Light Trail Maze

——Autonomous Mobile Robot for Maze Navigation and Mapping Based on 2D LiDAR

Reporter: Group 51

Report Time: 2025.10.23

Contents

01 Team Introduction

Team Members and Task Assignment - Hardware Group

Jiayi Wang
Team Leader

Telecommunications Engineering with Management

Coordinates and arranges the division of labor, responsible for team logistics support, vehicle construction, code writing, implementation of LiDAR functions, and joint debugging of LiDAR code.

Xinran Li
Member

Telecommunications Engineering with Management

Responsible for vehicle construction, writing LiDAR code, implementing LiDAR detection and data transmission functions, and creating PPTs and video editing.

Junyi Yan
Member

Telecommunications Engineering with Management

Responsible for vehicle construction, completing Bluetooth and LiDAR code writing, running and debugging the entire code, testing the basic functions of the vehicle, and improving related bugs.

Zixun Tang
Member

Telecommunications Engineering with Management

Responsible for vehicle construction, completing sensor code writing, refining and improving the PID algorithm, and debugging the basic functions of the vehicle.

Yanchen Longyu
Member

Electronic Information Engineering

Responsible for vehicle construction, completing sensor code development and debugging, and jointly debugging the transmission path and efficiency between sensors, LiDAR and Bluetooth.

Yunpeng Fu
Member

Electronic Information Engineering

Responsible for vehicle construction, completing Bluetooth code development, parsing data returned by LiDAR, and issuing further command operations to the motor.

Team Members and Task Assignment - Software Group

Shengyu Yang

Team Leader

Intelligent Science and Technology

Responsible for the backend mapping and navigation of the car, creating a visual UI interface to enable the car to smoothly carry out mapping and pathfinding work

Shengjie Fang

Member

Internet of Things

Mainly responsible for optimizing the entire process of SLAM mapping and A * pathfinding core algorithms, improving the positioning accuracy, map reliability, and path planning efficiency of the vehicle in complex scenes through technical iterations

Wanying Xing

Member

Internet of Things

Responsible for building the car, debugging its basic functions, coordinating and improving project processes, and writing PowerPoint presentations to report on the work

Yanqi Tan

Member

Internet of Things

Responsible for code writing,algorithm design, writing requirement documents and PowerPoint creation

02 Project Objectives

Project Objectives

Maze Navigation

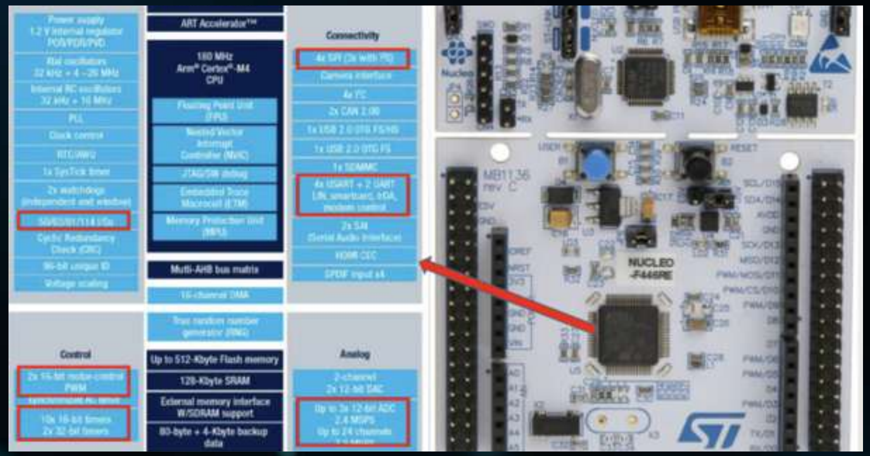
Use 2D LiDAR to realize real-time environment mapping, independently explore the maze and return to the starting point, and verify the hardware collaboration capability.

Hardware Collaboration

Integrate STM32, motors, LiDAR and Bluetooth modules to form a complete closed-loop control, ensuring real-time synchronization of data flow and command flow.

03 Hardware Architecture

核心组件



01

STM32F446RE

As the central controller, it is responsible for sensor data collection, PID motor control, and Bluetooth communication with the PC terminal, ensuring real-time performance.

02

RPLIDAR C1

360° scanning provides angle-distance data, supports SLAM mapping, with a scanning frequency of 10Hz and an accuracy of ±30mm.

03

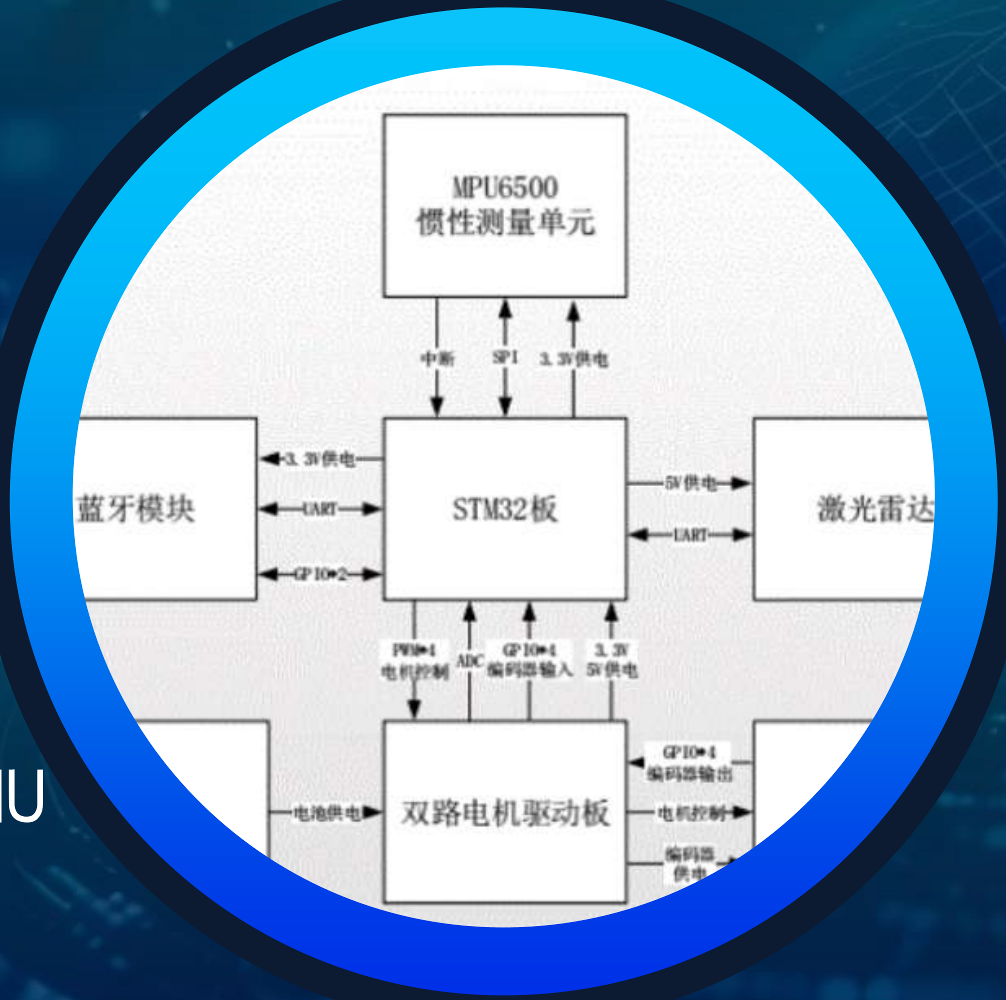
Motor and Drive

520 encoder motor + AT8236 driver enables closed-loop speed control, with an encoder resolution of 1200 pulses per revolution.

Schematic Diagram of System Architecture

STM32F446RE

Responsible for real-time sensor processing, motor control, and low-level communication with the host.



AT8236motor driver

Controls the 520 gear encoder DC motor (providing mobile capability with encoder feedback for speed control and odometry) to achieve precise speed and direction control.

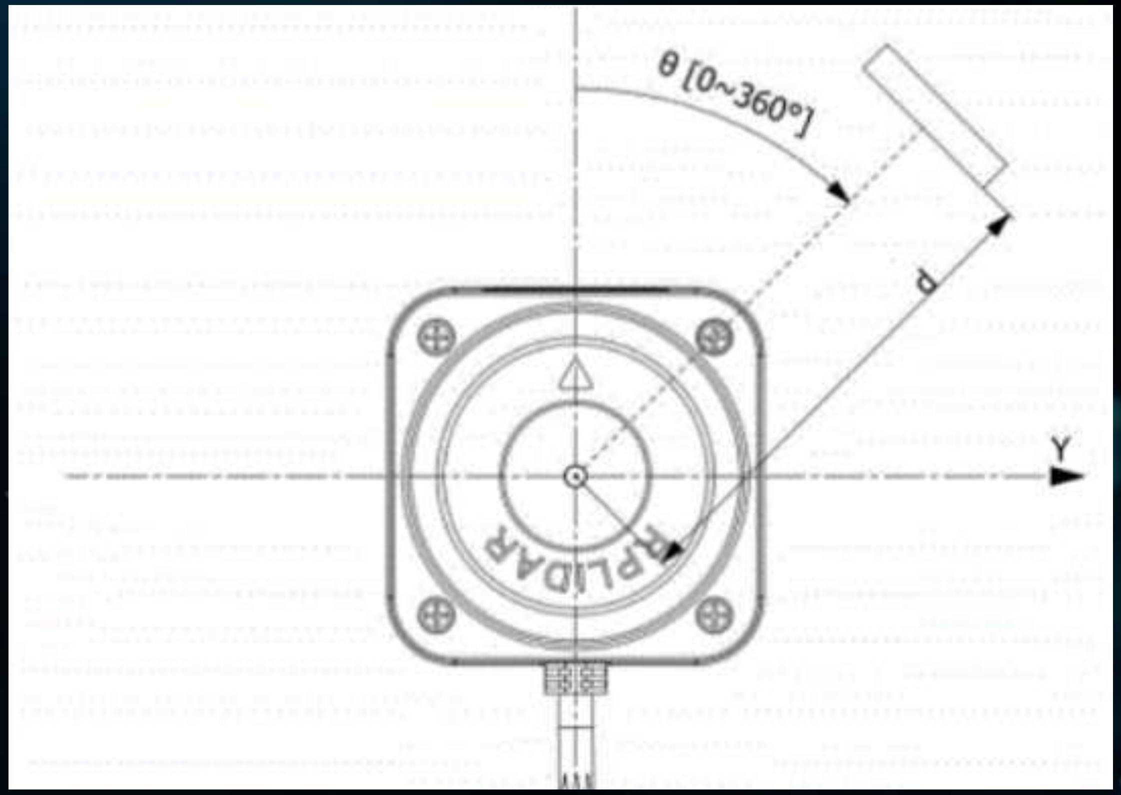
SLAMTEC RPLIDAR C1/MPU6500 IMU

2D LiDAR sensor captures angle and distance data of the surrounding environment, supporting real-time mapping and SLAM. / 6-axis inertial measurement unit (IMU) for direction tracking and odometry support.

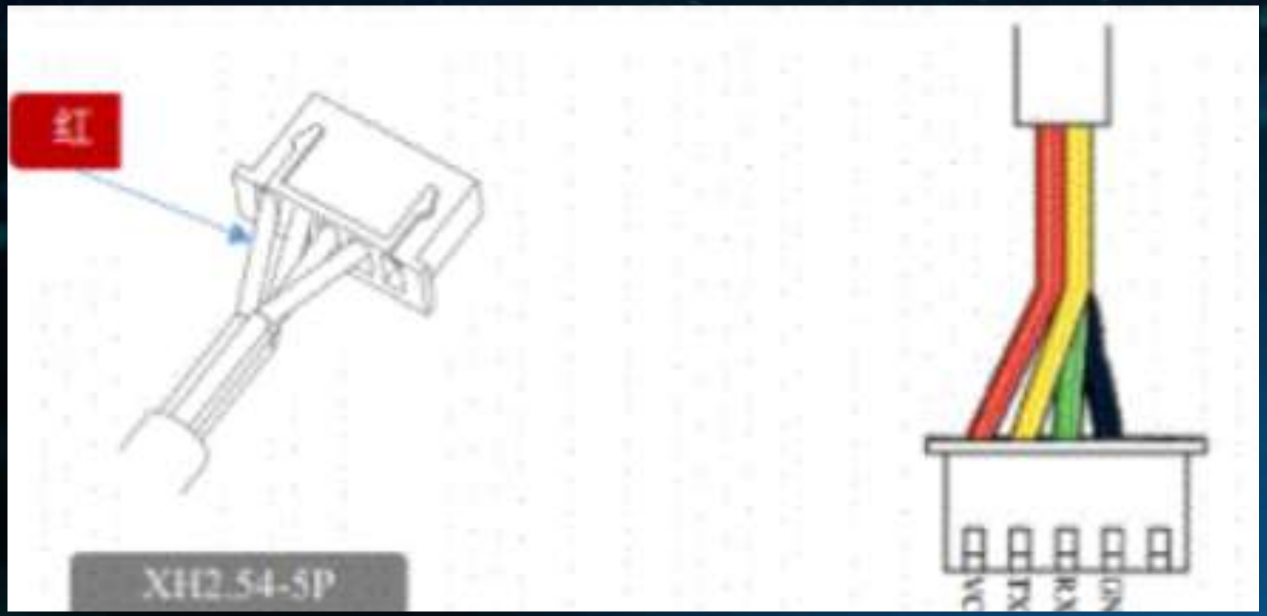
HC-04Bluetooth module

Enables wireless communication between the robot and the host.

2D LiDAR Sensor



A 2D LiDAR sensor using laser time-of-flight ranging technology captures angle and distance data of the surrounding environment, supporting real-time mapping and SLAM.



LiDAR Function Implementation

UART Communication

Reads LiDAR data packets via serial port at a rate of 115200 bps, and parses the angle, distance, and intensity fields.

Data Forwarding

Sends the processed data to the PC via Bluetooth for the SLAM algorithm to update the map in real time.

Data Filtering

Uses median filtering to remove outliers and improve mapping stability, with a filtering window length of 5 frames.

LiDAR Data Processing and Analysis

LiDAR Data Collection and Preprocessing

Obtain 360-degree environmental sensing data (angle and distance) through LiDAR sensors, and convert the data format to facilitate subsequent processing and analysis.

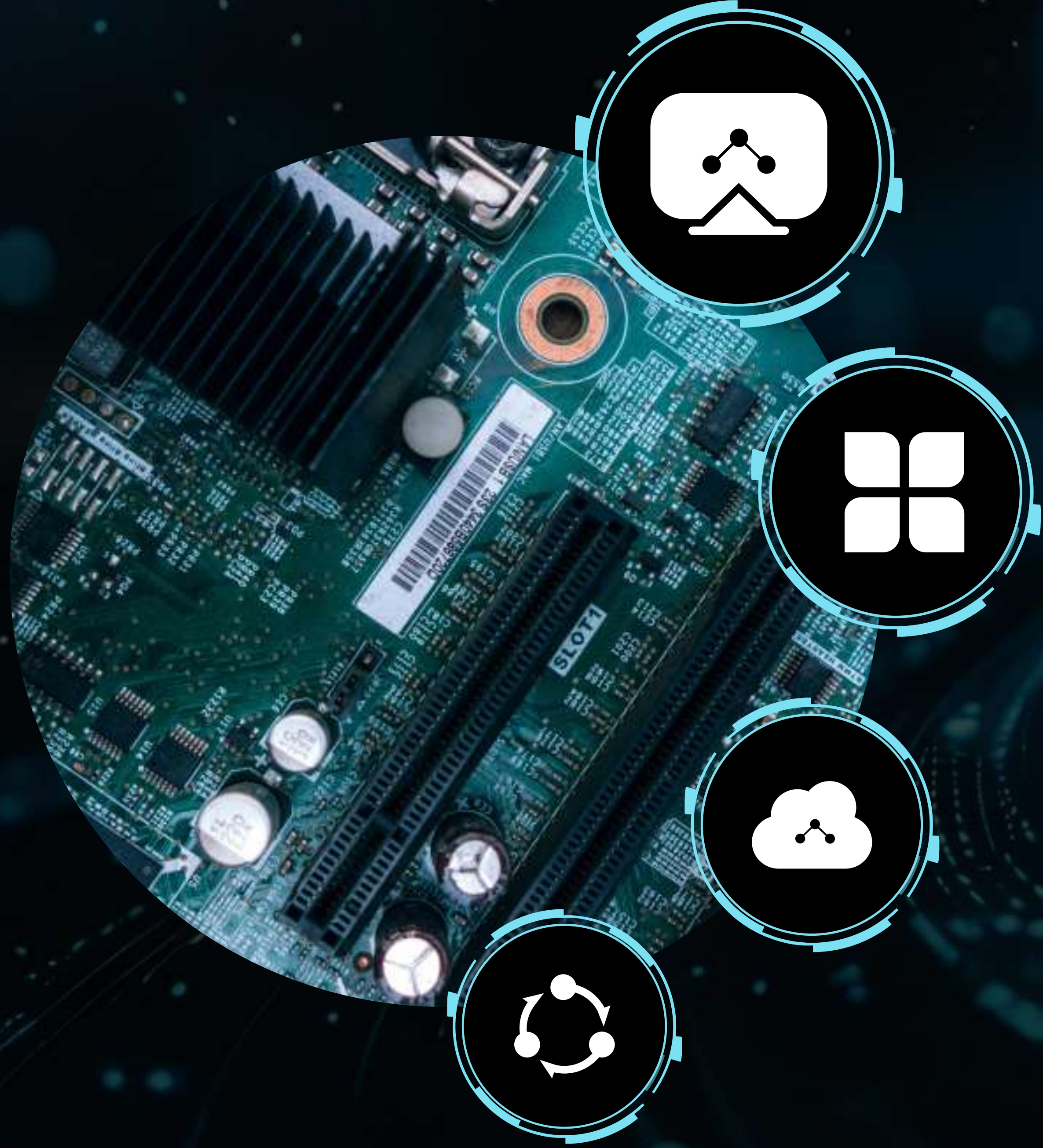
Algorithm Implementation and Performance Evaluation

Use SLAM algorithm for real-time map update and path planning. Through experiments and tests, verify the system's performance indicators, and evaluate the accuracy, stability and real-time performance of LiDAR function implementation.

Feature Extraction and Data Fusion

Extract key features from the preprocessed data, such as angle, distance and intensity, and use these features for further analysis and processing. Fusion of LiDAR data with other sensor data (such as IMU, encoder, etc.) to improve data accuracy and reliability, and enhance the overall performance of the system.

Expected Process



Expectation 1

2D LiDAR sensor interface: Collect continuous 360 degree environmental sensing data (angle and distance) and send the raw data to the host for SLAM mapping, visualization, path planning, and autonomous navigation.

Expectation 2

Interface with MPU6500 IMU: Read directional data, including three-axis gyroscope angular velocity, for calculating the yaw angle (Yaw) of the vehicle body, as a key feedback for directional closed-loop control.

Expectation 3

Interface with motor encoder: Read AB phase pulses through the encoder mode of the timer, accurately calculate the motor speed and rotation direction, and use them as feedback signals for speed closed-loop control.

Implementing Path and Future Optimization

Optimization Direction 1

Data fusion: For more advanced applications, the perception of gravity by accelerometers can be introduced, and the short-term accuracy of gyroscopes and the long-term stability of accelerometers can be combined through algorithms such as Kalman filtering to completely solve the problem of accumulated errors.

Optimization Direction 2

Reasonable configuration: In the initialization function of mpu6500. c, we selected the appropriate digital low-pass filter (DLPF) bandwidth through register configuration. This strikes a balance between filtering out high-frequency noise and ensuring a sufficiently fast response speed.

Optimization Direction 3

Software limiting: We expect to set maxOutput for the output of the angle PID in Angle_pid. c, which indirectly limits the maximum speed difference applied to the wheels and thus also limits the maximum rotational angular velocity of the car at the software level, preventing it from easily exceeding the hardware range.

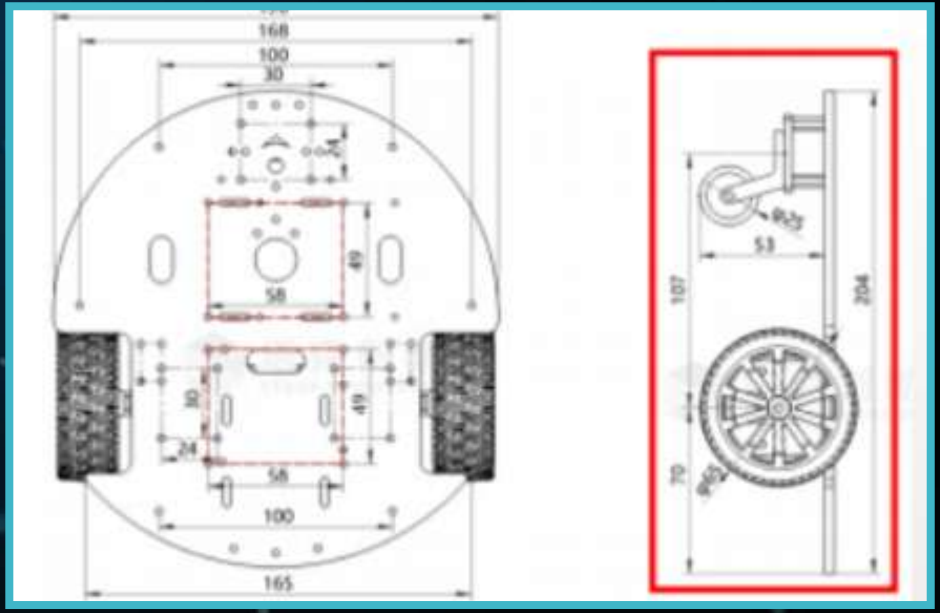
Optimization Direction 4

Algorithm robustness: Although not currently in use, sliding average or first-order low-pass filtering algorithms can be easily added to the control loop to smooth the raw readings of the gyroscope, further reducing the impact of instantaneous noise and vibration data on PID calculations.

Optimization Direction 5

Static zero calibration: We have designed the calibrate_gyro() function during the initialization phase of the code. This function runs when the car is powered on and kept completely stationary, and calculates the average drift of the gyroscope's Z-axis (gyro z-bias) through thousands of samples. In the subsequent control loop, we subtracted this bias from each raw reading, eliminating the cumulative impact of static drift at the root.

Implementing Path and Future Optimization



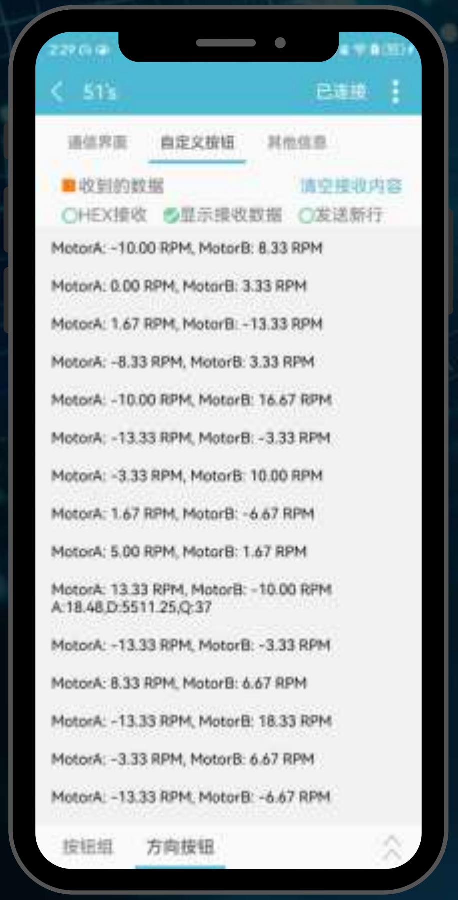
Expected Algorithm Model

- Control model: The industry's mature Cascade PID Control model is adopted to achieve dual closed-loop control of speed and direction.
- Outer Loop - Angle Loop:
- Input: Target angle (from remote control command) and current angle (from MPU6500 gyroscope).
- Algorithm: An independent angle PID controller calculates a velocity correction (rpm_correction) for correcting heading based on angle error.
- Function: Responsible for maintaining stable heading of the vehicle, resisting external interference, and achieving precise straight-line driving and steering.
- Inner loop - Speed Loop:
- Input: The target speed of the left and right wheels (obtained from the basic target speed ± speed correction) and their real-time speed (from the encoder).
- Algorithm: Two independent speed PID controllers calculate the final PWM duty cycle applied to the motor based on the speed error.
- Function: Responsible for accurately executing dynamic speed commands issued by the outer ring, ensuring the precision and stability of wheel speed.

Specific technical framework

- Core controller: High performance STM32F446RE MCU is used as the main control chip.
- Software architecture: Real time control system based on timer interrupt driver. The high priority PID closed-loop control logic is placed in the 100Hz interrupt of TIM5, and low priority tasks such as Bluetooth command reception and data upload are placed in the main loop of the main function, achieving the separation of real-time and non real time tasks and ensuring stable control and timely response.
- Driver and Execution:
- Motor drive: Drive an AT8236 dual H-bridge motor drive module in L298N mode through four PWM channels of TIM3. c, achieving independent speed and direction control of the left and right motors.
- Data feedback: TIM2 and TIM4 work in encoder mode and provide real-time feedback on wheel speed; The I2C1 bus is used to communicate with the MPU6500 and obtain vehicle attitude data.
- Human computer interaction: Use USART1 to connect HC-04 Bluetooth module, establish wireless serial communication with mobile phone or PC, for receiving remote control commands and sending real-time status data of the car.

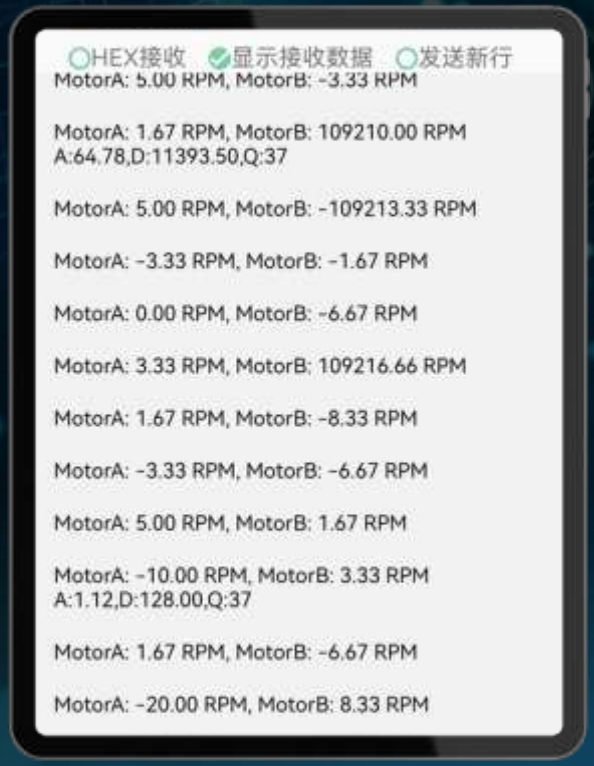
Radar Detects the Environment and Transmits Data



Bluetooth Communication Implementation

Implementation Path

We communicate with the HC-04 Bluetooth module through USART1 (9600bps) and use DMA for efficient data transfer. And we have added interrupt driven instructions to process single byte control instructions in real-time (such as 0=stop, 1=forward), and drive the motor through PWM (TIM3). At the first connection, the car will send a "Connected" prompt and actively feedback sensor data (such as LIDAR angle/distance) through UART.



Motor Control Implementation - PID Control

Speed Loop

Read the encoder pulse, calculate the actual speed, and output the PWM duty cycle through PID by subtracting it from the target speed.

Position Loop

Combining IMU attitude data, closed-loop correction of displacement error, positioning accuracy ± 5mm.

Fault Protection

Monitor motor overcurrent and stalling, trigger emergency braking and report status to ensure hardware safety.

Overall Implementation Path

01

Radar Scanning

The SLAMTEC RPLIDAR C1 has the characteristics of a wide ranging range (0.15-12m) and a high scanning frequency (10Hz), making it suitable for precise mapping in maze environments

02

Return Data

The raw data (angle and distance) collected by the LiDAR is preliminarily processed by the STM32F446RE microcontroller and transmitted in real-time to the host PC through the Bluetooth module for SLAM algorithm to construct an occupied grid map of the environment, supporting the robot to identify feasible paths and boundaries in unknown mazes

03

Bluetooth Communication

Transmit data packets to the host via Bluetooth, including 2D LiDAR scanning data (angle and distance) and odometer data (x, y,...)Receive navigation commands (waypoints/trajectories) from the host to perform real-time motor control

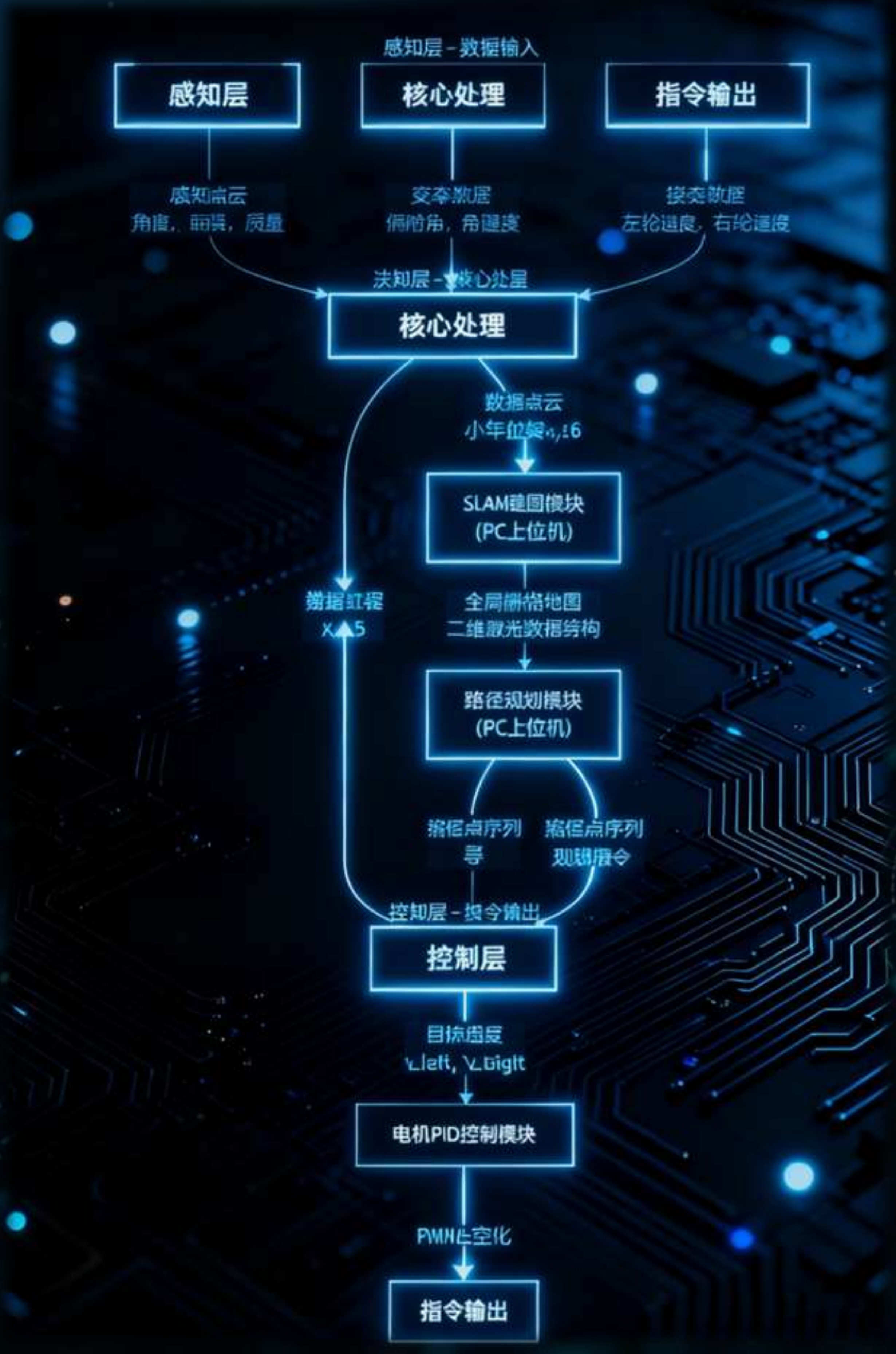
04

Motor Control

Control the 520 gear encoder DC motor through the AT8236 motor driverProcess navigation commands (waypoints/trajectories) and use PID control for real-time motor control of speed/position.

04 Software Architecture

Overall System Design and Data Flow



01

Clear Hierarchy

It demonstrates the classic machine software architecture from "perception" to "decision-making" and then to "control".

02

Data Abstraction

The arrows in the figure no longer transmit hardware signals (such as UART, PWM), but software defined data structures (such as point clouds, maps, waypoints), which reflects the core role of software.

03

Modular Design

Each rectangular box represents an independent software module with a single responsibility, making it easy for collaborative development and debugging.

Core Algorithm and Decision Logic

Exploring Status

This is the default state of the car. In this state, it does not care about the global optimal path, the only goal is to traverse unknown areas and construct a complete map.

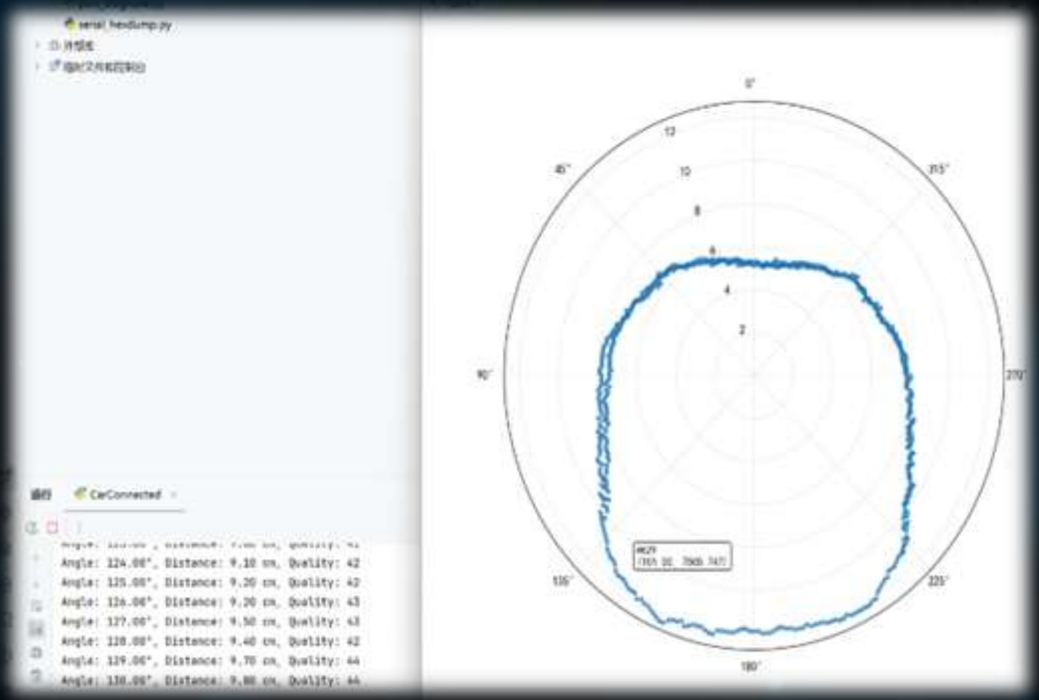
Planning Status

This is an instantaneous state. At this moment, the car pauses in motion and uses the fully constructed grid map to calculate the global optimal path from the current position to the target point.

Tracking state

Accurately and stably move along the planned sequence of path points until the endpoint.

Implementation，Debugging，and Innovation



Radar Data Analysis

- Single frame 360 degree scanning to obtain high-density, high-precision distance point clouds
- Can accurately interpret data from specific angles
- The quality data of this manuscript is a reliable foundation for SLAM mapping and obstacle detection algorithms

01 Architecture Innovation

Modular and decoupled design to improve development and debugging efficiency.

02 Decision Innovation

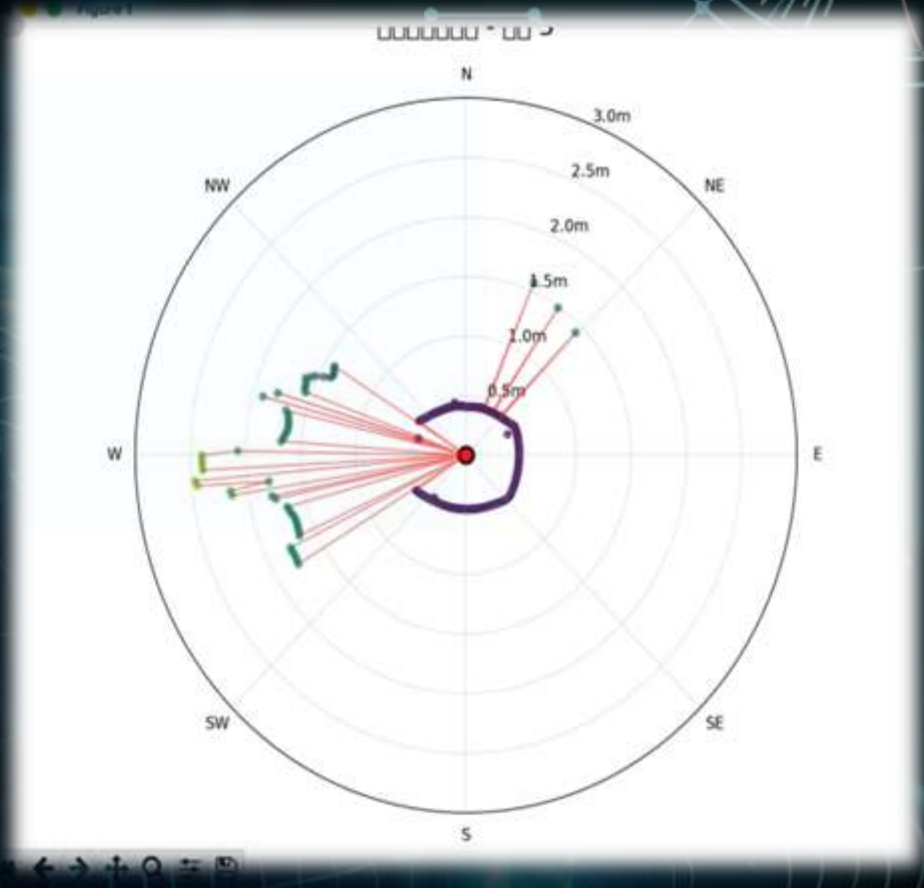
State machine management enhances the robustness of system behavior.

03 Data Innovation

Multi sensor practical fusion effectively solves the problem of odometer drifting.

04 Algorithm Innovation

The organic combination of exploration and planning algorithms is used for deep optimization of maze scenes.

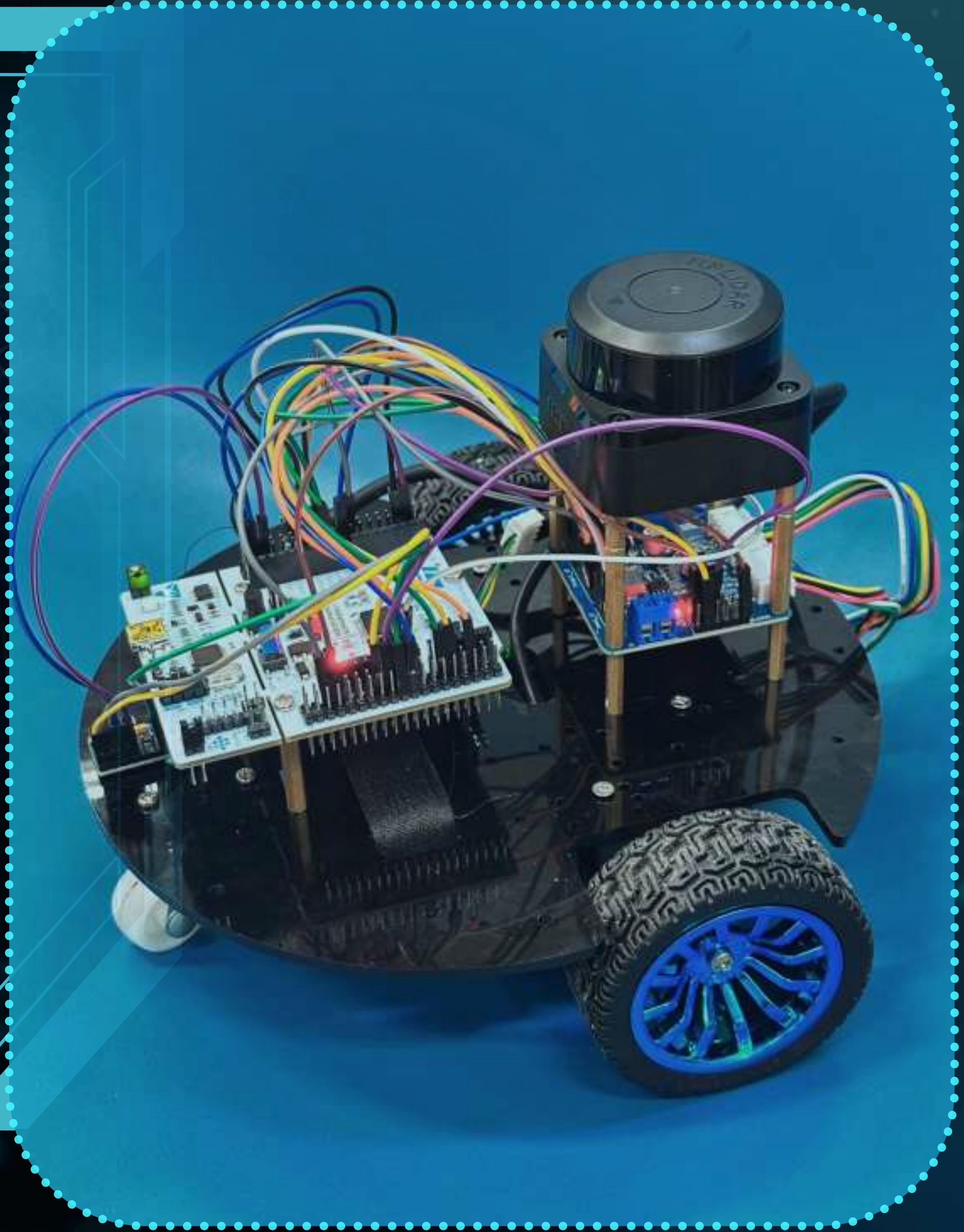


Real Time Radar Perception

- Real time presentation of LiDAR point cloud data by upper computer softwareThe polar coordinates clearly outline the outline of the maze walls, with a detection range of up to 3m.Proved the real-time and stability of data collection, transmission, and visualization pipelines.

05 Project Achievements

Achievement Demonstration Video



Demonstration over

Thanks for watching