

# 技术文档

## 1. 系统需求分析

### 1.1 功能性需求

#### 1.1.1 SLAM建图

机器人可以通过激光雷达、IMU传感器，实时获取周围环境数据，通过SLAM算法构建实时地图，帮助机器人感知实时环境信息，进行导航。

#### 1.1.2 实时定位

机器人可以通过激光雷达，实时获取周围的环境信息，通过匹配激光雷达点云数据与预先构建的环境地图信息，得到机器人当前的定位信息。

#### 1.1.3 自主导航避障

基于SLAM（同步定位与建图）技术的自主导航和避障功能利用激光雷达等传感器实时感知环境，生成高精度的环境地图，同时通过路径规划算法为机器人规划最优行驶路线。SLAM技术能够在未知环境中准确定位机器人位置，并连续更新地图数据，使机器人能够自主导航，避开障碍物，安全到达目标位置。

#### 1.1.4 路径规划巡逻

通过对工厂环境进行SLAM建图，机器人可以按照预先设定的导航点进行移动，按照设定好的路径进行巡逻。

#### 1.1.5 图像识别

机器人可以通过摄像头捕获到当前环境的图像信息，实时识别当前图像当中的人物、火焰、烟雾信息，起到对工厂安全的检测作用。

#### 1.1.6 服务器通信

在检测到人物、火焰、烟雾信息时，机器人会实时通知后端服务器，服务器将显示机器人当前位置出现安全问题。

### 1.2 非功能性需求

#### 1.2.1 SLAM建图准确性指标

位置精度：

- 指标：平均定位误差
- 标准： $\leq 5\text{ cm}$

### 1.2.2 定位准确性指标

位置精度：

- 指标：平均定位误差
- 标准： $\leq 5\text{ cm}$

### 1.2.3 路径规划准确性指标

路径偏差：

- 指标：平均路径偏差
- 标准： $\leq 10\text{ cm}$

路径长度：

- 指标：平均路径偏差
- 标准： $\leq 10\text{ cm}$

避障成功率：

- 指标：避障成功率
- 标准： $\geq 95\%$

### 1.2.4 运动控制准确性指标

偏航误差：

- 指标：平均偏航误差
- 标准： $\leq 0.034\text{rad}$

路径跟踪误差：

- 指标：平均路径跟踪偏差
- 标准： $\leq 5\text{ cm}$

### 1.2.5 环境感知准确性指标

检测率：

- 指标：检测率
- 标准： $\geq 98\%$

误报率：

- 指标：误报率
- 标准： $\leq 1\%$

漏报率：

- 指标：漏报率
- 标准： $\leq 2\%$

### 1.2.6 响应时间

初始响应时间：

- 指标：初始响应时间（Initial Response Time, IRT）
- 标准： $\leq 100\text{ ms}$

总响应时间：

- 指标：总响应时间（Total Response Time, TRT）
- 标准： $\leq 500\text{ ms}$

交互响应时间：

- 指标：交互响应时间（Interactive Response Time, IRT）
- 标准： $\leq 200\text{ ms}$

### 1.2.6 系统稳定性

长时间运行稳定性：

- 指标：平均故障间隔时间（Mean Time Between Failures, MTBF）。
- 标准： $\geq 48\text{ 小时}$

交互稳定性：

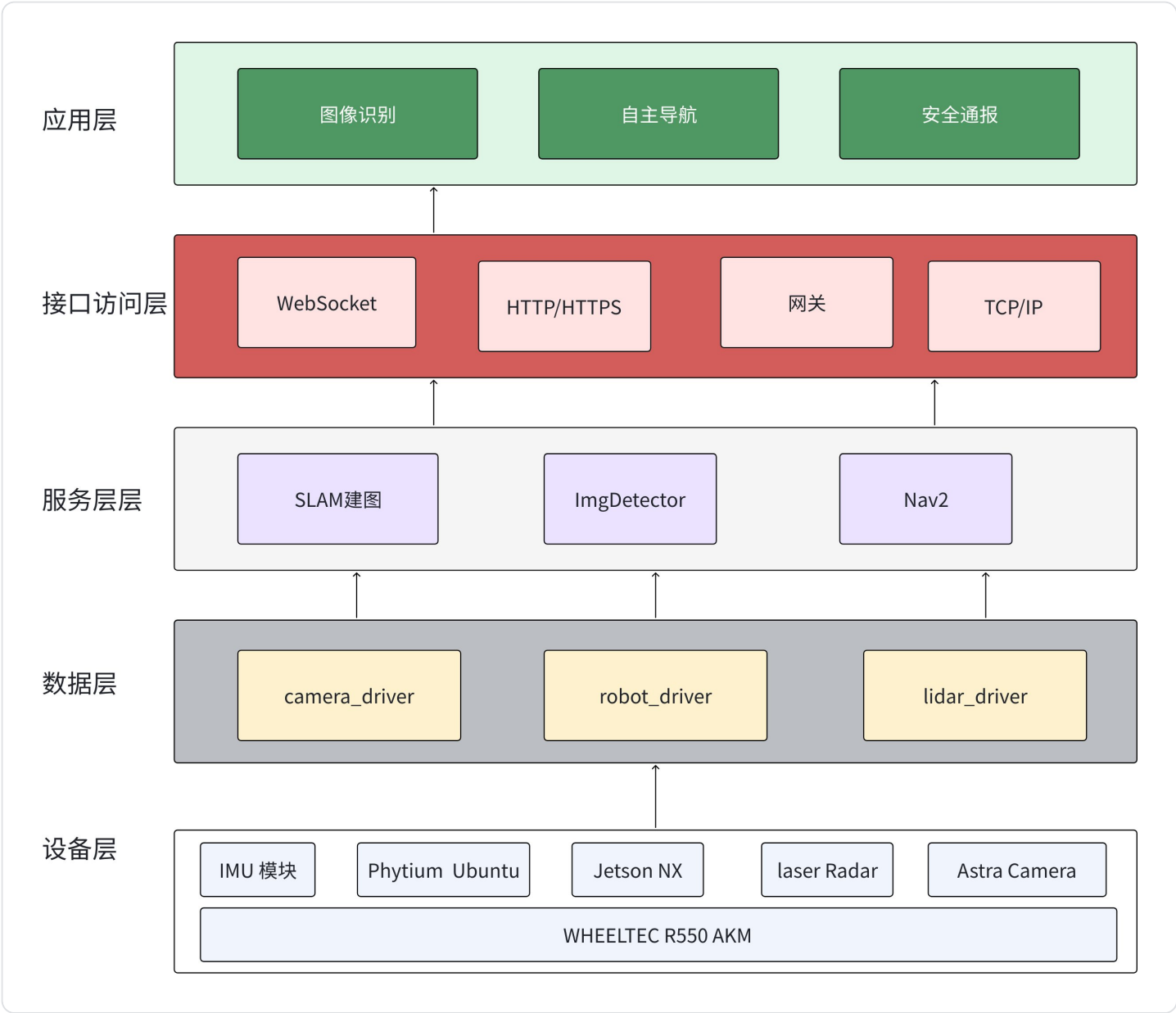
- 指标：交互错误率（Interaction Error Rate, IER）
- 标准： $\leq 1\%$

恢复能力：

- 指标：平均恢复时间（Mean Time to Recovery, MTTR）
- 标准： $\leq 1\text{ 分钟}$

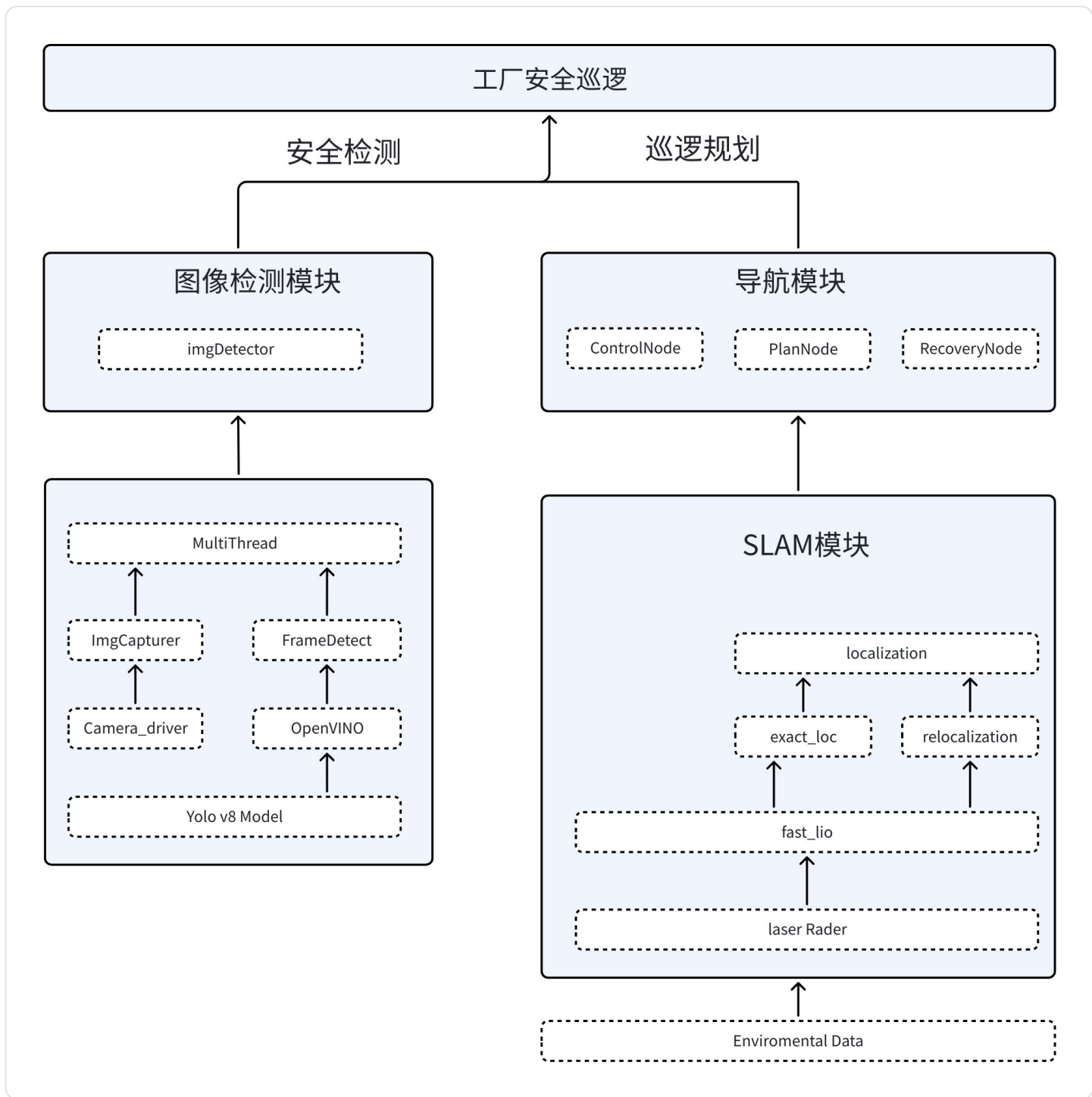
## 2. 系统架构设计

### 2.1 系统结构



## 2.2 模块划分与多核架构

### 2.2.1 系统模块划分



### 2.2.1.1 工厂安全巡逻

本项目设计开发的机器人的目标是能够在工厂环境内进行实时巡逻并进行安全监测，使用其上搭载的激光雷达进行SLAM建图定位，对工作环境进行基本建图，然后规划巡逻路线，在巡逻过程中通过相机实时识别图像当中的人物、火焰、烟雾信息，发送到后台服务器当中，起到对工厂环境内的安全巡检作用。

### 2.2.1.2 SLAM建图模块

小车搭载的激光雷达可以获取环境的3D点云数据，这些数据包含了周围环境的详细信息。数据处理与转换模块的核心功能是对3D点云数据进行降维、去噪和优化，使其适用于实时建图。SLAM模块能够高效地创建环境地图。SLAM建图过程包括定位和地图更新，通过融合传感器数据（如激光雷达、里程

计、IMU等），实现小车在未知环境中的自主定位和实时地图构建。完成建图后，系统将生成的地图发布供其他模块使用，这为后续的导航和路径规划提供了可靠的环境信息。

### 2.2.1.3 定位模块

为了实现机器人导航，我们引入ICP（Iterative Closest Point）算法作为重定位手段。ICP通过点云匹配，可在已有地图中重新校准机器人当前位姿，提升系统稳定性。此外，通过组合定位形式，将里程计坐标系与实时定位坐标系进行结合，得到高鲁棒性的定位功能。

### 2.2.1.4 导航模块

导航模块依赖于SLAM模块生成的地图，结合路径规划算法，为小车制定行驶路线。路径规划算法会考虑地图中的障碍物和目标位置，生成一条最优路径，使小车能够安全、高效地到达目标地点。导航模块不仅仅是简单地跟随路径，还包括动态避障和路径重新规划等功能。当小车在行驶过程中遇到新的障碍物或环境变化时，导航模块会实时更新路径规划，确保小车能够避开障碍物并调整行驶路线。此外，为了提高小车对环境的适应性，每隔一段时间，导航模块会与SLAM模块协作，对数据和地图进行更新。这种实时更新机制使小车能够在动态环境中保持高效和安全的自主导航能力。

### 2.2.1.5 图像识别模块

机器人通过使用其搭载的摄像头，获取当前环境的图像信息。使用Yolo v8模型对图像信息进行检测，获取当中的人物、火焰、烟雾信息。此外，该模块还使用了多线程并行处理技术，提高边缘计算单元上的图像检测效率。同时使用了OpenVINO开源框架，加速飞腾派上模型推理效率。

## 2.2.2 多核并行

飞腾派具有四个核心，系统可以同时支持四个进程的运行。在运行Linux操作系统时，可以根据不同的任务动态调度使用这四个核心。

在ROS2中，节点之间的通信没有显式地指定将节点分配到特定的核心上运行，其分配可以被视为随机分布，这使得节点的执行能够充分利用多核处理器的并行能力。

节点在多核CPU上的分配通常由操作系统的调度器决定，这是一个动态的过程。操作系统根据一些调度策略来决定将哪个节点分配给哪个核心执行。因此，不同的节点并行地在不同的核心上执行，从而实现更高的并发度和执行效率。

对多核系统的应用，我们采用了多线程处理同一任务的方式来提高任务的执行处理效率，让系统在边缘单元的运行速率更高，满足实时需求。

## 2.3 技术选型

### 2.3.1 fast\_lio建图

#### 点云数据预处理

每帧激光雷达点的坐标系都是当前时刻对应的运动坐标系，由于原始激光雷达点的采样频率非常高(如200HZ，这里指的是激光雷达在扫描时采样频率，不是单帧激光雷达数据的频率)，因此不可能在接受

到每个新点后进行处理，更实际的方法是在一定时间内积累这些点，并一次处理他们。

### EKF状态估计

fastlio算法的核心在于使用扩展卡尔曼滤波器，来融合激光雷达与惯性测量单元的数据。在滤波器中，激光点云数据与imu数据共同用于构建残差函数，并通过最小化残差来更新状态估计。通过EKF来减小惯量单元数据零漂突变，得到点云数据与惯量数据的紧耦合点云地图。

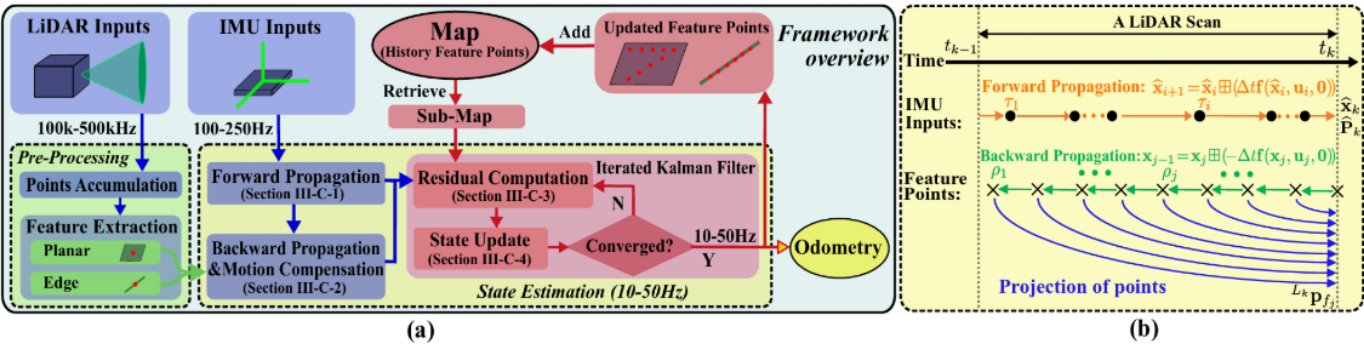
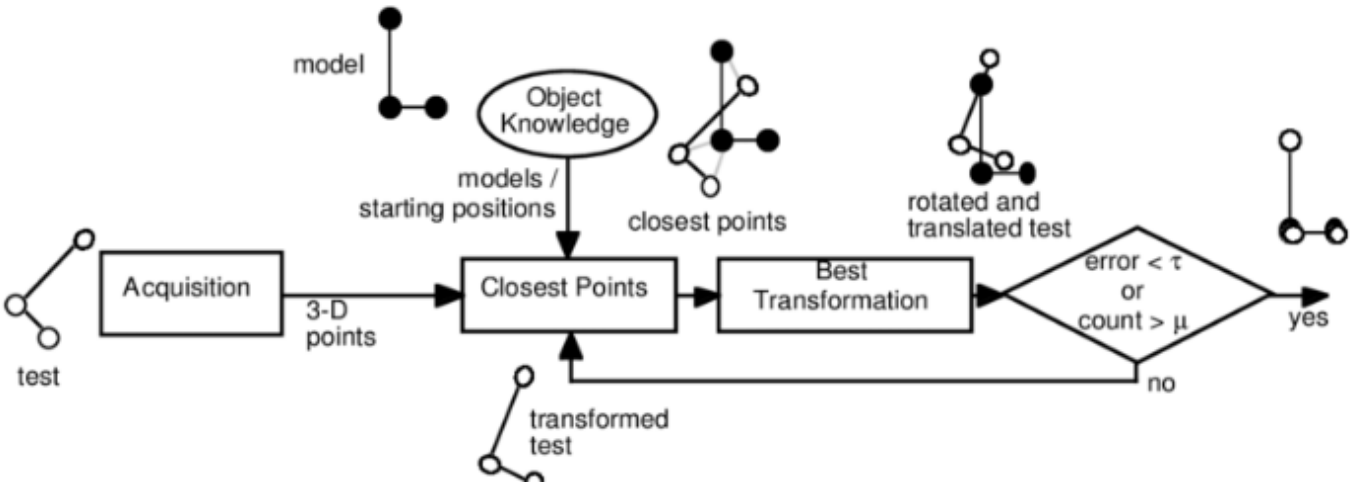


Fig. 2. System overview of FAST-LIO. (a): the overall pipeline; (b): the forward and backward propagation.

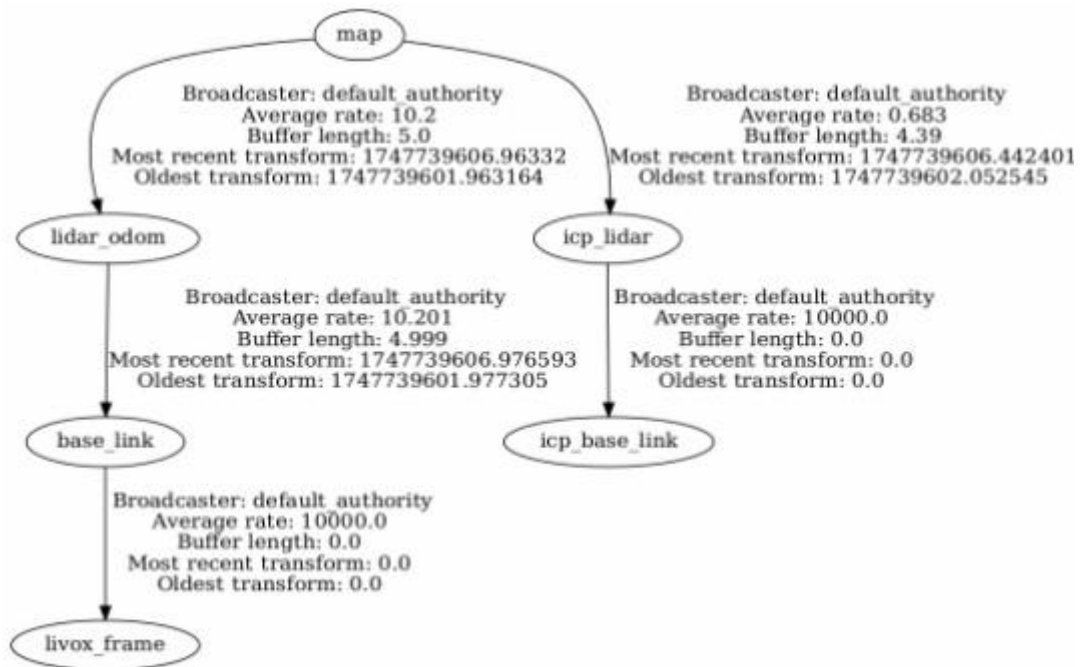
### 2.3.2 ICP重定位

ICP算法通过从数据点集中的每个点寻找参考点集中对应的最近点，形成匹配点对，然后将所有匹配点对的欧氏距离之和作为待求解的目标函数，利用奇异值分解求出旋转矩阵R和方向向量t以使目标函数最小，然后根据旋转矩阵R和方向向量t转换得到新的数据点集，并再次找到对应的点对，进行迭代求解。

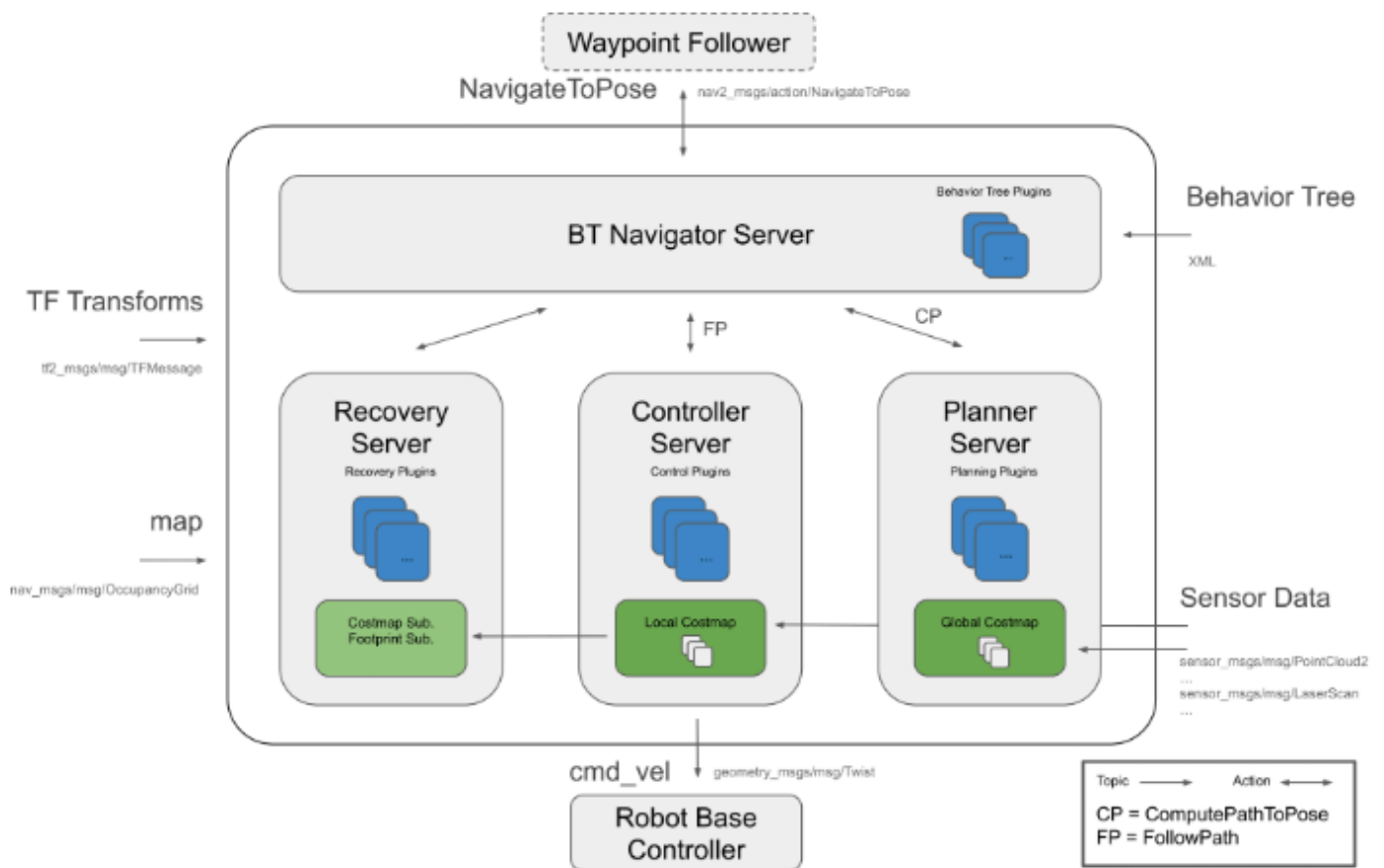


ICP算法使用时需要给出足够的初步位姿信息，否则迭代当中难以收敛。

通过组合定位的方式来获取更为精准的定位信息。使用里程计坐标与实际定位坐标，里程计坐标使用的是fast\_luo得到的紧耦合imu数据的预测点云数据来维护，实际坐标通过直接的激光雷达数据进行维护icp\_base\_link坐标系，作为定位使用的坐标系信息。



### 2.3.3 NAV2导航



nav2中提供导航行为树服务，其主要由规划节点（Planner）、控制节点（Control）、恢复节点（Recovery）三个节点构成。整个框架通过规划路径、控制机器人沿着路径运动、遇到问题自主恢复三者不断进行切换完成机器人的自主导航。

### 规划节点



规划节点会为机器人找到从a点b点的路径，nav2当中默认使用Nav2Fn\_Planner，基于网格搜索的方法规划，可以使用A\*或Dijkstra算法查找两点之间的路径。两种算法都需要通过全局代价地图来进行导航，全局代价地图基于已有的静态地图、当前激光雷达得到的动态障碍物信息与根据机器人外壳得到的膨胀范围得到。

## 控制节点

局部节点会为机器人规划几米以内的路径规划，构建机器人的运动轨迹，以避免动态障碍物。其需要局部代价地图，由当前激光雷达得到的动态障碍物信息与机器人膨胀范围组成。

## 恢复节点

恢复节点会在机器人找不到有效路径时，进行恢复行为，比如原地旋转、沿着给定方向移动、静止不动。

## 指定路径巡航

本机器人通过使用icp重定位得到其初始位置信息，通过在已有地图当中设置导航点，根据一定顺序进行循环导航，已完成工厂范围内的巡航功能。

## 2.3.4 YOLOv8人脸识别

### OpenVINO

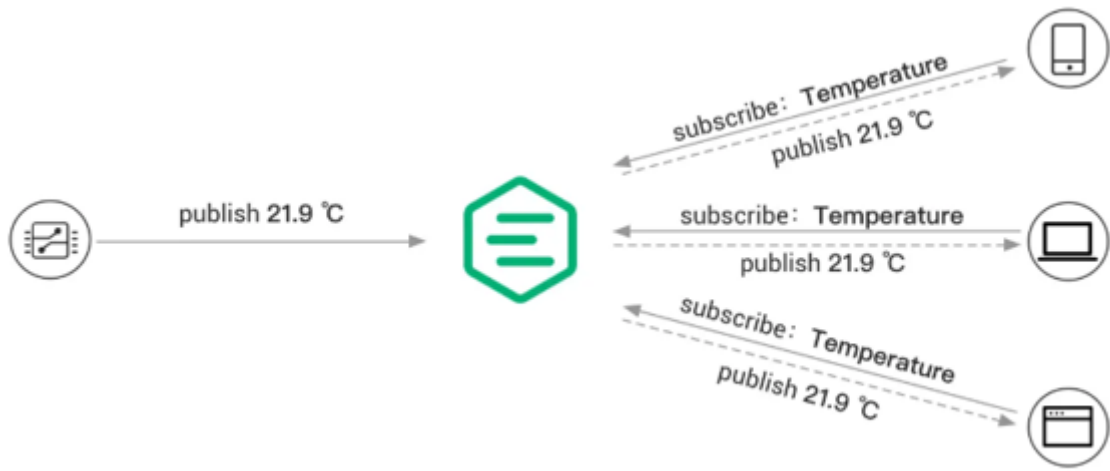
OpenVINO是英特尔推出的开源工具套件，用于优化和加速深度学习模型的推理。其核心原理围绕模型优化、硬件加速和跨平台部署展开。其最主要的部分是模型优化器与推理引擎。模型优化器能够将各种深度学习框架的模型转化为OpenVINO的IR格式，来压缩模型并且优化使用时的内存，从而达到加速作用。推理引擎让其能支持在各种硬件上执行模型的推理任务，附带有根据硬件特性自动选择最优的执行路径的功能，同时其对OpenCV图像处理库也进行了指令集优化，有显著的性能提升。

### 多线程

使用多线程技术加速图像识别效率，通过引入两个线程，一个线程进行图像捕获，放到队列当中，启动另一个线程专门进行图像识别，提高cpu的利用率，提高图像识别与推理的效率。

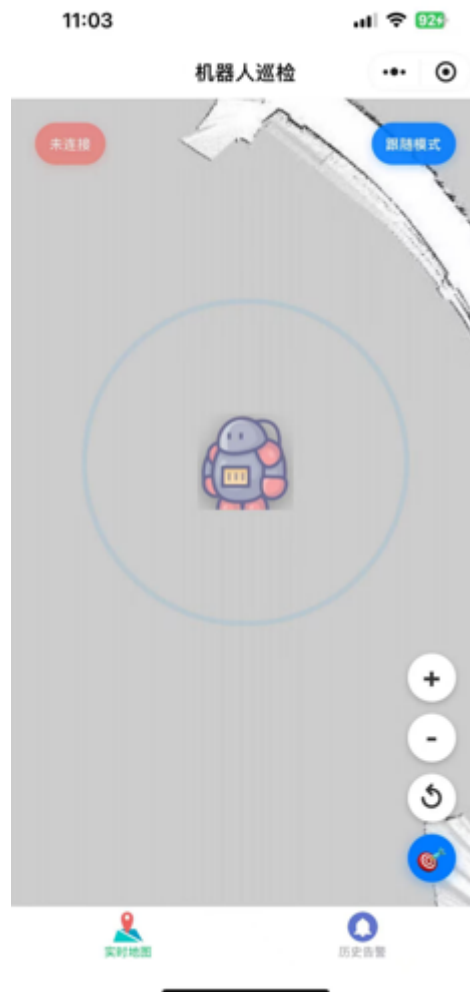
## 2.3.5 服务器

机器人通过mqtt协议与后端服务器进行通信，发送其坐标位置和预警信息。后端服务器采用springboot+mqtt架构，接收机器人的定位信息和预警信息后后，使用PostgreSQL数据库进行数据存储。



### 2.3.6 微信小程序

系统的展示部分通过微信小程序的形式进行展现。微信小程序通过websocket+http与服务器进行连接，获取当前机器人的工作情况以及历史数据，通过小程序可以查询到历史数据记录以及当前机器人的位置信息



## 2.4 接口描述

### 2.4.1 物理接口

- 以太网接口（Ethernet）：激光雷达通过以太网接口连接飞腾派，将获取到的激光扫描数据直接传输到飞腾派上进行处理和分析。
- USB接口（Universal Serial Bus）：USB接口是一种通用的串行总线接口。STM32开发板具有内置的USB接口，通过该USB连接线与飞腾派相连。飞腾派可以通过USB接口向STM32开发板发送命令或传输数据，同时STM32也可以通过USB接口接收来自飞腾板的数据。以此实现机器人底盘控制。同时，相机也通过usb接口与飞腾派相连，让飞腾派可以获取到实时的图像信息。

## 2.4.2 外部接口

主要用于机器人与服务器之间的通信，用于实时通知服务器当前工厂的安全信息。

API端点：

- URL：/api/report
- 方法：POST
- 描述：发送当前安全类型与定位信息。

请求格式：

- 请求头：Content-Type: application/json
- 请求体：

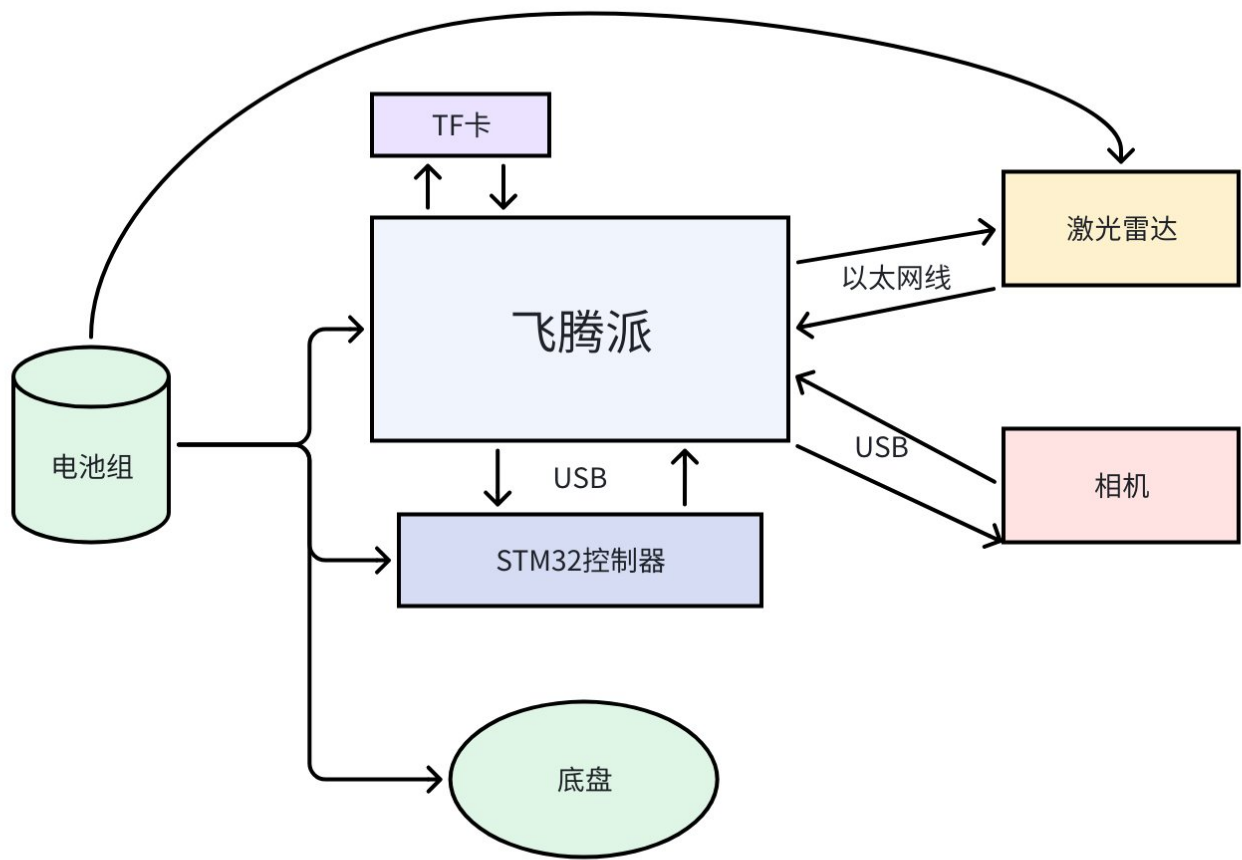
代码块

```
1  {  
2    "secure_type": "当前安全类型",  
3    "location": "当前位置"  
4  }
```

## 3. 系统详细设计

### 3.1 硬件设计

#### 3.1.1 开发板连线图



## 硬件配置说明：

### 1. 飞腾派

整个系统的核心控制部分，运行着Ubuntu20.04操作系统和ROS2。其作用包括以下几点：

- 传感器数据处理：处理来自激光雷达等传感器的数据，进行环境建模、导航、路径规划等。
- 高层次控制逻辑：执行机器人的高层次控制算法，如目标追踪、决策逻辑等，并将指令发送到STM32微控制器。

### 2. 操作系统

- Ubuntu20.04：Linux发行版，提供一个稳定和强大的开发环境。
- ROS2 (Robot Operating System2)：开源的机器人软件框架，提供了一套标准的服务（如硬件抽象、设备驱动、库、可视化工具等），用于构建机器人应用。

### 3. TF卡

- 一种小型存储卡，用于存储操作系统、ROS包、配置文件、日志数据等数据存储。

### 4. 激光雷达 (Lidar)

- 通过发送和接收激光束来测量到周围物体的距离和方向，生成一个二维或三维的点云地图，用于机器人的环境感知，帮助机器人进行定位、导航和避障。

### 5. Microcontroller STM32 (RTOS)

- 一个基于STM32芯片的微控制器，运行实时操作系统（RTOS），处理实时性要求高的任务。
- 任务：
  - 电机控制：根据飞腾派发出的控制指令，驱动马达实现机器人运动。
  - 传感器数据采集：采集底盘上的传感器数据，如速度、方向等，并实时反馈给飞腾派。

## 6. 底盘

- 通过STM32微控制器控制，驱动机器人移动，执行运动指令。

## 7. Battery（电池）

- 为飞腾派、stm32控制器、雷达进行供电。

### 3.1.2 传感器配置

#### 3.1.2.1 IMU模块

IMU惯导模块是高性能、多功能的九轴姿态传感器，适用于机器人、无人驾驶和工业自动化等领域。将IMU模块连接到ROS主控，运行ROS动包进行数据采集和处理。IMU模块发布的数据包括加速度、角速度和姿态信息，用户可以通过ROS话题订阅这些数据，并进行后续处理和应用。以下是该模块的详细介绍：

##### 1. 九轴传感器

- 三轴加速度计：测量线性加速度，提供移动信息。
- 三轴陀螺仪：测量角速度，提供旋转信息。
- 三轴磁力计：测量地磁场，提供方位信息，帮助进行姿态和航向参考系统（AHRS）的校准。

##### 2. 高性能和高精度

- 高采样频率：传感器采样频率高达1000Hz，确保数据的实时性和精确性。
- 卡尔曼滤波器：内置强大的Sigma-Point卡尔曼滤波器（SPKF），提供高性能的姿态解算和数据融合能力，增强抗磁干扰能力。

##### 3. 工业防水设计

- 防水外壳：工业级防水设计，适用于各种恶劣环境，确保传感器在潮湿和多尘环境中的稳定运行。

##### 4. ROS兼容

- ROS驱动支持：提供完整的ROS驱动包和使用教程，方便用户在ROS环境中快速集成和使用IMU模块。
- 数据可视化：支持通过RViz等工具进行数据可视化，实时查看传感器的姿态和运动信息。

##### 4. 多种接口

- USB串口输出：通过USB接口连接，数据传输稳定可靠。
- 波特率：默认波特率为921600，确保高速数据传输。

### 3.1.2.2 激光雷达

C16激光雷达以其高精度、高分辨率和全向扫描能力，在机器人和自动驾驶领域中具有广泛的应用。其采用的TOF技术和无刷电机设计，保证了产品的可靠性和稳定性。通过串口接口和广泛的电源兼容性，C16激光雷达可以方便地集成到各种系统中，提供强大的环境感知能力和数据支持。以下是该雷达的详细介绍：

#### 1. 高性能测距

- 测量范围：C16激光雷达的测量范围可以达到30米，适用于各种复杂环境的检测。
- 扫描频率：支持高频率扫描，确保快速和实时的环境感知。

#### 2. 全向360°扫描

- 360°扫描范围：通过旋转的激光器实现全向扫描，消除盲区，确保机器人能够全方位感知周围环境。
- 高分辨率：提供高密度的点云数据，有助于精细的环境建图和障碍物检测。

#### 3. 技术原理

- TOF（飞行时间）原理：C16激光雷达采用TOF技术，通过测量激光从发射到返回的时间差来计算距离，具有高精度和稳定性的特点。
- 高精度陀螺仪：内置高精度陀螺仪，提供精准的角速度数据，辅助环境感知和导航。

#### 4. 接口和兼容性

- 多种接口：支持串口、以太网等多种接口，数据传输稳定可靠，易于集成到各种机器人系统中。
- 兼容性强：与多种机器人操作系统（ROS）兼容，方便开发者进行集成和使用。

该产品的详细说明如下：

# C16多线激光雷达参数说明

线数：16	水平角分辨率：5Hz: 0.09°; 10Hz: 0.18°; 20Hz: 0.36°
激光波长：905nm	转速：300/600/1200rpm(5/10/20Hz)
激光安全等级：Class 1 人眼安全	以太网输出：100Mbps
测距方式：脉冲式	存储温度：-40℃~+85℃
测距能力：150米测量距离	工作电压：+9VDC~+32VDC
精度(典型值)：±3cm	工作温度：-20℃~+60℃
水平视场角：360°	产品功率：12W
垂直视场角：30°	尺寸：φ102mm*H78mm
垂直角分辨率：2°	防护等级：IP67
帧率：5/10/20Hz	冲击：500 m/sec²,持续11ms
重量：1050g(加1.2米线)	振动：5Hz-2000Hz,3G rms
UDP数据包内容：三维空间坐标、反射强度、时间戳等	
出点数：320000pts/s(单回波) 640000pts/s(双回波)	

## 3.1.2.3 里程计

GMR（巨磁电阻）角度传感器和绝对值编码器适用于精密测量和控制系统，广泛应用于机器人和自动控制领域，能够在移动机器人中，提供精确的方位和位置检测，支持SLAM和导航任务。

### 1. 高精度测量

- 绝对角度测量：提供360°的绝对角度测量，分辨率高达12位，确保精确的角度检测。
- 低角度误差：在整个使用寿命和温度范围内，角度误差小于1°，保证测量的稳定性和可靠性。

### 2. 多种接口

- PWM接口：支持PWM信号输出，适用于实时角度数据传输。
- 其他接口：兼容SENT、SPC和SPI等多种接口，灵活适用于不同应用需求。

### 3. 功能安全

- ISO 26262认证：符合ISO 26262 ASIL C和ASIL D功能安全标准，适用于高安全性要求的应用场景。

该产品的详细说明如下：

# 产品信息介绍

产品	GMR角度传感器
角度范围	360°
是否有机机械限位	无，可自由旋转
角度分辨率	0.09°
信号输出方式	PWM信号输出
线性误差	<±0.35°
最大转速	20000rpm
电压范围	3-5.5V
功耗	30mW
接口规格	XH2.54 3P
工作温度	-40°C~125°C

## 3.1.3 其他硬件说明

### 3.1.3.1 R550 PLUS Auto机器人

本系统将飞腾派搭载在R550 PLUS Auto机器人机身上，用飞腾派控制完成相关功能，连接调用传感器。其上配备STM32开发板，底层搭载FreeRTOS，使用雪地防滑橡胶轮，全金属车身，还配备了万元级底盘才有的高精度悬挂系统，在普通水泥、柏油路上优势明显，还可以适应不平的地面。底盘使用的阿克曼结构，是新能源汽车市场目前广泛应用的标准结构，配备九轴IMU传感器、镭神智能的多线激光雷达，可实现精准感知及实时场景构建功能，为我们提供了低成本高性价比的机器人二次开发平台。实物图如下：





### 3.1.3.2 STM32控制板

WheelBoard的C50C STM32机器人主控板是由轮趣科技（东莞）有限公司推出的一款高性能控制板，专为机器人应用设计。该主控板基于STM32微控制器，具备以下特点：

#### 1. 硬件特性

##### a. STM32微控制器：

- 处理器型号：采用STM32系列微控制器，提供强大的计算能力和实时处理能力。
- 内存和存储：配备足够的RAM和闪存，用于复杂算法的实现和数据存储。

##### b. 丰富的接口：

- 数字和模拟I/O：多个GPIO引脚可用于连接各种传感器和执行器。
- 通信接口：包括I2C、SPI、UART、CAN等多种通信接口，方便与其他设备进行数据交换。
- 电机控制接口：专门设计的电机驱动接口，支持PWM控制，能够直接控制DC电机、步进电机等。

##### c. 传感器支持：

- 超声波传感器：用于距离测量和障碍物检测。
- 加速度计：用于检测机器人的加速度和倾斜角度，辅助导航和稳定控制。
- 其他传感器：支持连接温度、湿度、光线等多种传感器，扩展机器人的感知能力。

#### 2. 软件特性

##### a. 开发环境：

- STM32CubeIDE：官方推荐的开发环境，提供一站式的开发和调试工具。
- 支持多种编程语言：包括C、C++等，开发者可以根据需求选择合适的语言进行开发。

##### b. 丰富的库和示例代码：

- 提供丰富的驱动库和中间件，方便快速上手开发。

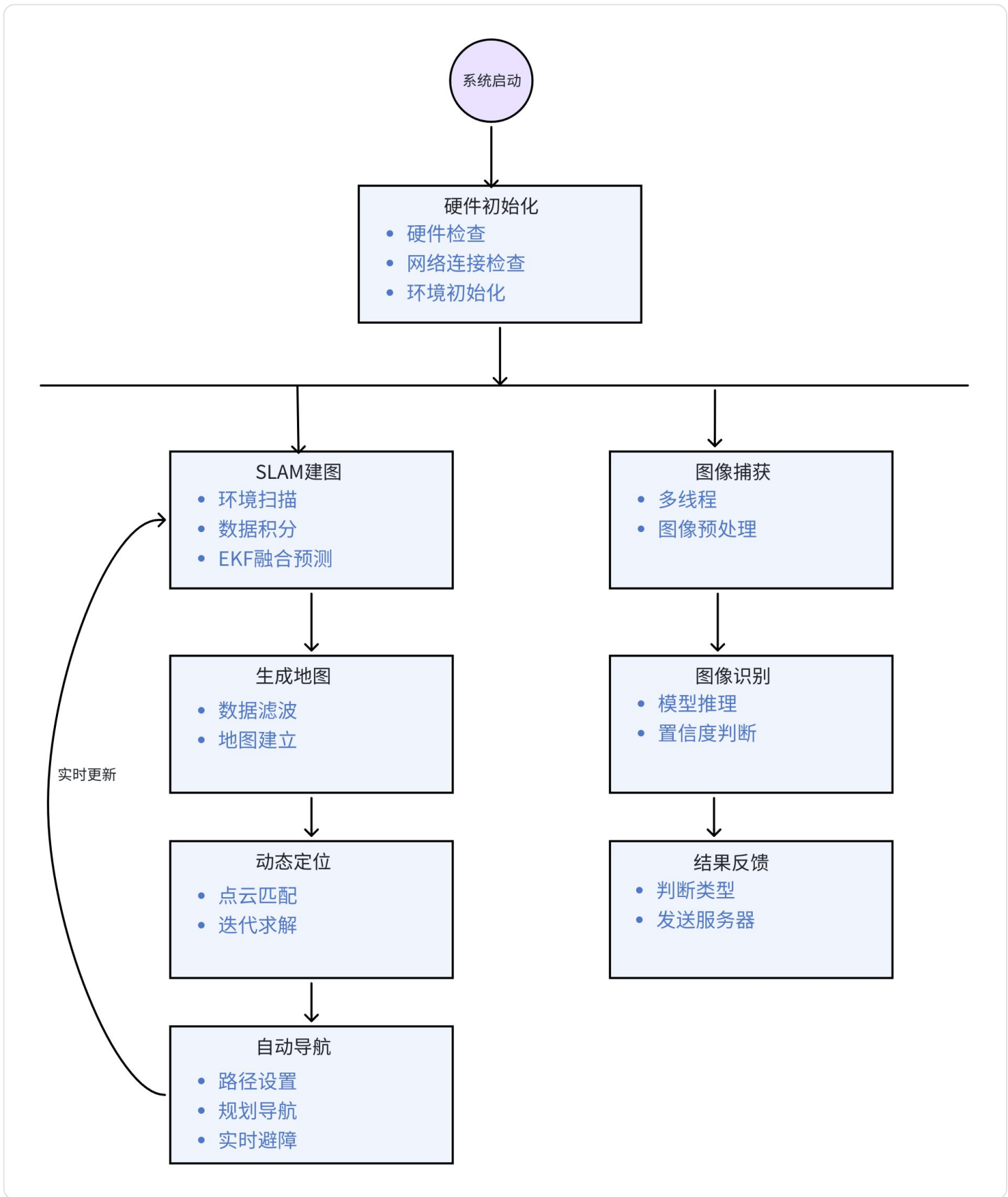
- 官方和社区提供了大量的示例代码和项目，涵盖常见的机器人控制和传感器应用。

c. 实时操作系统支持：

- 支持FreeRTOS等实时操作系统，提供任务调度和资源管理功能，提高系统的实时性和可靠性。

## 3.2 软件设计

### 3.2.1 软件流程图



- 本系统的主要流程可以阐述为：
1. 系统启动与初始化：系统对运行所需要的硬件环境进行检测，查看当前是否与控制板、相机、激光雷达、服务器连接，确保系统在正常状态下工作。
  2. 并行处理SLAM建图与图像识别功能
  3. 激光雷达获取环境信息，建立动态地图。相机启动多线程捕获图像帧，进行预处理。

4. 根据得到的点云信息与预先建立好的地图进行点云匹配，迭代求解，得到定位信息。将获取到的图像帧给到模型进行推理。
5. 进行路径规划并执行导航功能，在规划好的大致路线上进行巡检。如果图像识别到人物、火焰、烟雾信息，则反馈给服务器当前的定位信息与安全问题。
6. 机器人实时发送定位信息到服务器，服务器将定位信息发送给微信小程序，微信小程序实时刷新当前机器人的位置，如果机器人发送预警信息，则将预警信息存储到服务器的数据库当中，微信小程序会显示数据库当中的内容，用户可以在历史记录汇总看到具体的预警信息与时间点。

## 3.2.2 关键代码分析

### 3.2.2.1 fast\_lio建图

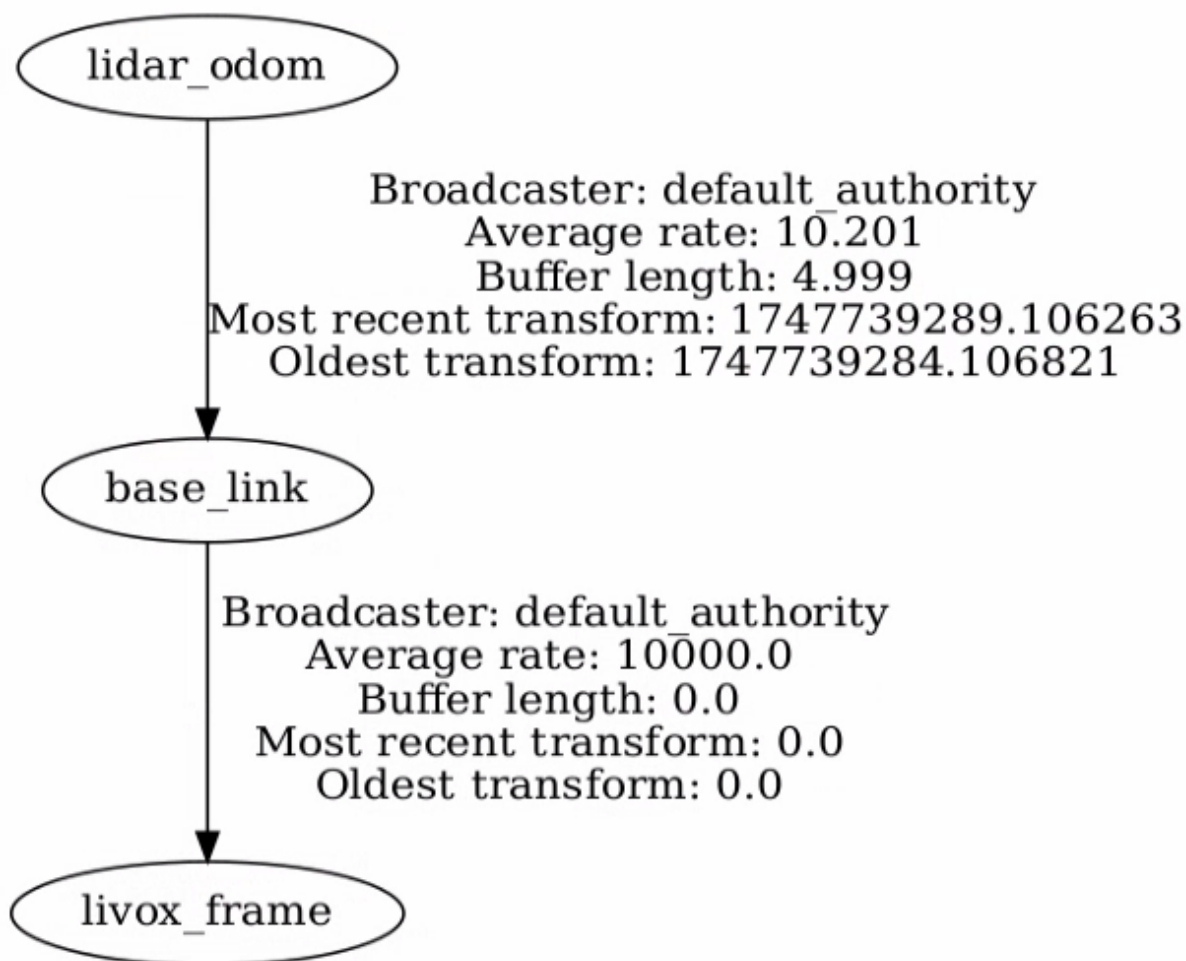
fast\_lio算法首先会对点云数据进行积分，来匹配imu单元收集频率，同时会对收集到的点云数据进行预处理，计算他的曲率并进行分类管理。然后将收集到的点云数据与imu数据输入到扩展卡尔曼滤波器当中，构建残差方程，求解更新量，获取更新后后的数据来更新地图，以这种紧耦合imu与点云数据的方式来获取高鲁棒性的点云地图。

以下为fast\_lio的节点通讯图



坐标系定义如下

- lidar\_odom: 通过fast\_lio建立的地图坐标系。
- base\_link: 机器人本地坐标系
- livox\_frame: 激光点云数据帧坐标系



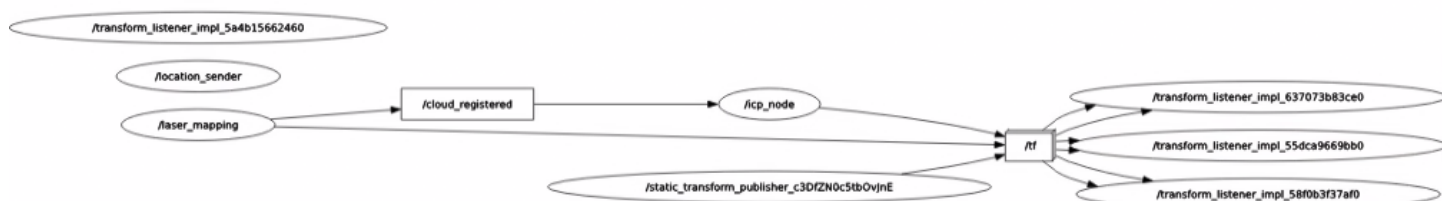
使用时需要根据激光雷达与机器人的相对位置定义坐标系，然后构建地图。

### 3.2.2.2 ICP定位

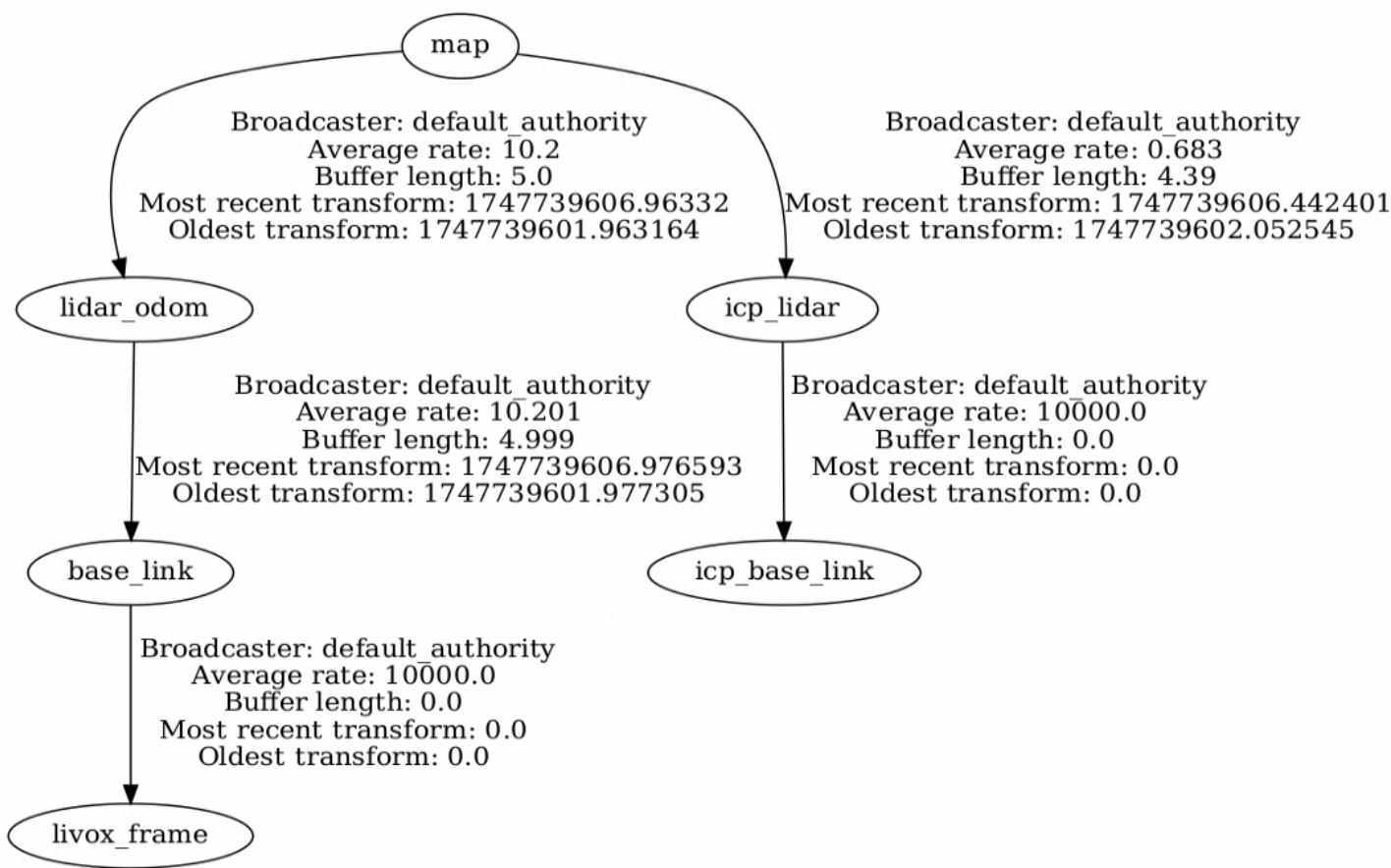
ICP定位算法是最为经典的机器人定位算法。其首先会将当前获取到的点云数据与预先建立好的地图来进行点云匹配，其使用了KD-Tree进行加速搜索临近点，提高初始点云匹配的效率。然后，将所有匹配点对的欧氏距离之和作为待求解的目标函数，利用奇异值分解求出旋转矩阵R和方向向量t以使目标函数最小，然后根据旋转矩阵R和方向向量t转换得到新的数据点集，并再次找到对应的点对，进行迭代求解。

本机器人通过组合定位的方式来获取更为精准的定位信息。使用里程计坐标与实际定位坐标，里程计坐标使用的是fast\_lio得到的紧耦合imu数据的预测点云数据来维护，实际坐标通过直接的激光雷达数据进行维护icp\_base\_link坐标系，作为定位使用的坐标系信息。

以下为定位的节点通讯图



以下为定位的坐标系关系图



3.2.2.3 图像识别

图像识别模块openvino框架，通过转换后的IR格式模型进行模型推理。图像识别时会启用多线程捕获图像数据帧，然后将图像数据帧输入到模型当中，得到置信度与结果锚框。

4. 系统测试与分析

4.1 功能测试

4.1.1 SLAM建图导航功能验证

- 传感器融合测试：
  - 测试目的：验证系统是否能结合激光雷达、IMU、里程计等多种传感器，实时获取数据。
  - 测试方法：在不同环境中进行测试，确保所有传感器的数据都能正确获取并融合。
  - 预期结果：传感器数据能够实时、准确地融合并提供完整的环境信息。
- 定位与地图构建测试：
  - 测试目的：验证SLAM算法是否能构建实时地图，实现精准定位。
  - 测试方法：在复杂环境中进行导航测试，验证地图构建的准确性和定位的精度。

- 预期结果：系统能够构建精确的实时地图，并能在不同环境中保持高精度定位。

## 4.1.2 图像识别功能验证

人物识别测试：

- 测试目的：验证系统能否正确识别图像当中的人物。
- 测试方法：使用多张含有人物的图像进行测试。
- 预期结果：系统能够正确识别出图像当中的人物位置。

火焰识别测试：

- 测试目的：验证系统能否正确识别图像当中的火焰。
- 测试方法：使用多张含有火焰的图像进行测试。
- 预期结果：系统能够正确识别出图像当中的火焰位置。

烟雾识别测试：

- 测试目的：验证系统能否正确识别烟雾当中的人物。
- 测试方法：使用多张含有烟雾的图像进行测试。
- 预期结果：系统能够正确识别出图像当中的烟雾位置。

## 4.2 性能测试

### 4.2.1 准确性测试

SLAM建图准确性测试：

- 测试目的：验证SLAM算法构建的地图和定位的准确性。◦
- 测试方法：通过在已知环境中行驶，对比构建的地图与实际环境的差异。
- 预期结果：SLAM建图的误差在可接受范围内，定位误差小。

图像识别输出准确性测试：

- 测试目的：验证图像识别结果是否准确。
- 测试方法：给出多个图片进行测试，检查正确率。
- 预期结果：模型识别图像准确率高，符合预期标准。
- 实际结果：识别准确率在70%

### 4.2.2 响应时间测试

SLAM建图速度测试：

- 测试目的：验证系统在复杂多变环境下能否进行及时准确的避障
- 测试方法：模拟商场中人流较多的环境，从广告车运行路线上经过



- 预期结果：系统快速检测到障碍出现，并且及时停车
- 实际结果：可以在1s完成响应

图像检测速度测试：

- 测试目的：验证系统对图像信息的反应速度。
- 测试方法：记录从图像出现到检测完成的时间。
- 预期结果：系统响应时间短，能够迅速处理图像信息。
- 实际结果：平均识别时间在1.08s

### 4.2.3 系统稳定性测试

长时间运转稳定性测试：

- 测试目的：验证系统在长时间运行中是否稳定，不出现故障。
- 测试方法：连续运行系统，观察是否有崩溃、错误或性能下降的情况。
- 预期结果：系统能够长时间稳定运行，不出现故障。
- 实际结果：系统在运行时会不定时因为内存冲突而发生节点崩溃的情况

## 5. 整体方案评价

当前我们的方案单独模块可以较为顺利的执行，但是加入导航模块后板载负载显著增加。导航模块的执行已经出现较为明显的卡顿现象。除此之外，当前通过多线程方案的图像识别会导致爆内存等问题导致程序崩溃。

为了让所有模块正常运行，曾尝试过将图像识别模块搬移到服务器上，板载只做到一个数据传输的网关作用，以此进一步减小边缘端负载。但板载端网卡传输速率有限，1080p30帧传输会有明显的卡顿与帧丢失。当前通过增加边缘端算力的形式，以增加一块jetson NX进行导航模块的执行，飞腾派单独运行图像识别的形式进行，可以满足基本的任务指标。