**【智能轮椅辅助控制装置】**

**软件设计说明书**

**SDD7**

**version 1.7.0**

分工说明

|  |  |  |
| --- | --- | --- |
| 小组名称 | 轮椅小队 | |
| 学号 | 姓名 | 本文档中主要承担的工作内容 |
| 20375173 | 陈浩宇 | 文档的编写与完善 |
| 21373260 | 董鹄铭 | 内容讨论 |
| 21371026 | 张飘 | 内容讨论 |
| 20377028 | 龚照徽 | 内容讨论 |
| 21373075 | 应卓航 | 文档审核 |

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| V0.1 | 2024.4.17 | 陈浩宇 | 应卓航 | 完成总体框架 |
| V1.0 | 2024.4.17 | 陈浩宇 | 应卓航 | 更新头控、手控功能 |
| V1.1 | 2024.4.20 | 陈浩宇 | 应卓航 | 更新了异常处理 |
| V1.2 | 2024.4.28 | 陈浩宇 | 应卓航 | 更新了系统结构和系统维护功能 |
| V1.3 | 2024.4.29 | 陈浩宇 | 应卓航 | 更新系统结构和接口定义 |
| V1.4 | 2024.4.29 | 陈浩宇 | 应卓航 | 更新避障和跟踪功能 |
| V1.5 | 2024.5.20 | 陈浩宇 | 应卓航 | 更新服务器端返回异常状态码，更新服务器端异常处理。 |
| V1.6 | 2024.6.9 | 陈浩宇 | 应卓航 | 更新核心控制器的实现 |
| V1.7 | 2024.6.18 | 陈浩宇 | 应卓航 | 更新异常检测与处理 |

目录

[1 范围 1](#_Toc20527)

[1.1 项目概述 1](#_Toc8883)

[1.1.1 项目背景 1](#_Toc11216)

[1.1.2 项目功能及非功能需求 1](#_Toc15921)

[1.1.3 应用场景 2](#_Toc11990)

[1.2 文档概述 2](#_Toc4794)

[1.3 术语和缩略词 2](#_Toc32261)

[1.3.1 术语 2](#_Toc3569)

[1.3.2 缩略词 2](#_Toc25777)

[1.4 引用文档 3](#_Toc6417)

[2. 需求概述 3](#_Toc11649)

[2.1.1 业务需求 3](#_Toc2354)

[2.1.2 数据需求 4](#_Toc25100)

[2.1.3 功能性需求 5](#_Toc12008)

[2.1.4 非功能性需求 6](#_Toc12426)

[3. 数据库设计 6](#_Toc26548)

[3.1 地图信息 6](#_Toc28663)

[4. 体系结构设计 7](#_Toc26652)

[4.1 系统层次设计 8](#_Toc15955)

[4.1.1 服务层 8](#_Toc27496)

[4.1.2 功能层 8](#_Toc20971)

[4.1.3 硬件层 8](#_Toc7760)

[4.1.4 核心控制器 11](#_Toc7621)

[4.2 系统结构设计 12](#_Toc1062)

[5. 接口设计 12](#_Toc10928)

[5.1 总体接口设计 12](#_Toc21055)

[5.1.1 软件外部接口 12](#_Toc20863)

[5.1.2 硬件外部接口 12](#_Toc23458)

[5.1.3 软件内部接口 13](#_Toc10365)

[5.1.4 硬件内部接口 13](#_Toc14160)

[5.2 用户界面接口设计 14](#_Toc17692)

[5.3 功能层接口 17](#_Toc23965)

[5.3.1 IMU惯性测量单元 17](#_Toc15458)

[5.3.2 激光雷达单元 18](#_Toc7532)

[5.3.3 摄像头单元 18](#_Toc24500)

[6. 详细设计 19](#_Toc4184)

[6.1 核心控制器 19](#_Toc24803)

[6.2 头部姿态检测方法 23](#_Toc11532)

[6.3 跟踪模式 25](#_Toc8101)

[6.4 自动避障 27](#_Toc20814)

[6.4.1 位置探测 27](#_Toc13286)

[6.4.2 过程建模 28](#_Toc10646)

[6.4.3 具体实现 29](#_Toc9107)

[6.5 过速预警 30](#_Toc18978)

[6.6 异常检测与处理 31](#_Toc26575)

[6.6.1 自动避障异常 31](#_Toc11753)

[6.6.2 头控模式异常 31](#_Toc17581)

[6.6.3 跟踪模式异常 32](#_Toc12759)

[6.6.4 后端异常 32](#_Toc20457)

[6.6.5 频繁点击按钮异常 33](#_Toc5942)

[7. 系统维护 33](#_Toc9778)

[7.1 系统升级 34](#_Toc11839)

[7.2 恢复出厂设置 34](#_Toc3080)

[8. 运行与开发环境 34](#_Toc4627)

[8.1 运行环境 34](#_Toc24116)

[8.1.1 Raspberry Pi 4B + RPLIDAR S1 + OpenMV4 H7 R2 34](#_Toc27062)

[8.1.2 OpenMV相关操作环境（MicroPython等） 34](#_Toc5996)

[8.1.3 Android 35](#_Toc32202)

[8.2 开发环境 35](#_Toc7069)

[8.2.1 Solidworks 35](#_Toc12560)

[8.2.2 Ros-melodic + Gazebo + RViz 35](#_Toc16105)

[9. 需求可追踪性说明 35](#_Toc6647)

# 1 范围

## 项目概述

### 项目背景

随着人类电子技术、计算机技术、先进制造技术的不断向前推进发展，智能硬件、可穿戴智能设备、人工智能等科技浪潮的兴起，人类的生活向着便捷、舒适、智能的方向被不断改变着，残障人士的生活也不断地被科技发展改变着。使用电动轮椅的人群得到人们的关注，其可靠性、安全性、便捷性都为科学、医疗工作者关心的内容。

目前，新型智能轮椅是国内外研究者的一大热门方向，基于脑电波、肌电接口、语音识别、人脸识别、表情识别等方式的智能轮椅层出不穷，但这大部分轮椅的运动部分都是直接将轮椅进行改装，受限于技术本身的难实现性及可靠性，研发工作量大且成本高昂。从用户角度考虑，其价格往往会过于昂贵、难于操作，可实现性不高。

基于上述一系列思考，我们觉得有必要对市面上几乎所有类型电动轮椅进行研究，设计出一款价格低廉、操作方便、高位截瘫患者能使用的轮椅控制器。

本项目拟采用Slam导航技术、低功耗低成本无线通信技术、3D打印技术、微型嵌入式系统等技术设计几款具有实用意义的智能硬件产品，该产品以市面上90%的电动轮椅为适用对象，可完全、可靠地适配于其上，提供更为友好、便捷的人机交互体验。

### 项目功能及非功能需求

本项目主要功能在于：利用一个头戴式控制装置以及其它配套的操作与探测装置，帮助具有肢体运动障碍，无法使用传统方法控制电动轮椅的残疾人士实现对于轮椅的辅助控制。同时拟开发一个移动端软件，帮助家属实现对于轮椅便捷的远程辅助控制。

除此之外，本项目主要具有以下三个非功能性需求：一是产品的安全性，二是产品的普适性，三是产品相关的性能需求，这一部分将在需求概述部分进行详细涉及。

### 应用场景

本产品计划主要应用于具有肢体运动障碍的残疾人士的日常以及出行场景，残疾人士用户可使用本产品实现对于轮椅的直接操控，从而加强生活的自主能力，避障功能也可以在这一过程中保障用户的人身安全，而家属也可在一同出行时使用手机端进行轮椅操作，比起直接使用固定在轮椅上的操纵杆而言更为便捷。

## 文档概述

本篇文档根据项目当前进展情况，总结并进一步拓展了《需求规格说明书》与《软件开发计划》中项目概述、需求分析、环境说明等部分的内容，并在此基础上给出了项目系统的整体架构，且为架构中的部分重要组件给出了说明，同时为相关组件的接口做出了定义，也为系统结构设计与《需求规格说明书》所分析的需求间建立了联系。

## 术语和缩略词

### 术语

本文档涉及的术语有：

* Ubuntu：一种Linux发行版；
* Python：一种编程语言；
* Rviz：一种机器人可视化平台；
* Gazebo：一种机器人仿真平台。
* Slam：一种同时实现设备自身定位和环境地图构建的技术

### 缩略词

本文档涉及的缩略词及全称有：

表 1 缩略词/全称对应表

|  |  |
| --- | --- |
| 缩略词 | 全称 |
| ROS | Robot Operating System |
| LTS | Long Term Support |

## 引用文档

1. 本项目需求规格说明书，编号：SRS72492，版本：version 1.0.0，发行日期：2024.3.26
2. 本项目软件开发计划，编号：SDP72492，版本：version 1.0.0，发行日期：2024.3.14

# 需求概述

### 业务需求

为具有肢体运动障碍的人群独立操作电动轮椅提供一个成本低、普及性强的辅助操作替代方案，使其能够通过头部装置直接操纵轮椅运动；为该类人群的家属提供一种便捷地帮助亲属操纵轮椅的方案；同时在这一过程中引入障碍识别以及自动避障功能，以保证用户在误操作等情况下的人身安全。

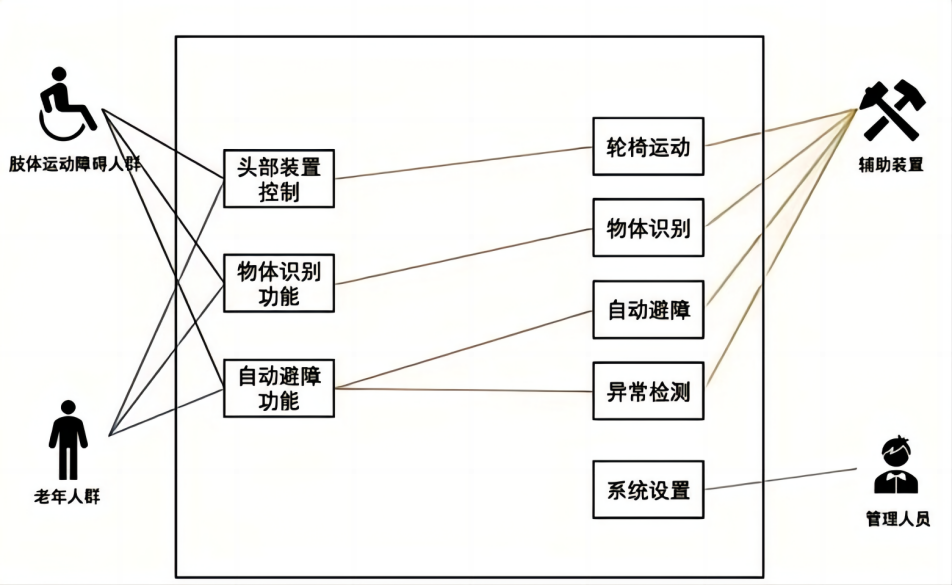


图 1 业务需求图示

### 数据需求

系统应从外部环境采集以下三种数据：一是头戴式控制器的相关姿态数据，这一部分由控制器内置的加速度传感器提供；二是激光雷达测量的障碍物相关数据；三是摄像头采集的图像信号。

同时系统还应从手机客户端接受来自客户的直接指令。

系统应利用激光雷达数据以及摄像头数据计算出障碍物的方位、距离等数据，并结合来自头戴式控制器的姿态指令以及来自手机客户端的直接指令，最终生成出具体的操作指令，将其发送至操纵装置，具体而言，由障碍物数据而生成的避障指令在这一过程中具有最高优先级，其次是客户端的直接指令，最后是来自头戴式控制器的姿态形式的操作指令。

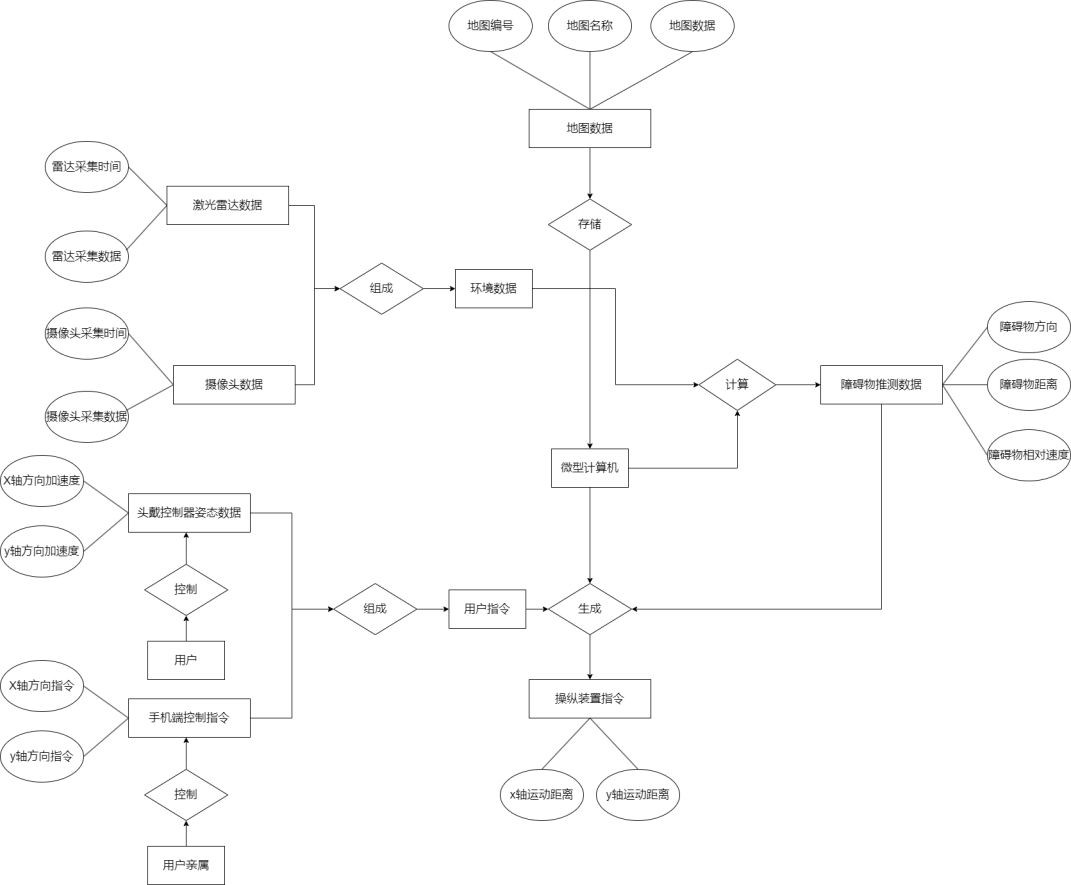


图 2 系统ER图

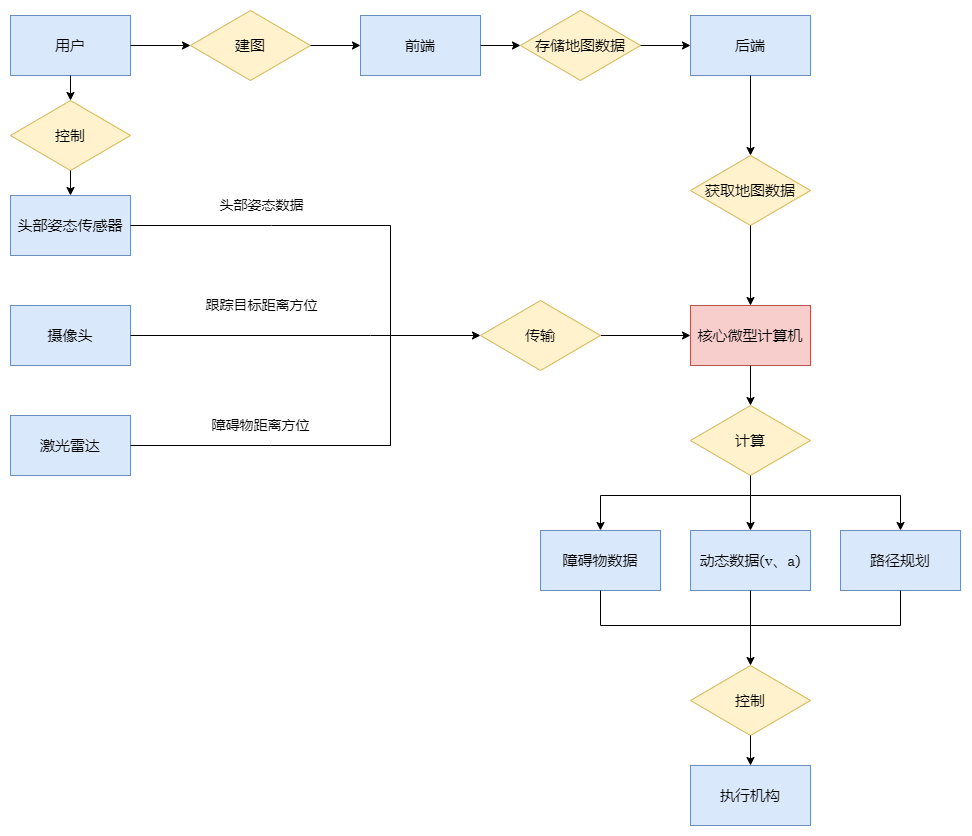


图 3 数据流程图

### 功能性需求

为顺利完成业务需求，项目应满足以下若干个功能需求：激光雷达、摄像头、均可顺利采集环境数据；手机客户端可以顺利产生控制指令，头戴式控制器可以按照预期返回以头部姿态形式呈现的控制指令；树莓派应可以根据激光雷达和摄像头数据推断出障碍物数据，并且应能使用障碍物数据和用户指令推断出最终指令；轮椅直接控制装置可以根据最终指令对应操控轮椅操纵杆；各个组件应与树莓派之间均可实现正常的通信。

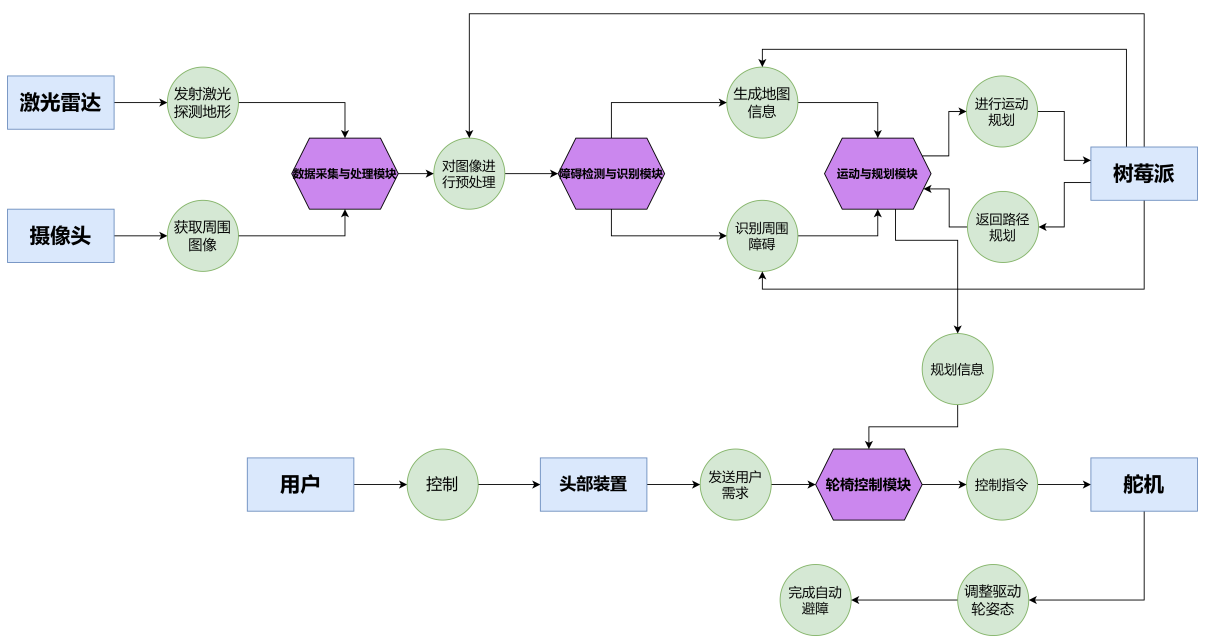


图 4 功能需求图示

### 非功能性需求

为满足项目要求，项目设计中应注意以下三个非功能需求：一是产品的**安全性**，本项目拟使用激光雷达为主要的障碍物探测手段，同时使用摄像头进行辅助探测，从而实现电动轮椅的紧急避障功能，避免使用者误操作可能带来的潜在风险；二是产品的**普适性**，本产品使用一个由电机控制的操作装置实现系统对于电动轮椅的直接控制，该装置可直接安装于轮椅操纵杆上，其应进行简单调整便能适配不同型号的常见电动轮椅；三是相关的**性能**需求，产品应能对用户的指令以及紧急避障指令进行快速的响应，将最大响应用时控制在一定范围之内，并且产品应能保持一定的续航能力，从而满足产品用户的外出需求。

# 数据库设计

## 地图信息

地图信息主要包括地图名称、地图文件路径、标注文件路径。每个地图都有一个唯一的名称，用String表示。为了服务于不同的地图场景，应对标记地点变化，每一个地图还存储了对应的地图文件路径和对应的标注文件路径，给用户提供地图选择和标注的服务。

表 2 地图信息数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 能否为空 | 自动递增 | 备注 |
| Map\_name | String | NULL | 否 | 否 | 地图名称 |
| Map\_sto\_path | String | 根目录 | 否 | 否 | 地图存储路径 |
| Map\_anno\_path | String | 根目录 | 是 | 否 | 地图标注路径 |
| Map\_pic\_path | String | 根目录 | 否 | 否 | 地图图片路径 |

# 体系结构设计

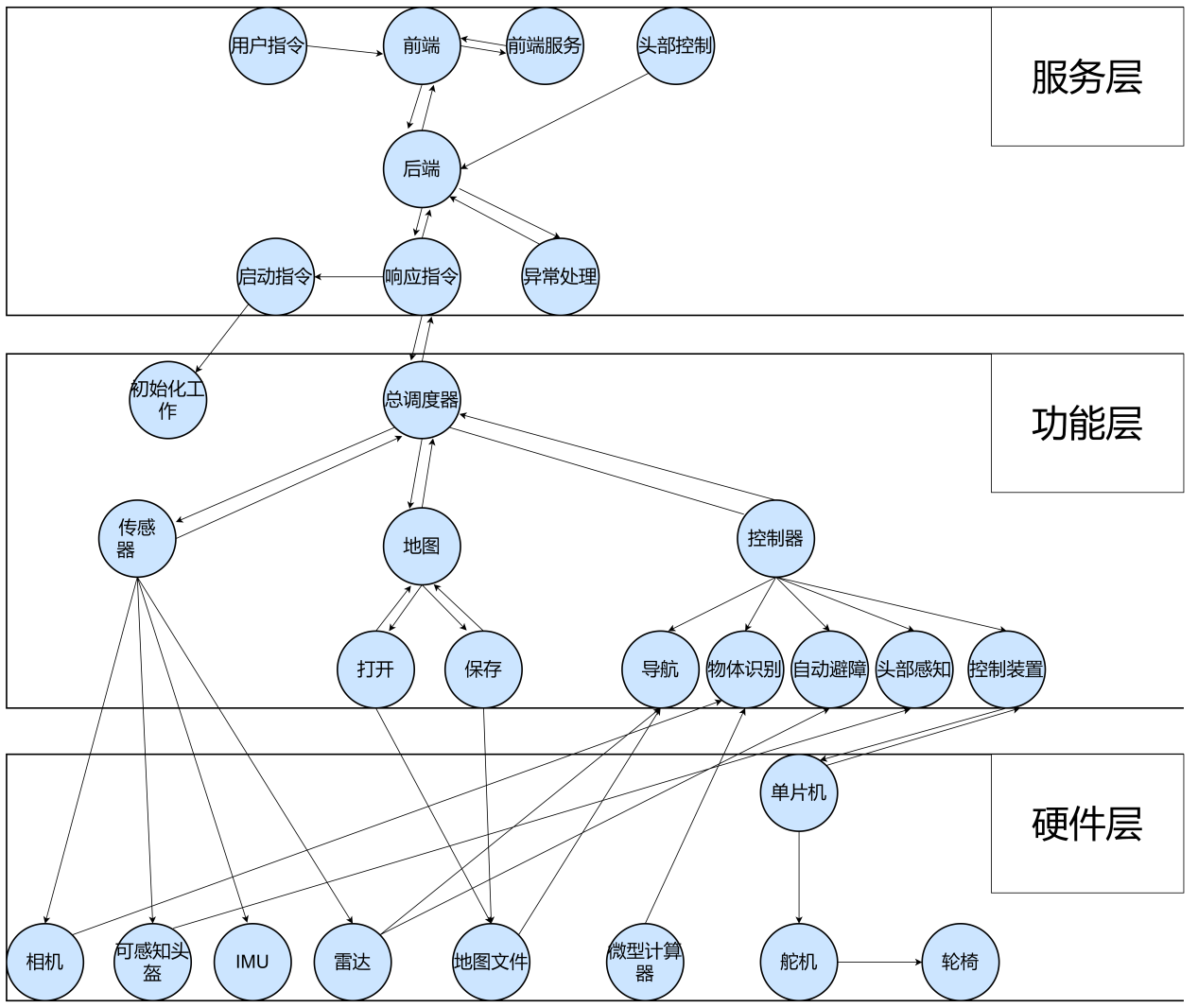


图 5 系统体系结构

项目总体结构包括三个层次，服务层、功能层、硬件层。

## 系统层次设计

### 服务层

在服务层，用户通过用户界面（前端）与系统进行交互，一方面能够通过手控模式对轮椅运动进行控制，控制轮椅的前后左右运动，同时能够选择进入脑控模式，轮椅乘坐者即可通过头部运动对轮椅进行控制。另外设有导航模式，建图之后即可通过地图进行目的地的导航，根据实际环境选择地图并做标注。同时界面还设有跟踪模式，可以控制轮椅跟随用户进行自动移动。

### 功能层

在功能层，总调度单元会根据用户界面发送过来的服务请求，启动对应的服务。并接收传感器模块的反馈信息，如轮椅使用者的头部微动信息和跟踪方位信息，并由此向总控制模块发送轮椅的控制信号。

在功能层，总控制模块主要根据调度单元分配的任务向不同的输出单元发送子任务参数，如向运动控制单元（舵机）发送轮椅移动参数、保存和打开地图文件等。传感器模块负责处理不同输入单元发送的信号，如激光雷达信号、OpenMV摄像头信号和头部姿态摆动方位信号等，然后向总调度单元发送处理结果如输入的控制指令、头部摆动方位和识别到的物体等消息。地图模块用于向调度器返回地图数据或者保存前端传来的地图数据。

### 硬件层

在硬件层，不同的功能单元与硬件层紧密结合，负责控制硬件完成最基本的任务，如控制舵机、跟踪用户、检测头部微动信息、自动避障、建图等。

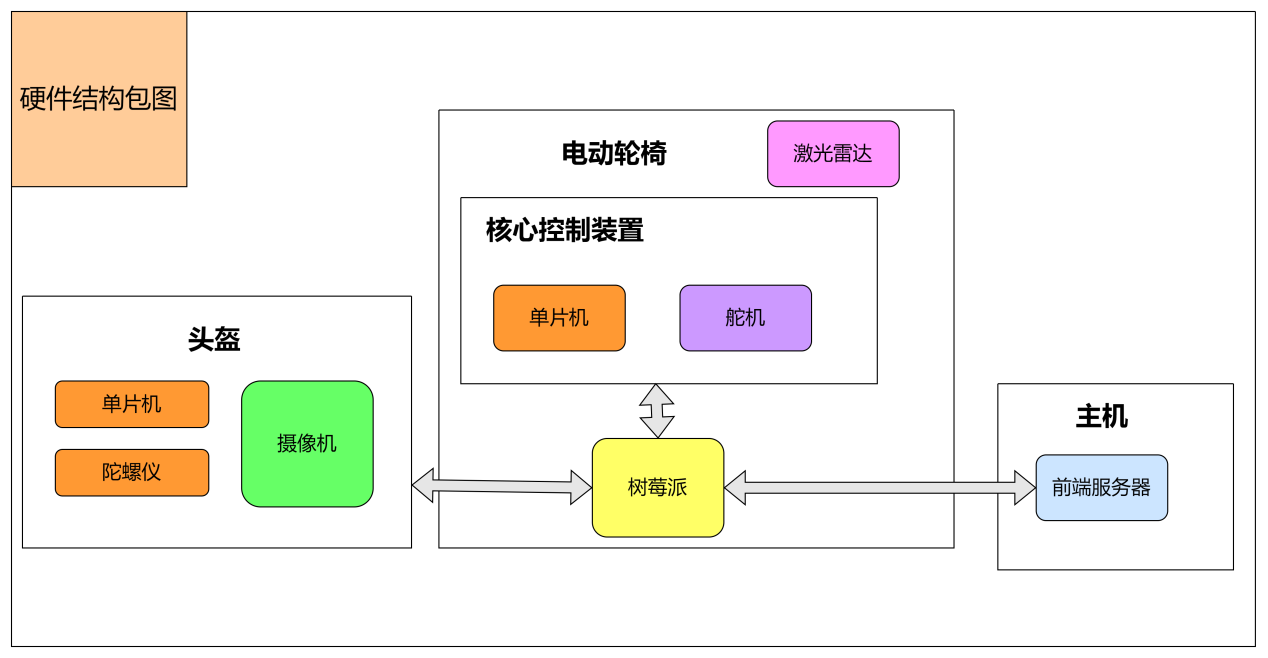


图 6 硬件架构包图

OpenMV摄像头负责跟踪任务，识别前方要跟踪的目标物体AprilTag，并根据物体的距离变化和方位变化生成响应的跟踪运动指令，总调度器收到后进而驱动控制模块队轮椅运动进行控制。

激光雷达模块用于避障和建图任务，当识别到前方物体距离过近时，会通知总调度器与前方障碍物距离过近，调度器会根据障碍物的距离和方位信息产生相应的避障指令，如通过左转或右转绕开障碍物前进。

IMU（惯性测量单元）是一种传感器系统，集成了多个传感器，用于测量物体的运动和姿态。IMU通常包括加速度计、陀螺仪、磁力计等传感器。本项目所使用的IMU为十轴IMU，是基于MEMS技术的高性能三维运动姿态测量系统，包含三轴陀螺仪、三轴加速度计，三轴电子罗盘运动传感器和气压计。通过集成各种高性能传感器和运用自主研发的姿态动力学核心算法引擎，结合高动态卡尔曼滤波融合算法，实现了高精度、高动态、实时补偿的三轴姿态角度估测，可以适应不同的应用场景。

卡尔曼滤波（Kalman Filter）在惯性测量单元（IMU）的数据处理和融合中起着关键作用。卡尔曼滤波是一种递归算法，能够在动态系统中进行最优估计，减少噪声和误差，提高姿态和位置的精度。卡尔曼滤波基于两个主要步骤：预测和更新。预测过程主要是利用系统的运动模型（通常是物体的运动方程）预测下一时刻的状态。预测当前状态和协方差矩阵，表示系统在没有新测量数据时的估计。更新过程主要是利用新的测量数据更新状态估计，然后计算卡尔曼增益（Kalman Gain），用于平衡预测值和测量值的权重。最后更新状态估计和协方差矩阵，综合考虑预测和测量的不确定性，得到更精确的估计。

图 7 IMU内部结构图

① IMU模块的核心模组：所有计算位姿都通过此核心模组完成。

② 外接引脚：可连接GPS模块或者I2C设备。

③ 串口引脚：输出TTL信号，可与STM32等单片机通讯。

④ CP2102芯片：USB转串口功能。

⑤ D2指示灯：模块状态指示灯，模块正常情况下常亮。

⑥ D1指示灯：CP2102状态指示灯，USB在通讯时点亮，空闲时熄灭。

⑦ Type-C接口：连接到模块的串口，做为通讯接口使用。

⑧ 固定铜柱：用于安装金属外壳或者安装GPS模块。

表 3 引脚功能说明

|  |  |
| --- | --- |
| **NC** | **空** |
| D1 | 连接GPS模块的数据接口 |
| GND | 接地 |
| 5V | 5V |
| GND | 接地 |
| SDA | I2C-串行数据线 |
| SCL | I2C-串行时钟线 |
| 3V3 | 3.3V |
| RX | 串口-数据接收引脚 |
| TX | 串口-数据发送引脚 |
| GND | 接地 |
| 5V | 5V |

### 核心控制器

电动轮椅上的控制器是本项目中最重要的部分，设计为二自由度无极解耦控制结构——将轮椅控制器摇杆的类球面运动解耦为二自由度的无级运动，提高控制的可靠性。它驱动两个舵机，一个负责直线运动，另一个负责旋转运动，二者相互配合共同完成轮椅的实际运动控制。在此给出它的详细架构：

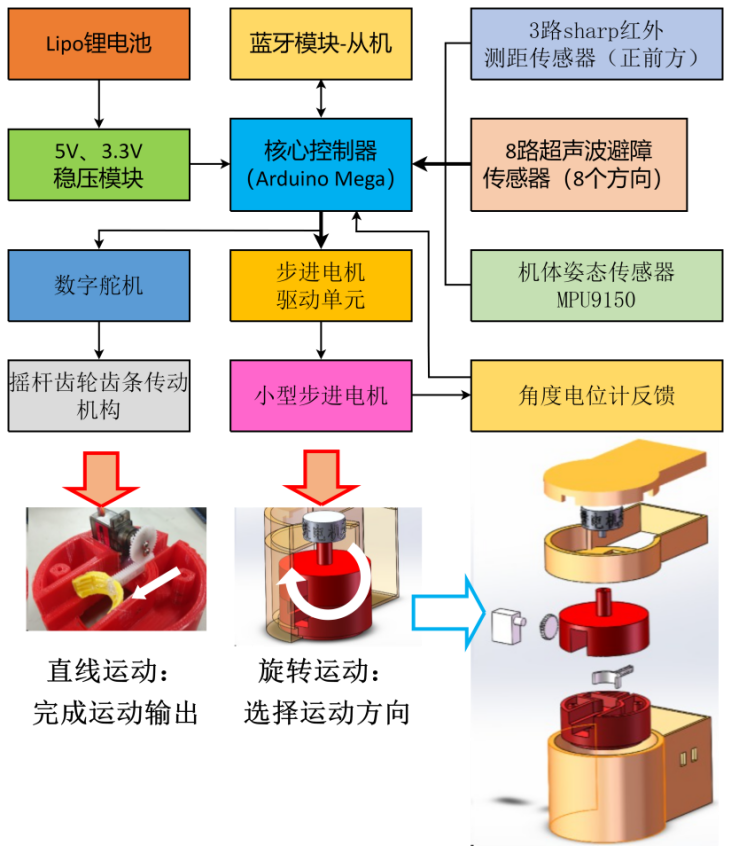


图 8 核心控制器

## 系统结构设计

将系统层次中的主要模块进行封装，即可得到系统结构图。

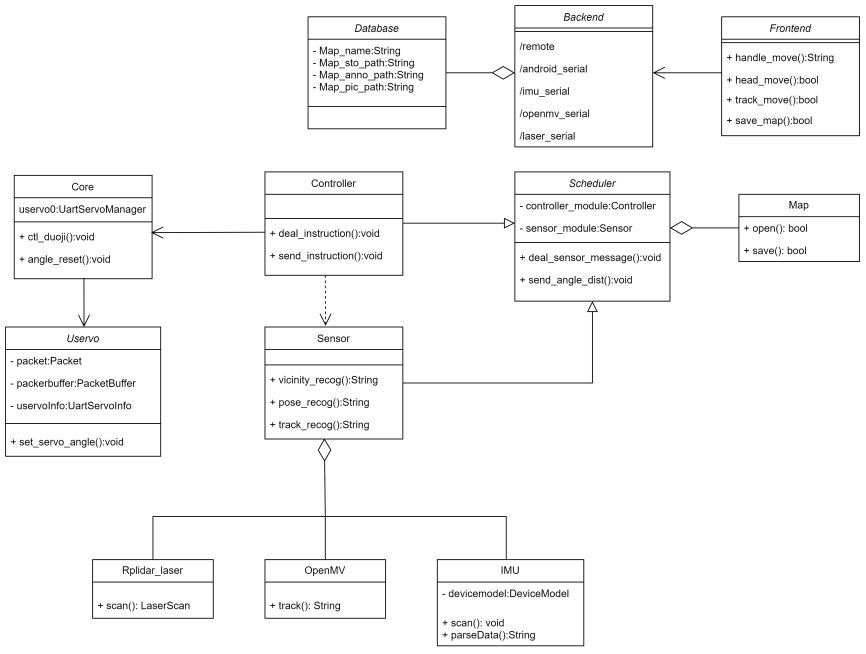


图 9 系统结构类图

从UML类图的角度，重新刻画了系统的结构，同时给出了各个模块的接口，对于接口的具体设计将在下一节说明。

# 接口设计

## 总体接口设计

### 软件外部接口

头部姿态检测接口：通过惯性测量单元获取头部姿态数据，并使用卡尔曼滤波方法获得准确的欧拉角。

环境感知接口：使用激光雷达模块、摄像头模块块获取环境信息，实现避障和跟踪功能。

### 硬件外部接口

轮椅控制接口：轮椅辅助控制单元与轮椅原有的控制器连接，通过操纵杆插入延伸控制机构，使用二自由度伺服系统控制轮椅的运动。

通信接口：环境感知单元采用低功耗低成本蓝牙无线通信技术和Wifi与其他设备进行通信，传输环境感知数据。

### 软件内部接口

头部姿态检测模块与轮椅辅助控制模块之间的接口：建立头部姿态与主体运动意图之间的映射关系，将头部姿态数据通过蓝牙传递给轮椅辅助控制模块，实现对轮椅的控制。

环境感知模块与轮椅辅助控制模块之间的接口：将环境感知数据通过树莓派传递给轮椅辅助控制模块，用于实现避障和跟踪功能。

### 硬件内部接口

头部运动传感单元内部接口：惯性测量模块收集头部姿态数据并传递给卡尔曼滤波方法进行处理，获取准确的欧拉角数据。

轮椅辅助控制单元内部接口：接收头部姿态数据和环境感知数据，根据设定的控制策略进行轮椅运动控制。

环境感知单元内部接口：各个传感器模块之间相互传递数据，将感知到的环境数据整合后传递给轮椅辅助控制单元进行处理。

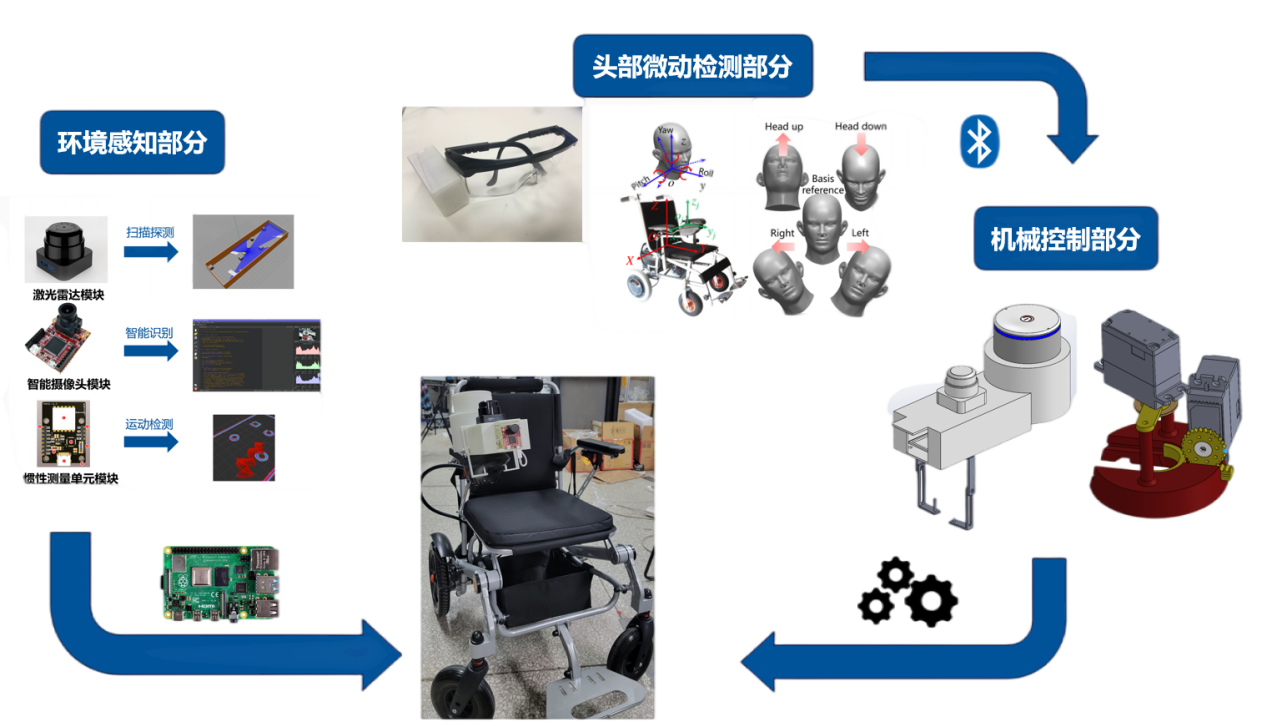


图 10 总体接口设计图

## 用户界面接口设计

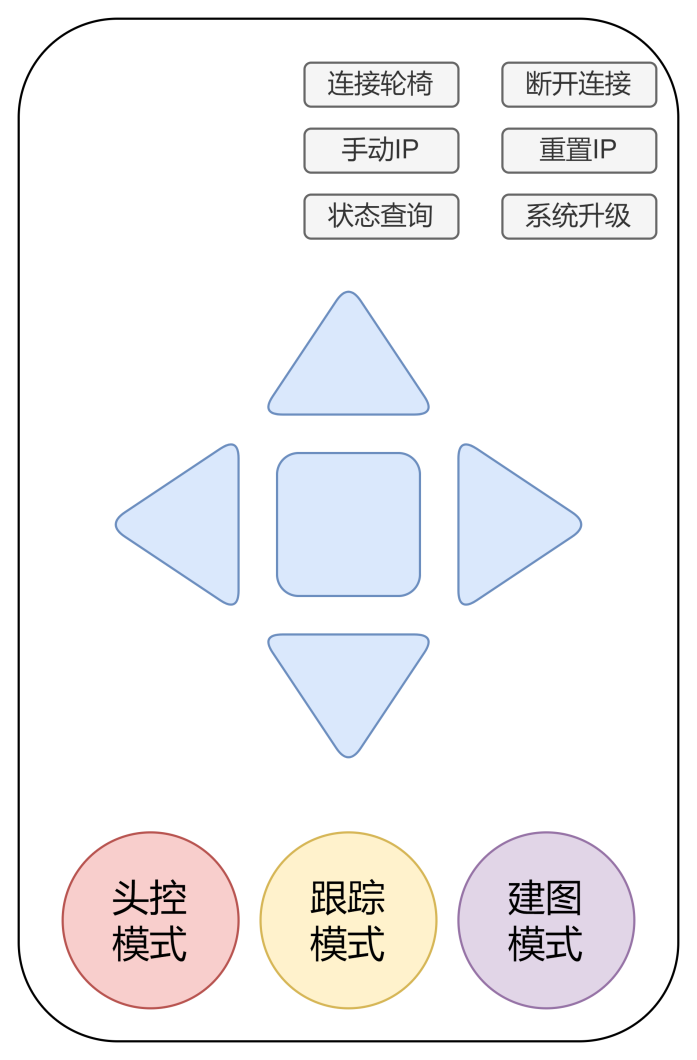
****

图 11 Android App设计界面图

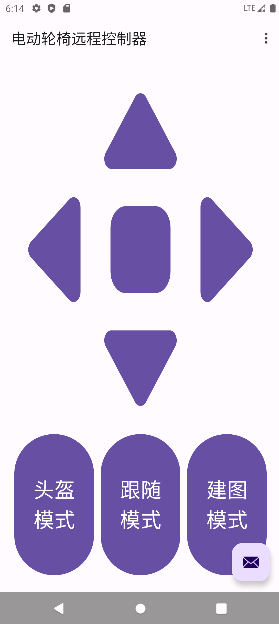
 

图 12 Android App实际界面图

用户主界面上主要显示用户可以执行的四个功能，手控模式、头控模式、跟踪模式和建图模式。顶部右上角有一个设置按钮，有连接轮椅、断开连接、手动配置轮椅IP、重置轮椅IP和检查当前连接状态共5个选项功能。



前后端通过Rosbridge进行通信，Rosbridge是一个允许在ROS系统中进行与非ROS客户端通信的桥梁。它基于WebSockets和JSON进行消息传递。Android端启动一个话题/remote，然后在这个话题下发布消息。ROS端订阅该话题，收到消息后进行相应的处理。

表 3 手控模式接口定义

|  |  |
| --- | --- |
| 手控模式 | |
| 通信话题 | /remote |
| 参数 | String：12（前进），14（左转），16（右转），18（后退） |
| 操作 | 1表示当前为手控模式，2、4、6、8分别表示前进、左转、右转、后退 |
| 返回 | msg，成功时code为200，失败时返回对应错误码400。 |

表 4 头控模式接口定义

|  |  |
| --- | --- |
| 头控模式 | |
| 通信话题 | /remote |
| 参数 | String：2 |
| 操作 | 2表示进入轮椅乘坐者头椌模式 |
| 返回 | msg，成功时code为200，失败时返回对应错误码401。 |

表 5 跟踪模式接口定义

|  |  |
| --- | --- |
| 跟踪模式 | |
| 通信话题 | /remote |
| 参数 | String：3 |
| 操作 | 3表示进入跟踪模式，开始处理OpenMV摄像头的跟踪消息 |
| 返回 | msg，成功时code为200，失败时返回对应错误码402。 |

表 6建图模式接口定义

|  |  |
| --- | --- |
| 建图模式 | |
| 通信话题 | /remote |
| 参数 | String：4 |
| 操作 | 4表示调用建图脚本，启动机器人建图服务。 |
| 返回 | msg，成功时code为200，失败时返回对应错误码403。 |

表 7 系统升级接口定义

|  |  |
| --- | --- |
| 系统升级 | |
| 通信话题 | /remote |
| 参数 | String：5 |
| 操作 | 后端调用脚本，完成更新操作。 |
| 返回 | msg，成功时code为200，已经是系统最新版本则返回300，失败时返回对应错误码404。同时会给出相应提示。 |

## 功能层接口

功能层接口与硬件结合紧密，控制模块下的功能单元完成控制模块下达的指令，进行移动、跟踪、避障等功能，由于大部分节点都能够直接发布具体的控制指令，不需要功能单元在单独发送，因此控制模块下的功能单元不做接口设计。传感器模块下的功能单元主要负责收集传感器信息并发送给传感器模块集中处理。

### IMU惯性测量单元

IMU模块被集成到了imu包中，该包与IMU进行蓝牙连接，当接收到数据时通过回调函数对数据进行解析然后将指令消息发布到/imu\_serial话题。

表 8 IMU接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| scan | 无 | /imu\_serial | 与IMU进行蓝牙连接，根据测量到的角度生成对应的运动指令 |

### 激光雷达单元

雷达单元已经集成在了/laser\_serial这个包中。雷达可以360°扫描轮椅四周进行建图，同时也可以根据前方障碍物的距离进行避障。

表 9雷达单元接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| yahboomcar\_laser | /scan  /velocity | /laser\_serial | 对雷达探测到的信息和轮椅当前速度进行解析，然后发布避障指令 |
| rplidar\_ros | 无 | /scan | 与激光雷达设备进行连接，分析前方探测到的障碍物的距离和方位 |

### 摄像头单元

摄像头使用OpenMV，它是一个可编程的摄像头，通过MicroPython语言，可以简便高效的实现一些图像处理的算法。

表 10 摄像头接口定义表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 订阅 | 发布 | 说明 |
| Tracker | 无 | /openmv\_serial | 封装了摄像头信号相关的对于跟踪AprilTag的处理细节，根据方位和距离生成相应的控制指令 |

# 详细设计

在前文架构总体设计的基础上，我们借助已经写好的各种信息处理节点和控制节点对各个任务进行了更细致的设计。在各层通信上并没有采用层次化的通信方式，而是根据ROS独有的通信机制，最终在具有层次的架构设计上实现了更灵活的功能实现。

## 核心控制器

根据图13与图14所示，为本装置的整体结构示意图，它清晰地描绘了整个系统的基本布局。本装置的机械结构设计精妙且全面，旨在实现电动轮椅的自动化控制。整个系统以固定元件为基础，承载了所有关键组件，确保了系统的稳定性和可靠性。单目视觉传感器和激光雷达传感器被安装在固定元件之上，它们如同系统的“眼睛”，用于实时感知周围环境，为系统决策提供关键信息。这些信息通过数据线传输到树莓派计算平台，该平台作为系统的“大脑”，负责处理这些感知数据，并发出相应的控制指令。

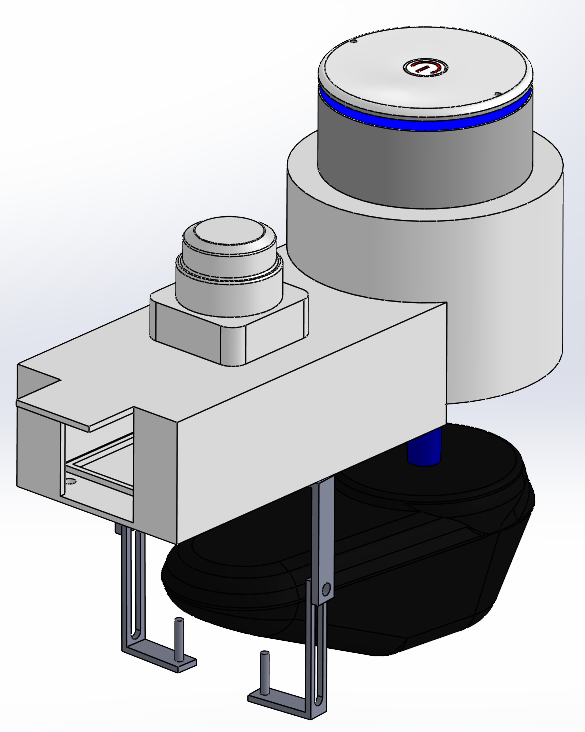


图 13整体结构图

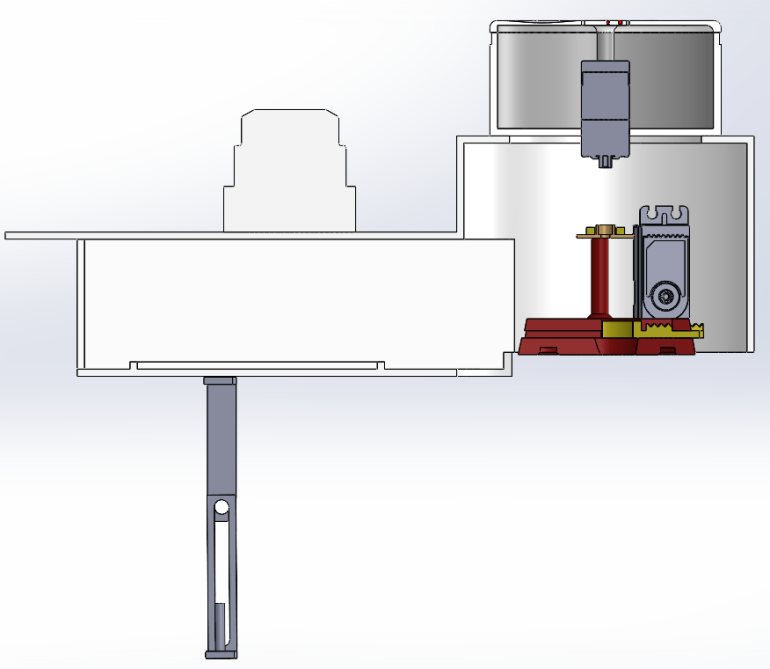


图 14结构剖面图

如图15所示的多自由度推杆机构是本装置的核心部件，它实现了机械传动的关键功能。这个机构按照功能可以分为两个部分：行进方向旋转机构（图15上半部分）和运动速度控制机构（图15下半部分）。

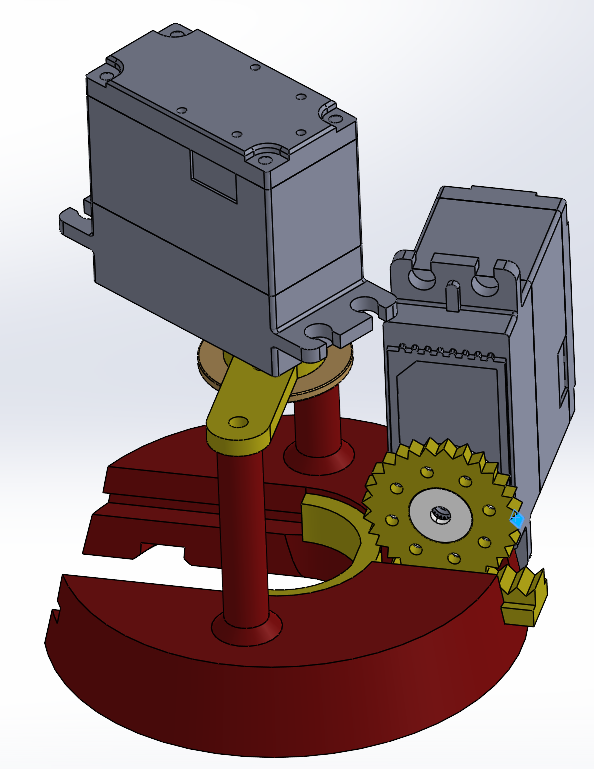


图 15 多自由度推杆机构

图16为转轴的结构示意图，图17为多自由度推杆机构的3D结构爆炸视图。行进方向旋转机构通过旋转舵机的驱动，实现了轮椅方向的调整。旋转舵机与舵机舵盘相连，舵盘再通过舵机连接件与旋转舵机安装孔固定，最终带动转轴转动。当舵机接收到来自树莓派计算平台的指令后，它会驱动舵盘旋转，从而改变轮椅的行进方向。

运动速度控制机构则通过推进舵机实现。推进舵机通过推进齿轮与推杆相连，当推进舵机接收到速度控制指令时，它会驱动推进齿轮转动，进而推动推杆前后移动。推杆前端设计为弧形板，与电动轮椅的控制器摇杆接触，将推杆的移动转化为电动轮椅的运动速度变化，实现了速度的精确控制。

此外，推杆安装槽和摇杆安装孔等细节设计也体现了机械结构的精细和巧妙。推杆安装槽用于固定推杆，防止其在工作过程中发生偏移；摇杆安装孔则用于安装摇杆，使得用户可以手动干预或调整轮椅的运动状态。

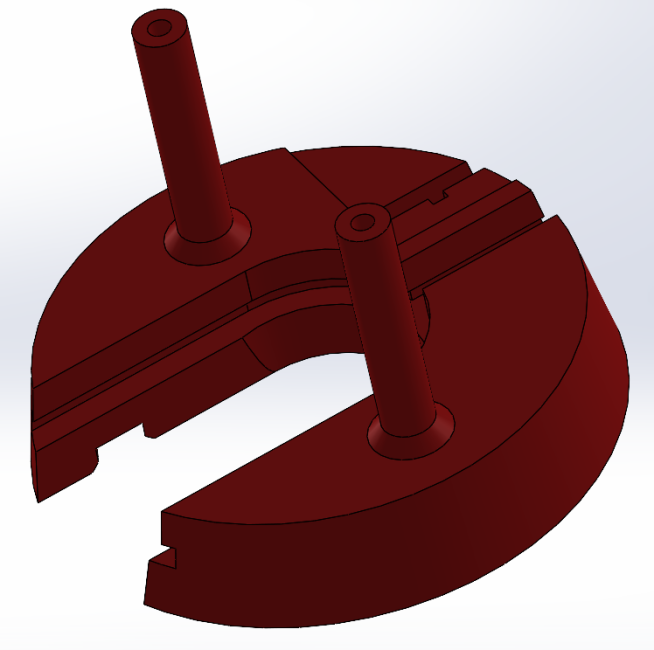


图 16 转轴结构示意图

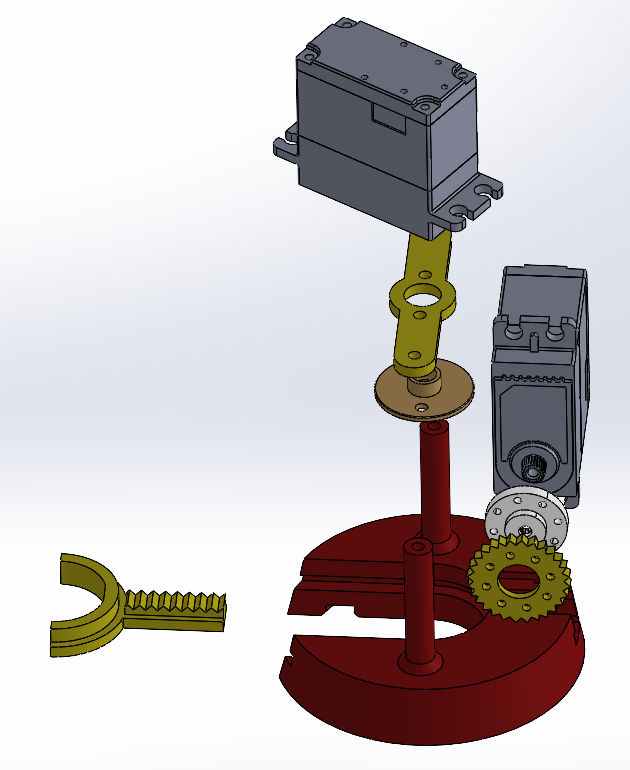


图 17 多自由度推杆3D结构爆炸视图

整个装置与电动轮椅的连接设计也十分独特，图18所示为本装置与电动轮椅控制器的连接示意图。本装置完全独立于电动轮椅，可使用固定元件安装于大多数包含遥控器的电动轮椅。固定装置可以在一定范围内自由变换夹紧的宽度及高度，适用于各种型号的电动轮椅遥控器。这种设计不仅方便了装置的安装和拆卸，也提高了装置的通用性和实用性。

总的来说，本装置的机械结构设计充分考虑了电动轮椅的自动化控制需求，通过巧妙的机构设计和精细的部件配合，实现了对电动轮椅行进方向和运动速度的精确控制。同时，装置还具有良好的通用性和实用性，可适用于多种型号的电动轮椅。

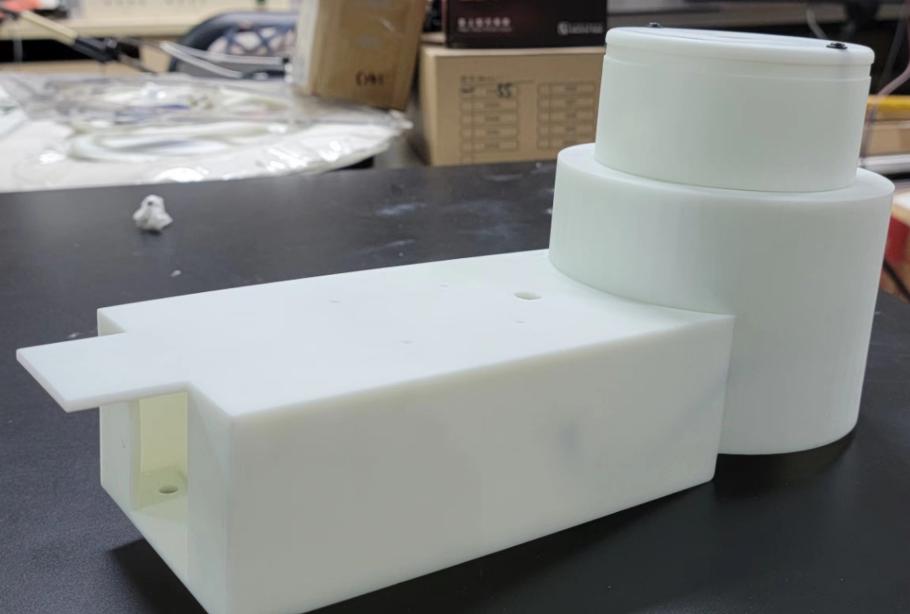


图 18 装置连接示意图

## 头部姿态检测方法

由于常用的IMU在测量角度等数据时存在测量误差和噪声，测量得到的横滚角和俯仰角需要使用滤波器进行处理以获得更好的测量结果。常用的滤波器有互补滤波器和卡尔曼滤波器。

互补滤波的基本思想是对不同传感器的数据进行加权并平均，权重可以根据每个传感器的特性和测量噪声自适应调整。对于IMU测量得到的姿态角修正，互补滤波通常分别处理加速度计和陀螺仪数据并对其进行加权平均，以获得姿态估计结果。

卡尔曼滤波的主要思想是将系统的状态表示为向量，并在每一步更新状态向量的估计。虽然卡尔曼滤波器的计算量比互补滤波器要大得多，实时性稍差，但其鲁棒性和精度却比互补滤波器要好。因此，本系统在滤波算法中采用卡尔曼滤波处理。系统的状态空间方程如下。

其中 **X**k−1是系统状态向量，**A**是系统状态 **X**k−1对系统状态 **X**k影响的状态转移矩阵。**B**是控制向量 **U**到状态向量 **X**的输入矩阵。**H**是观测矩阵，**Z**是系统输出向量。ωk−1是过程噪声，通常假设为零均值高斯白噪声，其协方差矩阵由以下方程 **P**(ω) ~ **N**(0, **Q**k ) 表示。**V**k是测量噪声，再次假设为零均值高斯白噪声，其协方差矩阵由以下等式 **P**(V) ~ **N**(0, **R**k ) 给出。状态方程中的过程噪声 ωk−1和测量噪声 **V**k是不可确定的，它们代表了状态转移过程和观测过程中的不确定性和随机扰动。

卡尔曼滤波过程共5步，经过多次迭代，得到的可以视为用户头部的姿态欧拉角（俯仰角和横滚角），如图所示。

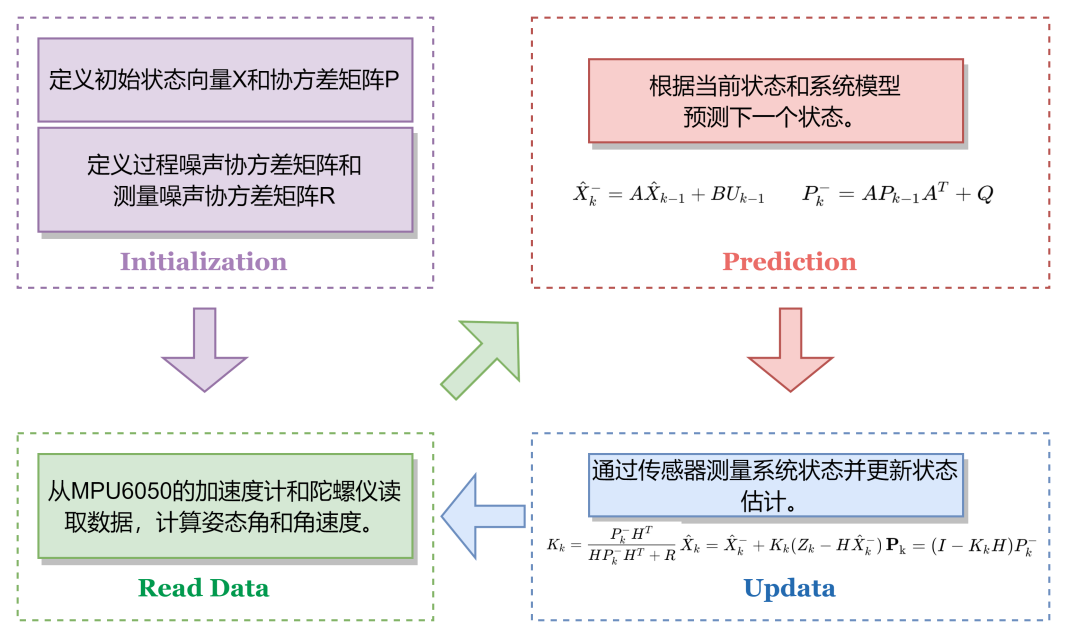


图 19卡尔曼滤波流程图

头部感知模块首先初始化DeviceModel实例，传入设备名称、MAC地址和回调函数。然后调用openDevice()方法与IMU（集成了卡尔曼滤波器的十轴IMU）开启并建立连接。然后启动蓝牙设备的通知函数client.start\_notify()，用于接受来自IMU的头部姿态检测数据，然后通过onDataReceived()回调函数将头部姿态数据进行解析，生成相应的运动指令后发布到话题imu\_serial下，调度器模块收到话题下的消息后指挥控制器调度轮椅的运动。

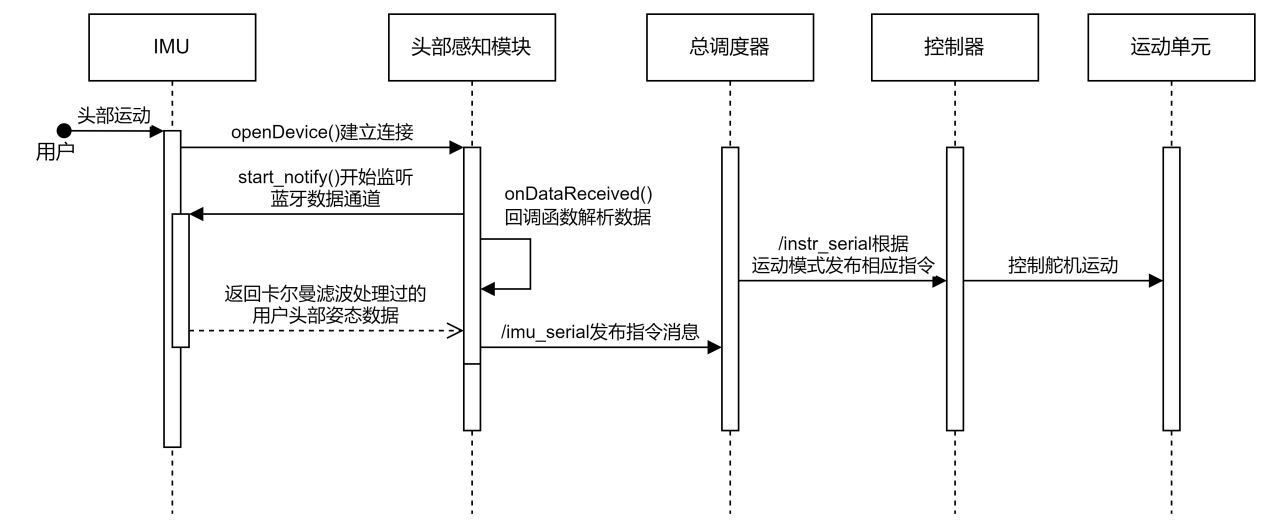


图 20头控模式时序图

## 跟踪模式

跟踪模式主要依赖于OpenMV摄像头识别出探测范围内的AprilTag并对其位置进行定位来实现。AprilTag是一种视觉基准系统，可用于帮助摄像头实现3D定位，其相较于二维码对抗图像畸变的能力更强，易于被摄像头识别。在光线状态理想时，AprilTag的识别范围约为0.3m~2.8m，轮椅会根据标志的具体距离与角度进行前进、后退、转弯等运动。另外摄像头可在一定的光线范围内自动调整曝光时间，优化识别效果。

首先摄像头检测前方视野中的Tag，若检测到则计算其在三维空间中的位置和旋转角度。具体方法为首先通过MicroPython的Image库中封装好的函数img.find\_apriltags()获得当前帧图像中捕捉到的AprilTag，该类中包含了当前Tag在x、y、z三个轴上的坐标和旋转角度。然后根据这六个坐标解析出当前轮椅应该运动的方向，核心代码如下。



实现逻辑如下：

如果Tag当前在x轴上偏离坐标原点，说明需要转动来跟踪目标，即若偏左则左转，偏右则右转；如果direction不为0且actual\_x的绝对值小于end\_d，说明已经接近中线，设置direction为0（停止左右移动）。

如果Tag当前在z轴（即前后距离）上的坐标发生变化，需要进行前后的移动。如果actual\_z的绝对值大于start\_f，说明标签距离较远，需要前进，设置moving为1（前进）。如果actual\_z的绝对值小于start\_b，说明标签距离太近，需要后退，设置moving为-1（后退）。如果当前在前进状态且actual\_z的绝对值小于end\_m，说明已经接近目标位置，停止前进，设置moving为0。如果当前在后退状态且actual\_z的绝对值大于end\_m，说明已经远离目标位置，停止后退，设置moving为0。

最后根据解析得到的direction和moving生成对应的运动指令即可，如果direction和moving均为0，说明需要停止，设置command为5（停止）。如果direction不为0，根据direction的值设置左右转向命令：direction为-1时设置command为4（向左转），为1时设置command为6（向右转）。如果moving不为0，根据moving的值设置前进或后退命令：moving为1时设置command为2（前进），为-1时设置command为8（后退）。

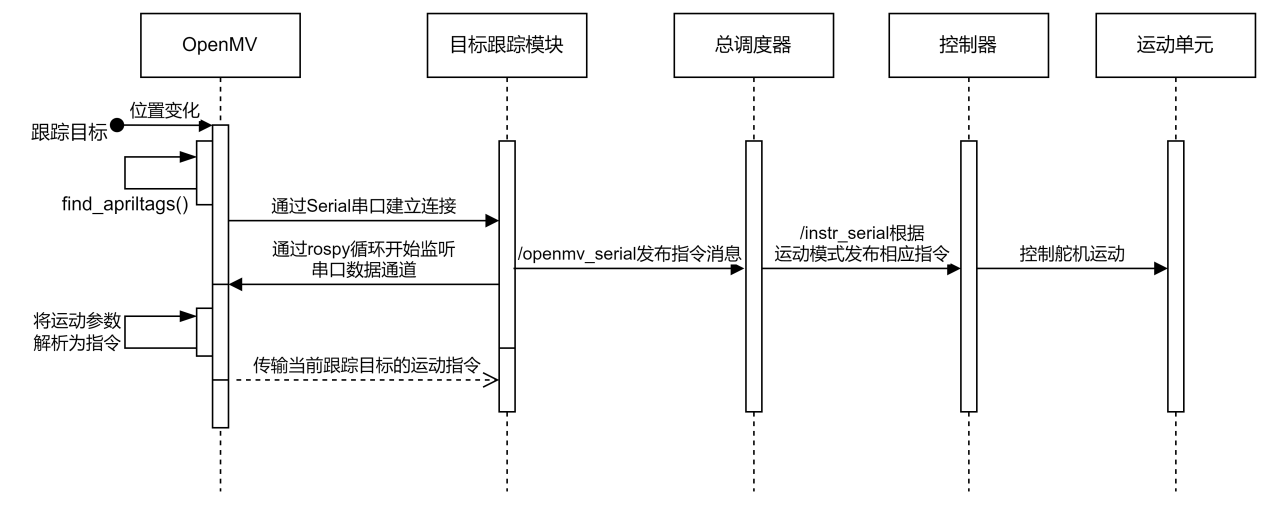


图 21跟踪模式时序图

## 自动避障

### 位置探测

使用2D激光雷达实现前方120度内的障碍探测，通过轮椅与障碍物间的两方向距离判断是否有碰撞风险，从而下达避障指令。基于地图信息和实时捕获的环境信息，在轮椅运动过程中可以实时检测运行，并不断向轮椅控制系统发出调整指令。

通过雷达传回的反射点距离与对应角度计算轮椅距离障碍物在X轴与Y轴方向上的距离，仅当某个反射点与轮椅在Y轴的投影重合并且在X轴上的距离小于一定阈值时，才将该反射点纳入到避障计算之中。同时过滤了极小距离内的反射点以及探测角度外的反射点的相关数据，防止将客户或装置本身误判为障碍。

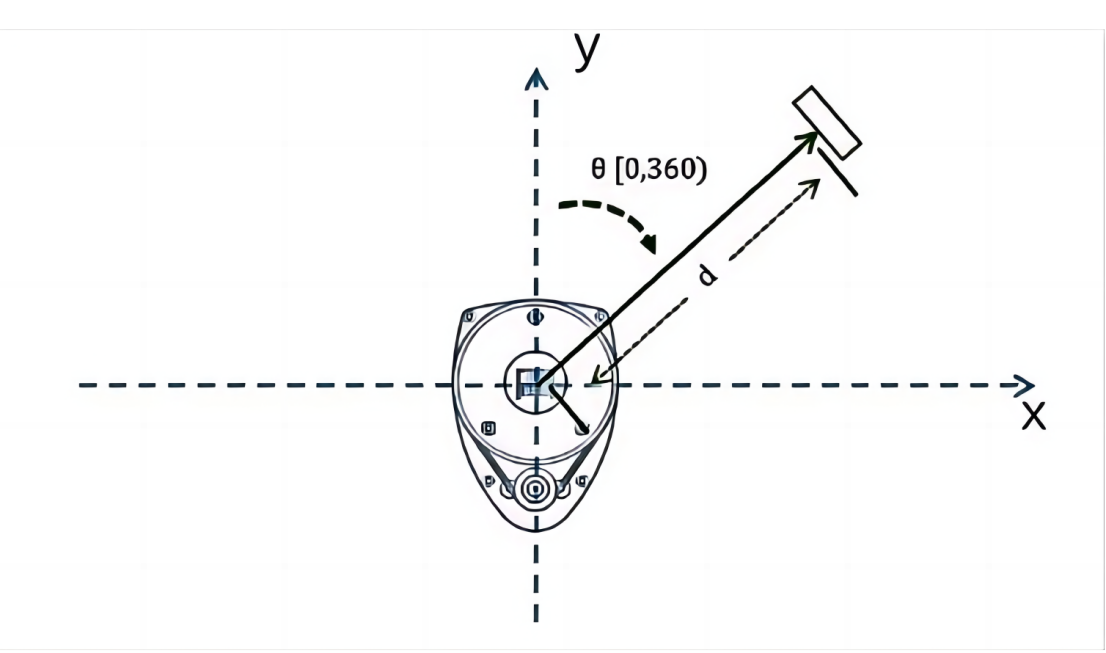


图 22雷达障碍探测示意图

### 过程建模

为了最大程度的保证轮椅乘坐者的安全，我们精确控制轮椅从雷达探测到障碍物到轮椅停下来的制动距离，建模过程如下。

通过实验测得，轮椅的后退推杆的推动距离与轮椅的加速度大致存在线性关系，即

其中，是轮椅后退过程的最大减速度，是推杆的最大推动距离。通过实验测得约为，推杆的最大推动距离测得约为5cm。

雷达测得的为轮椅到障碍物的直线距离和对应的角度，我们需要将直线距离d分解为轮椅前进方向（即X轴方向）和垂直于前进方向（即Y轴方向）的分量。

为了确保轮椅能够在接近障碍物时减速至0，我们只需要考虑前进方向上的距离分量。因为避障主要是为了避免前方碰撞，所以可以忽略垂直方向的分量。

为了保证轮椅能够在碰到障碍物之前减速至0，我们需要使用基本的运动学方程，即

其中，v是最终速度，为0；u是初始速度，即当前速度；s是距离障碍物。

考虑到此时速度减为0时，最坏的情况下距离障碍物的距离为0，这显然是不允许的，因此添加的一个额外的推动距离，保证速度减为0时距离障碍物仍存在一段距离，本项目设置=20cm，即

通过以上公式可得，

制动后可保证速度减为0时距离障碍物仍有20cm的距离，更加安全。根据轮椅当前速度、最大推杆距离和最大加速度来决定当前轮椅的避障过程的推杆距离，从而实现精准避障。

### 具体实现

该模块通过两个包来实现避障功能，rplidar\_ros负责与激光雷达设备连接，并对前方障碍物进行处理解析为对应的距离和角度信息，通过/scan话题发布；yahboomcar\_laser负责订阅/scan话题发布的消息和/velocity话题发布的轮椅当前速度的消息，然后对消息进行解析生成对应的控制指令。

接收到激光雷达发来的数据后，需检查数据类型是否为LaserScan。然后将激光雷达的范围数据转换为Numpy数组，初始化前方预警计数器。遍历激光雷达数据，根据角度和距离计算前方和侧方距离。根据侧方距离判断是否存在碰撞风险，并记录预警信息。如果存在多个预警数据，找出最小距离及其对应角度，并重新计算前方和侧方距离。如果前方预警计数器超过预设阈值，发布后退指令“8”和对应的推杆推动距离，否则发布保持不变指令（“5”）。**由于装置安装在轮椅的右侧扶手上，乘坐者离装置左侧较近，所以避障范围应该将左侧阈值角度适当减小，避免激光雷达误将乘坐者视为障碍物从而产生错误避障信息。**

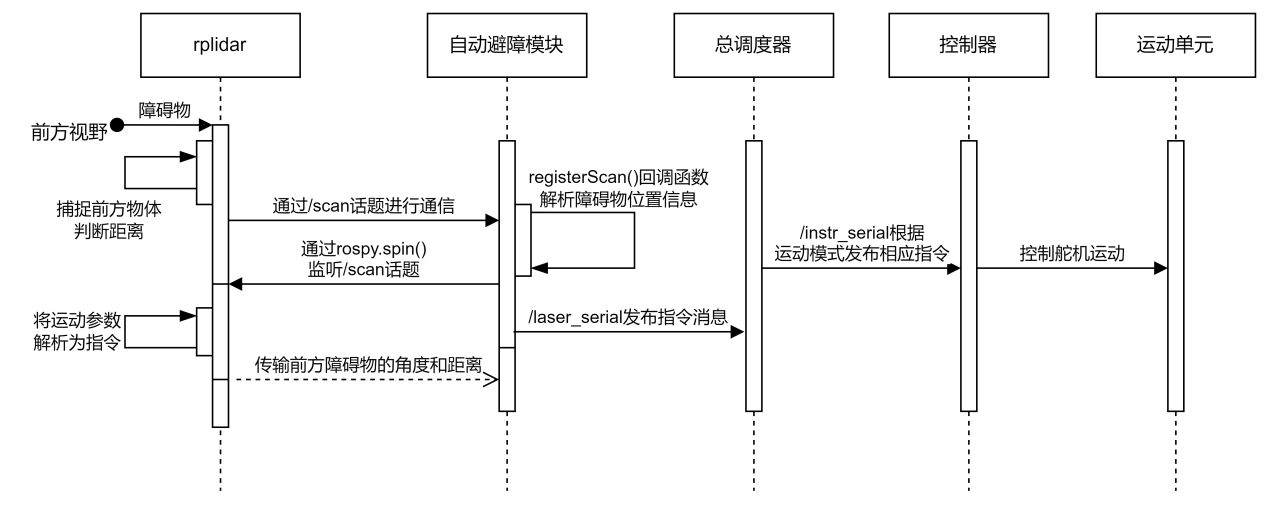


图 23自动避障流程图

## 过速预警

该功能通过轮椅IMU收集的相关数据，在轮椅于Y轴方向运动加速过快时控制轮椅的速度及时进行运动的减速或者调停。

具体实现通过在考虑IMU测得的加速度时消去了重力加速度的影响，增强了在非水平地形下的测量精准性。同时轮椅IMU测得的相关数据亦可在非水平地形下增强头部姿态探测的准确性。

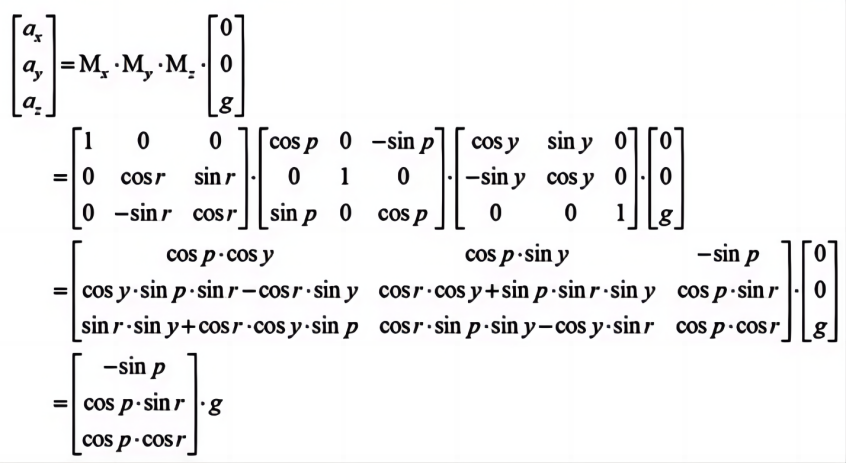


图 24消除重力加速度示意图

## 异常检测与处理

### 自动避障异常

为了最大程度保证轮椅乘坐者的安全，考虑到通信过程中可能出现的不能及时发送和收到数据等异常问题，我们设置了以下保障措施。雷达避障模块检测到障碍物后，会发布对应的控制指令和推杆距离，调度器收到后会通知控制器进行制动。避障模块在发布后会启动一个Timer来监测轮椅当前速度是否减小，若在阈值时间内轮椅速度并未发生改变，则直接对舵机模块进行驱动，将速度减为0。

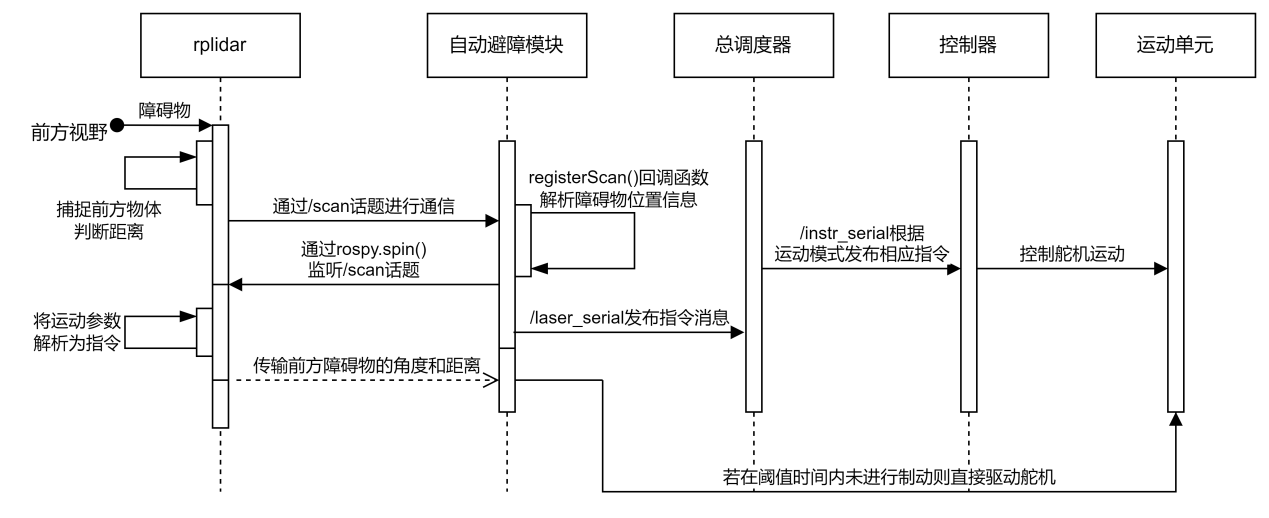


图 25自动避障异常处理

### 头控模式异常

考虑到IMU与树莓派蓝牙连接的不稳定，可能出现运动过程中蓝牙突然断开连接从而造成不必要的情况，需要对头控模式进行特殊的处理。在头控模式下，若头控指令在阈值时间内未发生变化，则可以认为可能发生了连接断开的情况，需要对IMU的蓝牙连接状态进行检测。若头部感知模块检测到与树莓派的连接已断开，则立即将轮椅速度制动为0，同时重新与IMU建立连接，最大程度保证轮椅乘坐者的安全。

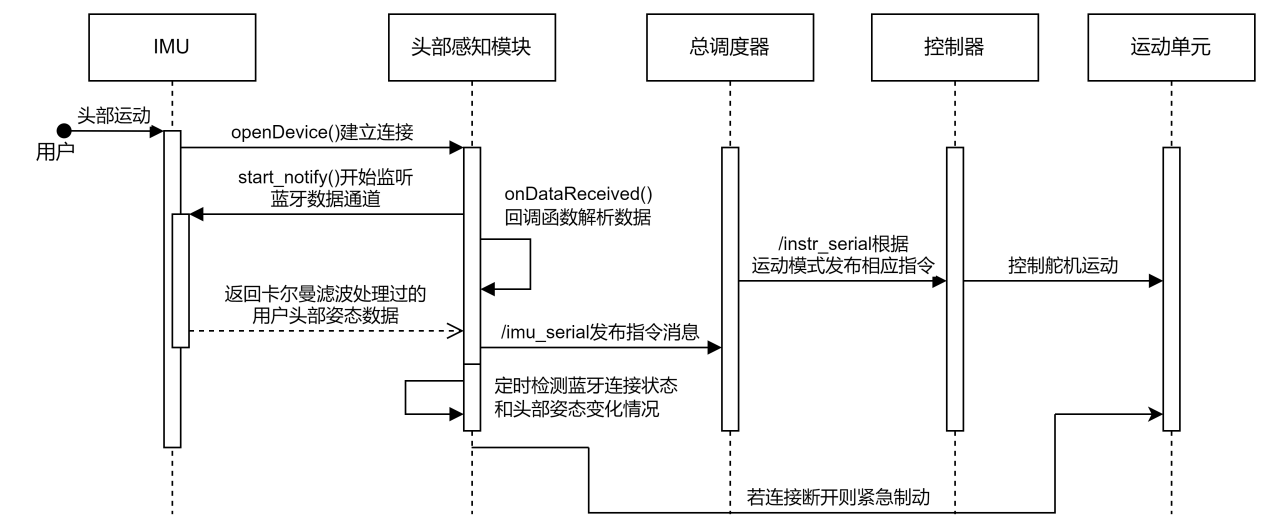


图 26 头控模式异常处理

### 跟踪模式异常

跟踪过程中跟踪目标可能突然消失在视野中，此时应将轮椅速度制动为0保证安全，同时需要在APP上提示用户跟踪目标消失在视野中。

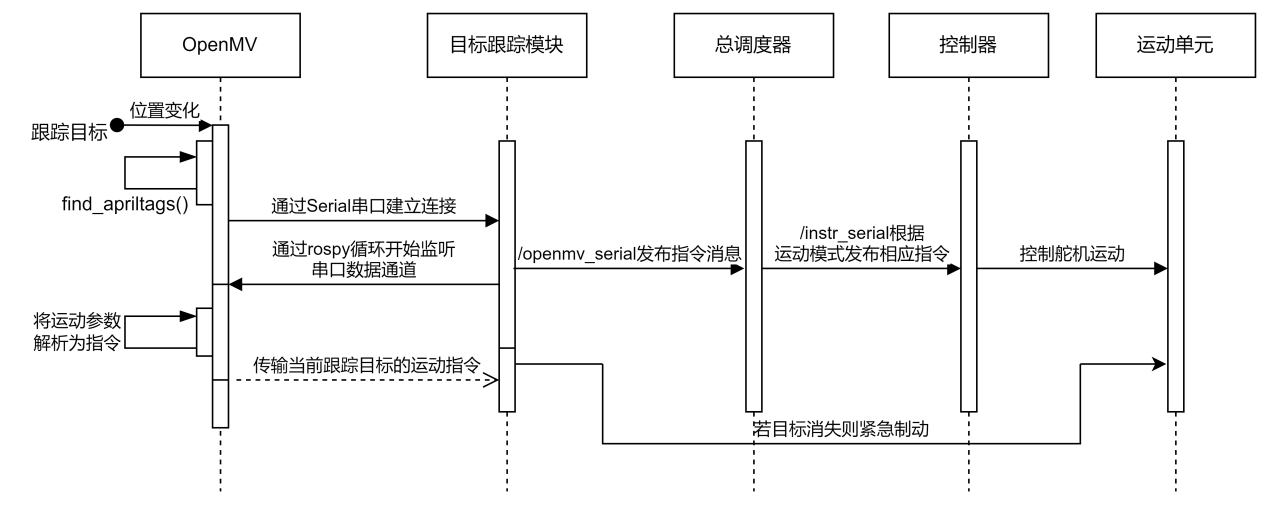


图 27 跟踪模式异常处理

### 后端异常

用户界面后端出现异常无法响应前端的请求时，前端需要设置一个时限，当超过这一时限就向用户反馈，防止出现长时间等待无响应。

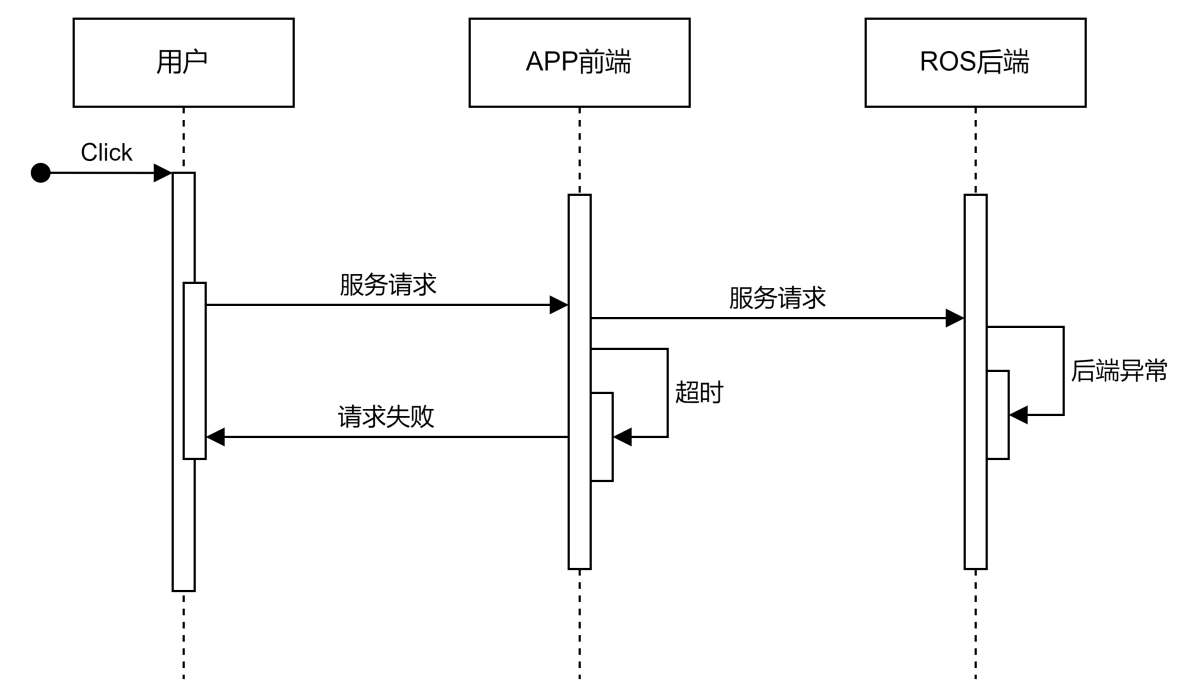


图 28 后端异常处理时序图

### 频繁点击按钮异常

用户在使用过程中可能会出现频繁点击界面某一按钮的情况，需要限制为只响应第一次请求，避免后端由于请求过多而崩溃。

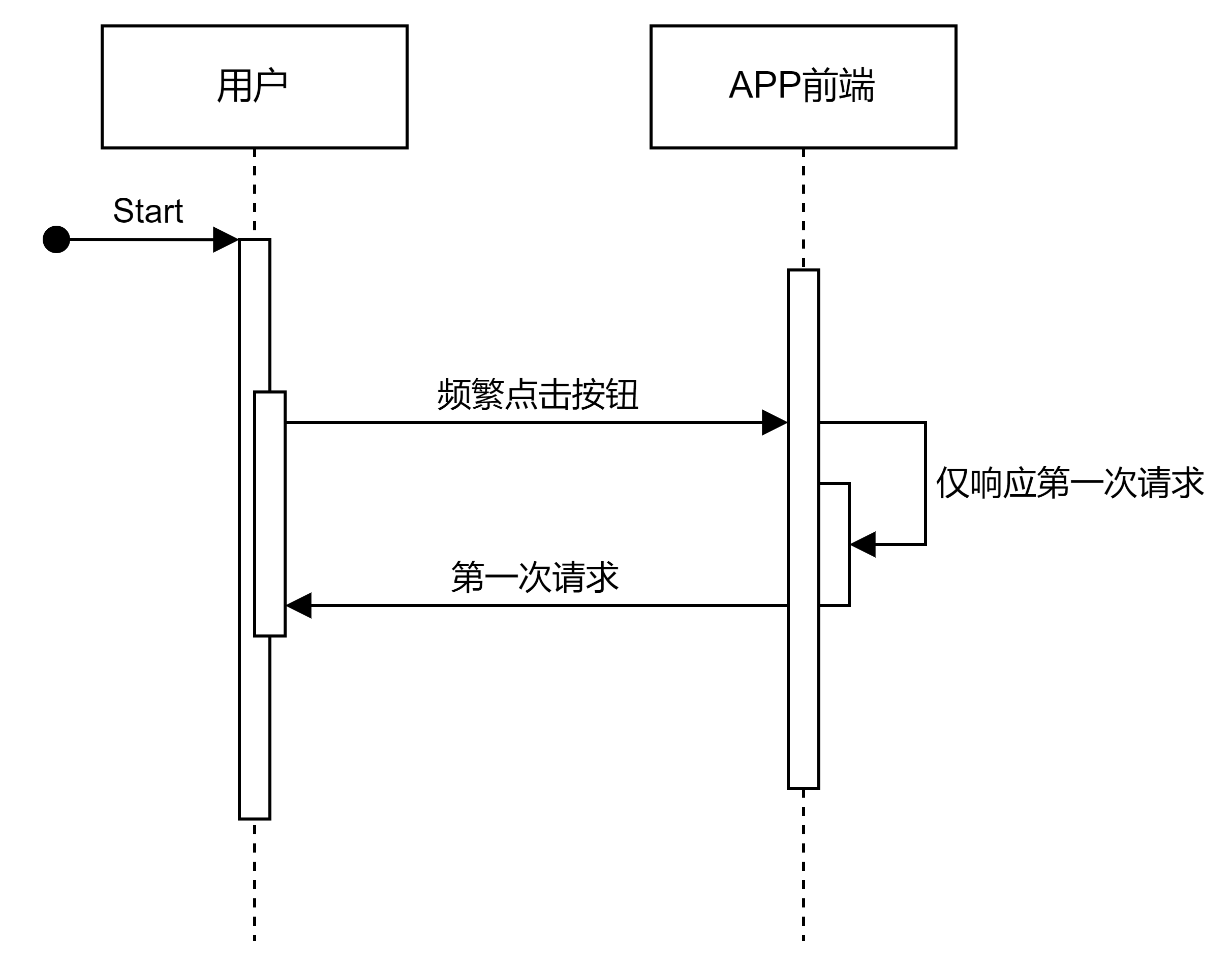


图 29 频繁点击异常处理时序图

# 系统维护

在用户需要重置系统，如环境变化，或者有新的功能发布需要进行系统更新时，用户能够通过APP的对应按钮实现这些需求。

## 系统升级

用户点击APP右上角系统升级按钮后，会向后端调度中心发送一个系统升级的请求，调度中心收到后会执行对应的脚本文件，即从远端仓库中拉取最新的代码，然后catkin\_make编译完成系统升级。若系统升级成功则返回200，并由APP为用户显示相应的“升级成功”的提示；若已经是系统的最新版本则返回300，并由APP为用户显示相应的“已经是最新版本”的提示；若由于网络原因不能更新成功，则返回404，提示用户“网络不稳定”。

## 恢复出厂设置

在用户选择重置IP时，APP会将轮椅中的树莓派的IP地址更新为默认的出厂地址，避免IP地址频繁发生变化导致的连接不稳定情况。

# 运行与开发环境

## 运行环境

### Raspberry Pi 4B + RPLIDAR S1 + OpenMV4 H7 R2

计划在项目实际部署时使用一台安装了Ubuntu系统的树莓派Raspberry Pi 4B，负责整套系统的控制与运算，其接受来自激光雷达（型号为RPLIDAR S1）、摄像头（型号为OpenMV4 H7 R2）、指令发送装置的数据，分析处理后生成最终的控制指令，发送至轮椅操控装置，其中激光雷达用于探测前方障碍物，其拥有足够的探测半径，并可在室外使用，满足产品的使用场景；摄像头用于捕捉轮椅前方的视觉图像，从而辅助探测障碍。

### OpenMV相关操作环境（MicroPython等）

在实际运行时实现与摄像头的通信，接受来自摄像头的图像信号，以便进行进一步处理，运用OpenMV内置库Image帮助实时识别跟踪物体，同时可以辅助实现项目的避障功能。

### Android

与用户交互的APP通过Android11进行开发，通过XML和Java8进行用户界面的编写，并且使用WebSocket与轮椅中的树莓派通信。

## 开发环境

除去运行所需的环境之外，项目开发过程中还需额外使用以下环境：

### Solidworks

用于完成轮椅操纵装置以及头戴式指令发送装置的物理建模工作，便于按规格制造实物以及对于两种装置的进一步改进。

### Ros-melodic + Gazebo + RViz

部署于虚拟机内，组成本项目的开发环境，用于在开发中早期以及实际设备未就位之时进行项目的部分仿真工作，方便调试。

# 需求可追踪性说明

在系统设计方案中，服务层用于完成用户与项目本身的交互与控制，其最主要的功能是实现功能性需求中所提出的两种控制方式，即通过手机客户端与头戴式控制器产生两种不同形式的指令，对轮椅进行直接的控制。

功能层面的组件则是对于项目运行时所需处理的业务的抽象表示，其通过综合硬件层的基础硬件功能来实现更为复杂的业务，例如通过自动避障模块执行紧急情况下的避障，满足对于安全性的非功能需求；使用物体识别模块实现分析得到障碍物数据的功能；并且，使用树莓派充当总调度器，满足硬件组件之间的通信需求。

硬件层则由项目基础的硬件组件以及这些组件所能提供的最为基础的功能所构成，在数据收集方面，有激光雷达与摄像头单元完成采集环境信息的功能，惯性传感器收集头部微动数据，辅助完成生成用户命令的功能；在数据处理方面的功能，包括障碍物数据的分析以及控制指令的决定，则由微型计算器（树莓派）承担；在指令的执行方面，使用了自行设计的带有舵机的操纵装置完成直接控制轮椅操纵杆的功能，其的设计也体现了产品的普适性这一非功能需求。