

# ROS based LiDAR autonomous navigation robot design and application

Guoyun Zhu

Shenzhen Pengcheng Technician College, Shenzhen 518000, China

**Abstract:** ROS has become the mainstream of robot application development, in which SLAM synchronous positioning and mapping is the key core technology of robot autonomous navigation. In this paper, a ROS based Lidar autonomous navigation robot is designed and developed for the teaching needs of service robots and artificial intelligence majors. The robot can be used as a professional teaching carrier to meet the practical teaching of professional courses such as Linux, ROS and intelligent robot. With Jetson Nano as the main control, the system is equipped with laser radar, robot arm and handle, which can realize SLAM mapping, navigation and robot arm taking and placing materials. The API application function based on Python is designed and developed to realize the motor control, voltage reading, IMU data reading and odometer data reading functions of the autonomous navigation robot. Serial communication is adopted between the upper computer and the lower computer.

**Key words:** ROS; Lidar; Synchronous positioning and composition; Autonomous navigation

## Introduction

ROS is an open source meta-operating system for robots. It not only provides hardware abstraction, underlying device control, common function implementation, interprocess message passing and package management, but also provides related tools and libraries. For acquiring, compiling, editing code, and running programs across multiple computers for distributed computation. ROS follows the principle of “don’t reinvent the wheel” and supports code reuse in robotics research and development .

## 1. Overall design of ROS robot

ROS robot system can be divided into ROS host (upper computer), auxiliary control board and drive (lower computer), control terminal three parts. ROS host adopts JETSON NANO, calls middleware API through nodes, and communicates with sub-control board to achieve IMU data acquisition, odometer data acquisition, voltage information acquisition, motor drive and so on. ROS host communicates with GUI and APP through the server to transmit control signals and some information. The handle communicates directly with the nodes in ROS to publish the control signals of the motor and steering gear to the corresponding topics, and the base node subscribes to the related topics and sends the control signals to the auxiliary control board to perform the corresponding functions.

## 2. ROS robot hardware design

### 2.1 Hardware Selection

JETSON NANO, due to its small size and powerful functions, can bring enough computing power to the robot terminal, support high-resolution sensors, parallel processing of multiple sensors and other advantages, becoming the first choice for ROS robot host to run the ROS system. The auxiliary control board uses GD32F10RCT6 as the control chip, responsible for driving and collecting data; MPU9250 is selected as the motion sensor of the 9-axis gyroscope, and 2.4G remote control handle is configured.

### 2.2 Hardware system block diagram

The ROS robot system block diagram is shown in Figure 1.

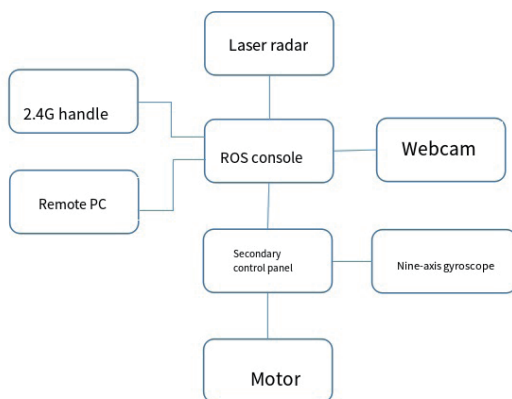


Figure 1: ROS robot system block diagram

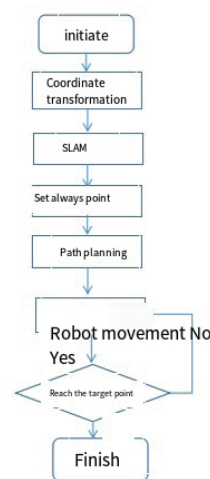


Figure 2: Software development flow chart

### 3. ROS robot software design

#### 3.1 Configuration of software development environment

The development of ROS robot needs to configure the corresponding development environment first. First install the compact linux system on the main board, and install the ROS component on Ubuntu18.06 on the ROS main control board. In order to improve the development efficiency, Has installed the visual code IDE in Ubuntu, and finally put the driver code of the car into the ROS environment to run.

#### 3.2 ROS robot software development process

Create ROS workspace, write topic communication, write robot node code, start node operation program to realize the operation control of robot car and robot arm. The upper computer and lower computer communicate through serial port. The software development flow chart is shown in Figure 2.

### 4. Secondary development of ROs-based robot

#### 4.1 System development tools and environment

##### 4.1.1 VMwar VM Software

VM is a quick tool to build multiple systems, and a virtual machine file that has been built can be directly copied and used.

##### 4.1.2 MobaXterm Remote ssh tool

Using MobaXterm, you can easily and quickly log in to the robot's Linux system through SSH and execute related commands.

##### 4.1.3 notepad++ Simple code editing tool

Notepad++ is a windows text editing tool, it can be very convenient to see the text code Spaces, carriage returns and other placeholders that can not be seen by the naked eye. The use of Notepad++ tool will greatly improve the efficiency of system development.

##### 4.1.4 NoMachine Remote Desktop Sharing Tool

NoMachine is a remote desktop tool that makes it easy and fast to connect your local computer to the desktop of a remote Linux system. Start by connecting the computer to the robot's wireless signal. After opening NoMachine, configure the host name, the robot IP address, click the "Connect" button, and the software will connect the robot and display the remote desktop of its system.

##### 4.1.5 Win32DiskImage Firmware backup tool

This tool can be used to backup the entire image on the TF card as an.img file and save it to the computer.

##### 4.1.6 Raspberry Pi Imager Firmware burning tool

Use the Raspberry Pi Imager to easily and quickly burn the system image onto the TF card.

#### 4.2 Secondary development of ROS robot

The ROS robot system provides Middleware API middleware, which contains two classes MiddleWare\_() and SERVO\_(). MiddleWare\_() mainly controls motor movement, car speed, gets odometer, gets IMU data, gets voltage value, etc. SERVO\_() mainly controls the steering gear joint, the mechanical arm mode, and obtains the status information of the steering gear.

##### 4.2.1 MiddleWare\_()API function

(1)Init() function: driver board initialization, start to upload data, no parameters.

(2)SetParams() function: set encoder resolution, wheel radius, left and right wheel base /2, front and rear wheel base /2.

(3)SetVelocity() function: set the speed in front of the car to be positive, the speed on the left side of the car to be positive, and the car turning counterclockwise to be positive.

(4)SetWheelSpeed() function: set the left front wheel speed, the right front wheel speed, the left rear wheel speed, the right rear wheel speed.

(5)GetOdom() function: set mileage value.

(6)ClrOdom() function: clear the mileage information.

(7)GetIMU() function: read the angular acceleration data, linear acceleration data and geomagnetic data of 9-axis gyroscope MPU9250.

(8)GetWheelSpeed() function: read the left front wheel speed, right front wheel speed, left rear wheel speed, right rear wheel speed.

(9)GetBattery() function: Return the actual voltage value, floating-point number.

(10)LowPwr() function: The driver board comes with a buzzer, as a low battery alarm function, can also be used as a buzzer function.

##### 4.2.2SERVO\_()API function function

(1)serial\_status() function: Obtain serial port initialization status.

(2)init() function: initializes to obtain the current information of all servos.

(3)release() function: Releases servo serial port resources.

(4)setAsyn() function: enable/disable the background servo information synchronization function.

(5)nav\_mod() function: the robot arm enters the folding mode, and does not block the radar when navigation is built.

(6)def\_mod() function: the robot arm enters the default mode and is in upright state.

(7)set\_deg() function: set the ID number and Angle value of the steering gear.

(8)get\_status() function: read the status information, including the id of the steering gear, target coordinate, current coordinate, torque, current, speed, temperature, motion state, steering gear state.

- (9)readback\_mod() function: the steering gear corresponding to id number enters the read-back mode.
- (10)normal\_mod() function: the steering gear corresponding to id number enters the normal execution mode.

## 5. Application of ROS robot in teaching

During the development of ROS robot system, Ubuntu18.06 with ROS components installed has been made into a virtual image file, which can be loaded after VMware software is installed on the PC to realize the learning of relevant knowledge and skills. Based on ROS robot system, artificial intelligence and service robot professional construction course system has been optimized, and four courses “Linux Operating System”, “PYTHON Language Programming”, “ROS System Design and Simulation” and “Intelligent Robot Application” have been developed and offered.

### 5.1 Linux Operating System

The course design is based on the training of high-skill applied talents in vocational colleges to determine the course objectives and design the course content; The structure of students’ knowledge, skills and quality is designed with the cultivation of technical application ability as the main line, and the teaching content system is constructed with application as the theme and characteristic. Through the study of this course, students are required to be familiar with the basic operation of Linux operating system, master the configuration management of Linux operating system, software use and programming environment deployment.

### 5.2 Python Language Programming

Through the “Python Language Programming” course, students can proficiently use python language to program design, and use programs to control hardware devices. Proficient in python language programming, master python programming methods, proficient in python language sequence control, judgment control, loop control and other control methods to lay a good foundation for artificial intelligence thinking.

### 5.3 ROS System Design and Simulation

This course covers the basic knowledge of ROS system, the installation of ROS system, the framework of ROS operating system, the overall architecture of ROS, the use of command line tools, the startup of ROS nodes, the realization of topic communication mechanism, and the realization of service communication mechanism. Through the study of this course, students can master the internal working mechanism of ROS. To be familiar with the design and development of ROS system step by step from the easy to the difficult.

### 5.4. Comprehensive Application of Intelligent Robot

With the foundation of the previous courses, students can enter the Comprehensive Application of Intelligent Robots course to learn, use ROS robots to complete visual perception processing, robot motion control, robot map construction and path planning, customize scenarios to complete the design and function realization of different application scenarios, and improve the comprehensive application and development ability of ROS robots.

## Epilogue

This paper proposes and designs a software and hardware development and design method for ROS based lidar autonomous navigation robot, and gives the secondary development process and method for ROS robot. Through practical teaching application, the performance and function of the robot are tested, which has strong universality. It can be applied to the training of talents in artificial intelligence and ROS robot application in vocational colleges and universities, and has good practicability and economic value. In the future, supporting teaching resources can be developed for this equipment to better serve the training of highly skilled robot talents.

## References:

- [1] Weiwei Sun,Ling Wang,Min Huang,Laser Navigation System for Mobile Robot Based on Robot Operating System [J]. Science Technology and Engineering, 2019, 19 (20) : 203-210.
- [2] Haixing Zhong, XU Lunhui, GUO Yating, ROS Based Control System Design for laser-guided Mobile Robot [J]. Automation and Instrumentation, 2018,33 (12)