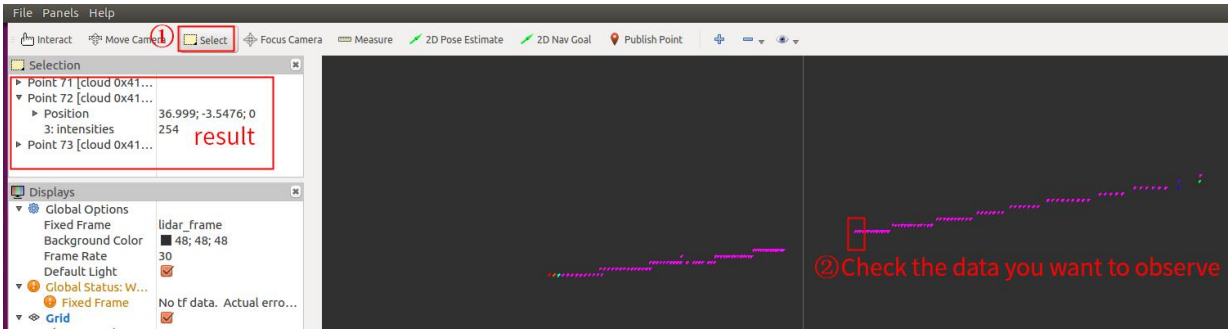


If you need to view the data at a particular point in the radar chart, you can click Panel->Selection in turn.



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