**全国大学生物联网设计竞赛文档**

AR502边缘计算低功耗视觉巡检系统

|  |  |
| --- | --- |
| 学校名称： | 常州工学院 |
| 团队名称： | AR502边缘计算  低功耗视觉巡检系统 |
| 队长： | 顾舒腾 |
| 队员1： | 朱佩韦 |
| 队员2： | 许子涵 |
| 队员3： | 崔正阳 |

全国大学生物联网设计竞赛组委会

2024年7月

**AR502边缘计算低功耗视觉巡检系统**

# 摘要

随着石化工业园区向大型化、规模化发展，现场安全监管和设备维护任务日益繁重。智能巡检机器人作为特种机器人的一种，其在电力、石化、煤矿等场景的应用日益增多，特别是在人口老龄化和劳动力短缺问题的背景下，国家的投资重点逐渐从“基础建设”转向“运维”，智能巡检机器人的发展状况与前景成为热点问题。

本设计根据某石化工业园区的实际巡检需求，整合了Android、iOS、Harmony OS多端操作系统的统一NT架构APP，5G通信技术，以及以综合管理系统为核心的智能巡检解决方案，设计出一套基于openEuler\_embedded操作系统的视觉巡检小车操作平台。

该平台是通过5G移动通信技术将自动采集和人工记录的数据实时发送至AR502H-CN网关，进而传输至综合管理系统进行数据分析。此外，平台采用视觉识别和SLAM建图技术，管理人员可通过手机APP或电脑网页查询巡检人员工作状态及位置信息等，并根据相关信息及时设置或修正小车的巡检路线。

设计内容：需求分析和总体方案设计、数据采集设备设计、手持移动终端APP和嵌入式Linux平台开发、综合管理系统搭建等设计模块、数据的传输与存储安全等设计模块

设计优势： 1. 通过多次巡检任务的测试，系统达到了设计的预期目标功能，有效提升了巡检质量管控，实现了设备运行参数和安全隐患的及时数据录入；2. 本设计为化工园区的安全生产和智能化管理提供了一定的技术支持，同时展示了智能巡检机器人在工业自动化和智能制造领域的应用前景。

**关键词：AR502H-CN 边缘计算 openEuler\_embedded Yocto**

**目 录**

[摘要 I](#_Toc172749498)

[第一章 设计需求分析 3](#_Toc172749499)

[1.1特种机器人的市场需求与发展背景 3](#_Toc172749500)

[1.2系统设计目标与解决的核心问题 4](#_Toc172749501)

[1.3现有产品或应用的改进 4](#_Toc172749502)

[第二章 特色与创新 6](#_Toc172749503)

[2.1 创新优势分析 6](#_Toc172749504)

[2.1.1 技术创新点 6](#_Toc172749505)

[2.1.2 应用创新点 7](#_Toc172749506)

[2.2创新设计 7](#_Toc172749507)

[第三章 功能设计 8](#_Toc172749508)

[3.1 功能模块概述 8](#_Toc172749509)

[3.1.1数据采集与实时监控 8](#_Toc172749510)

[3.1.2智能巡检路径规划 8](#_Toc172749511)

[3.1.3故障检测与报警 8](#_Toc172749512)

[3.1.4多端操作界面 8](#_Toc172749513)

[3.1.5 综合管理系统 9](#_Toc172749514)

[3.1.6 安全与隐私保护 9](#_Toc172749515)

[3.1.7 环境适应性设计 9](#_Toc172749516)

[3.1.8 远程控制与协同作业 10](#_Toc172749517)

[3.2多端操作界面 10](#_Toc172749518)

[3.2.1 多平台兼容性 10](#_Toc172749519)

[3.2.2 用户体验（UX）设计 10](#_Toc172749520)

[3.2.3 JVM的JAVA程序源码设计 11](#_Toc172749521)

[3.3综合管理系统 11](#_Toc172749522)

[3.3.1 前端设计 11](#_Toc172749523)

[3.3.2 后端设计 11](#_Toc172749524)

[3.3.3 系统架构 12](#_Toc172749525)

[第四章 系统实现 13](#_Toc172749526)

[4.1物联网技术架构 13](#_Toc172749527)

[4.1.1 感知层 13](#_Toc172749528)

[4.1.2 传输层 13](#_Toc172749529)

[4.1.3 控制层 16](#_Toc172749530)

[4.1.4 软件开发技术 17](#_Toc172749531)

[4.2数据挖掘和可视化 27](#_Toc172749532)

[4.2.1ModelArts 28](#_Toc172749533)

[4.2.2Astro大屏应用 Astro Canvas 28](#_Toc172749534)

[第五章 其他内容 29](#_Toc172749535)

[5.1 工业设计与成本计算 29](#_Toc172749536)

[5.1.1 工业设计 29](#_Toc172749537)

[5.1.2 成本计算 30](#_Toc172749538)

[5.1.3 法规遵从性 30](#_Toc172749539)

[5.1.4 市场推广策略 30](#_Toc172749540)

[5.1.5 风险管理 30](#_Toc172749541)

[参考文献 31](#_Toc172749542)

# 第一章 设计需求分析

## 1.1特种机器人的市场需求与发展背景

智能巡检机器人作为特种机器人的一种，其设计需求源于对特定作业环境的自动化和智能化改造。这些机器人主要用于替代人工对设备进行检查和维护，特别是在电力、石化、煤矿等高风险或环境恶劣的场景中。随着机器人技术的快速发展，加之人口老龄化和劳动力短缺问题的加剧，企业对安全生产的重视程度不断加深，国家投资重点也逐渐从“基础建设”向“运维”转变，使得智能巡检机器人的发展状况与前景成为业界关注的热点问题。



图1 智能巡检机器人和传统人工巡检对比

## 1.2现有产品或应用的改进

横向对比现有产品，通过设计感知发现以下触点都具备：

**数据采集与通信**：采用基于STM32微处理器的数据采集设备，实现多路4-20mA电流信号的采集，并通过5G技术将数据快速、安全地传输至综合管理系统。

**智能巡检终端**：开发了适用于多种操作系统的移动终端APP，实现了任务接收、故障信息上报、巡检人员位置记录等功能，提升了用户体验和操作便捷性。

**综合管理系统**：搭建了一个功能全面的综合管理系统，使管理人员能够便捷地查询巡检人员的工作状态、场地信息，并设置巡检路线、质量要求，以及发布临时巡检任务。

在经过调研主流巡检机器人，以及相关园区解决方案后，以下创新设计未被发现：

**AI大模型：通过车载NXP-imx8 作为主控实现边缘处理机器人、巡检小车具有10 Tops算力，结合数据网关AR502H-CN的容器能力，本地部署数据预处理本地离线大模型，通过ModelArts训练处理相关数据。与传统的训练、部署方式相比，没有做到大模型在工业、生产场景的的全场景覆盖。**

**边缘嵌入式场景下负载需求：不同于服务器端的高并发场景下，边缘计算水平对实时性与可靠性要求极高。**

**分布式架构的国产化实例：边缘计算在该种场景下，多OS的通讯不能单独依靠传统的单一协议实现多种通讯协议的交互与通讯，在国内虽然有不少通讯组织，但尚未形成能够轻松对接多种“端-边-云”产品的现象级实例。**

**云平台IOT接口能力调用：接入通过纵向对比，在对华为相关产品调研时，我们发现，华为云IOT平台对自家产品支持最佳。采用华为云IOT解决方案。**

## 1.3系统设计目标与解决的核心问题

本项目旨在对某石化工业园区的日常巡检工作进行深入调研，结合现场巡检情况和实际管理需求，以实现巡检工作的数字化转型为目标。为此，我们设计并开发了一套基于openEuler\_embedded操作系统的视觉巡检小车及系统，



图2 现有巡检与数通产品横纵向对比

该系统包括跨平台（Android、iOS、HarmonyOS）的统一NT架构TO-C 场景APP、5G移动基站通信技术，以及以综合管理系统为核心的智能巡检解决方案。系统设计的核心思想是通过数据采集设备自动采集和存储原本需要人工记录的数据，利用AR502H-CN边缘计算网关对数据预处理和5G移动通信技术将数据实时发送至综合管理系统，并通过车载小型低成本视觉识别和SLAM建图功能实时获取巡检场地中工作人员的位置信息，并对危险行为做出预警。

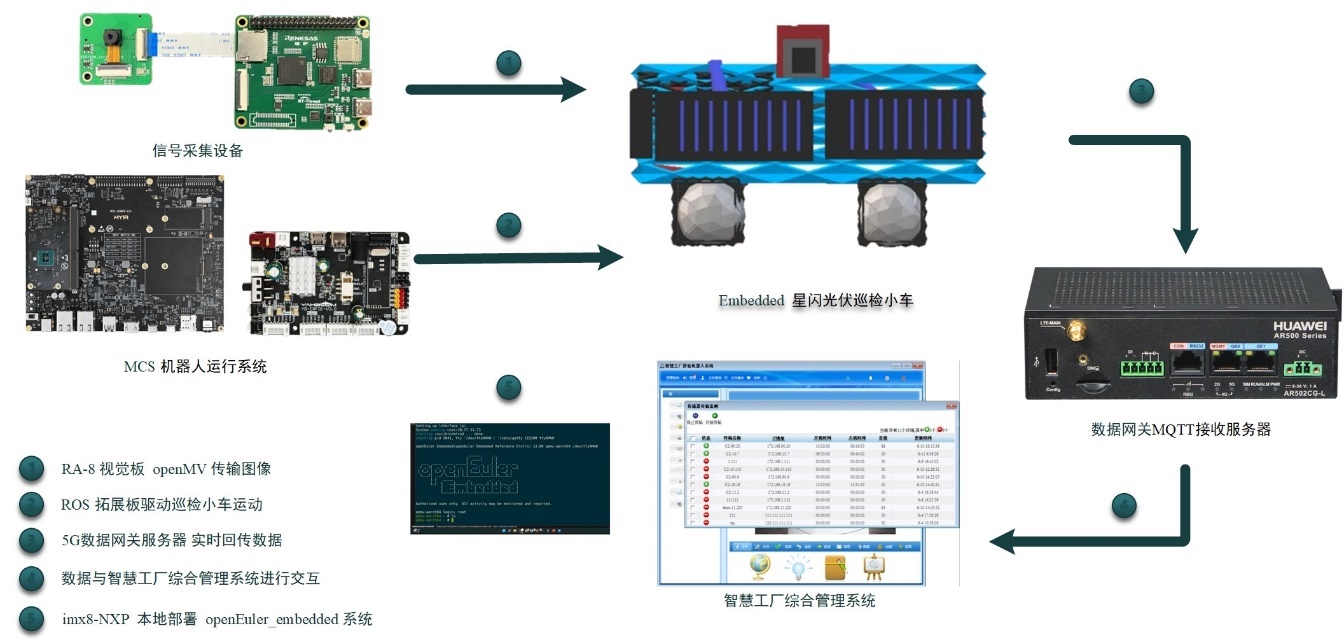


图3 AR502边缘计算低功耗视觉巡检系统

# 第二章 特色与创新

## 2.1 创新优势分析

### 2.1.1 技术创新点

本项目在智能巡检机器人领域实现了多项技术创新，具体包括：

1. **跨平台兼容性**：开发了一套支持Android、iOS、HarmonyOS等多端操作系统的统一NT架构APP，实现了不同设备间的无缝连接和操作一致性，提高了用户的便利性和系统的普及性。
2. **5G通信技术的集成应用**：利用5G移动基站通信技术，大幅提高了数据传输速度和稳定性，确保了巡检数据的实时性和准确性，同时降低了通信延迟，提升了系统响应速度。
3. **基于openEuler\_embedded的系统开发**：采用国产Linux系统发行版openEuler\_embedded作为机器人操作系统，增强了系统的安全性和可控性，同时支持了国产软件生态的发展。
4. **数据采集与智能处理**：设计了基于STM32微处理器的高性能数据采集设备，结合先进的视觉识别和SLAM建图技术，实现了对巡检环境的高精度感知和实时监控。
5. **综合管理系统的智能化**：构建了一个功能全面的综合管理系统，集成了任务下发、数据记录、报警信息查阅等功能，实现了对巡检工作的全面数字化管理。
6. **数据安全与隐私保护**：结合区块链和IPFS技术，实现了数据的去中心化存储和传输，保障了数据的不可篡改性和安全性。同时，采用SecGear可信计算框架对敏感数据进行加密处理，确保了数据的端到端安全。



图3 高稳定性多OS混合部署

### 2.1.2 应用创新点



图4 智能巡检角度的应用创新点

本项目在智能巡检机器人的应用方面也展现了创新性：

1. **现场作业的自动化**：通过自动化的数据采集和处理，减少了人工巡检的需求，降低了作业风险，提高了作业效率。
2. **实时监控与快速响应**：实现了对巡检现场的实时监控，结合5G通信技术，能够快速响应巡检中发现的异常情况，及时采取措施。
3. **智能路径规划与任务调度**：综合管理系统支持智能路径规划和任务调度，优化了巡检流程，提升了巡检工作的系统性和科学性。
4. **多维度数据分析与决策支持**：系统能够对采集到的多维度数据进行深入分析，为管理人员提供决策支持，帮助企业实现更精准的运维管理。
5. **环境适应性强**：智能巡检机器人设计考虑了多种复杂环境，具有良好的越障能力和环境适应性，能够满足不同场景下的巡检需求。
6. **赋能产业升级**：项目的实施推动了石化工业园区等传统产业的数字化、智能化升级，为产业转型提供了新的动力和方向。

通过这些特色与创新点，本项目不仅提升了智能巡检机器人的技术性能和应用效果，也为相关行业的数字化转型提供了有力的技术支撑和示范作用。

## 2.2创新设计

智能巡检小车驱动部分的微处理器采用STM32F103RCT6对全向巡检小车的移动进行PWM闭环控制。调用封装完成openMV视觉库的RA-8对图像数据进行采集与预处理，并通过串口通讯将采集处理后的数据分送至MQTT通信网关和MYD-JX8MP上位机。通讯模块采用EB25模组，将微处理通讯来的数据通过Nearlink\_SLB技术发送至部署综合管理系统的上位机imx8-NXP。通过AR502H-CNp配置边缘网关，稳定传输关键数据，提高IOT在石化、电力等行业的应用稳定性。

# 功能设计

## 3.1 功能模块概述

### 3.1.1数据采集与实时监控

**功能描述**：智能巡检机器人具备高精度数据采集功能，能够实时监控设备状态和环境参数，并通过5G网络将数据传输至综合管理系统。

**作用**：确保了巡检数据的实时性和准确性，为后续的数据分析和决策提供基础。

**选型设计：采用轻量化RA8-MCU的Rt-thread视觉开发板的解决方案，在小车上同时部署openMV来做视觉识别模型，通过传输视频流到上位机imx8-NXP 中**

### 3.1.2智能巡检路径规划

**功能描述**：系统能够根据巡检区域和任务要求，智能规划巡检路径，优化巡检流程。

**作用**：提高了巡检效率，减少了无效巡检和遗漏，确保了巡检工作的全面性和系统性。

### 3.1.3故障检测与报警

**功能描述**：机器人搭载的传感器和视觉识别系统能够检测设备异常和潜在故障，并实时发出报警。

**作用**：实现了对设备故障的早期发现和预警，降低了事故发生的风险。

### 3.1.4多端操作界面

**功能描述**：开发了适用于Android、iOS、HarmonyOS等多端操作系统的统一APP，提供用户友好的操作界面。

**作用**：提升了用户体验，使得不同设备用户都能便捷地接收任务、上报信息和管理巡检工作。



图5 多内核操作系统模型

### 3.1.5 综合管理系统

**功能描述**：构建了一个集成了任务下发、数据记录、报警信息查阅等功能的综合管理系统。

**作用**：实现了对巡检工作的全面数字化管理，为管理人员提供了强大的数据支持和决策工具。

### 3.1.6 安全与隐私保护

**功能描述**：结合区块链和IPFS技术，实现了数据的安全存储和传输，并采用SecGear框架对敏感数据进行加密处理。

**作用**：保障了数据的安全性和隐私性，防止了数据泄露和篡改，增强了系统的信任度。

### 3.1.7 环境适应性设计

**功能描述**：智能巡检机器人设计考虑了多种复杂环境，具备良好的越障能力和环境适应性。

**作用**：确保了机器人能够在不同场景下稳定工作，满足了多样化的巡检需求。

### 3.1.8 远程控制与协同作业

**功能描述**：支持远程控制机器人执行巡检任务，并能够实现多机器人协同作业。

**作用**：提高了巡检工作的灵活性和协同效率，适应了大规模巡检作业的需求。

通过上述功能设计，本项目不仅满足了石化工业园区等特定场景下的巡检需求，而且通过技术创新和智能化管理，提升了巡检工作的效率和质量，为相关行业的数字化转型提供了有力的支持。

## 3.2多端操作界面

在智能巡检机器人系统中，多端操作界面的设计至关重要，因为它直接影响到操作人员的用户体验和系统的易用性。以下是对多端操作界面的详细介绍：

### 3.2.1 多平台兼容性

设计目标：确保操作界面在不同的设备和操作系统上都能正常工作，包括桌面电脑、笔记本电脑、平板电脑和智能手机。

实现方法：采用响应式设计和跨平台开发框架，如React Native或Flutter，以实现代码的一次编写，多端运行。

1. 安装Flutter SDK、设置IDE
2. 设计软件UI
3. 进行统一代码封装

### 3.2.2 用户体验（UX）设计

设计目标：提供直观、易用的操作界面，减少用户的学习成本。

实现方法：进行用户研究，了解操作人员的需求和习惯，设计符合直觉的操作流程和界面元素。

1. 使用Figma进行UI交互设计
2. 采用商用协议开源组件库
3. 做好用户设计,方便用户

### 3.2.3 JVM的JAVA程序源码设计

功能描述：开发运行在Java虚拟机（JVM）上的Java应用程序，用于控制和管理机器人。

**实现方法：**

1. 控制逻辑：编写代码实现机器人的基本控制功能，如启动、停止、移动、转向等。
2. 用户界面：设计和实现用户界面，提供机器人状态显示、控制命令输入等功能。
3. 后端服务：开发后端服务，处理数据存储、业务逻辑、与机器人通信等任务。

## 3.3综合管理系统

### 3.3.1 前端设计

设计目标：创建一个直观、易用的用户界面，提供机器人控制和数据展示功能。

实现方法：

1. 界面布局：设计清晰的界面布局，将常用的控制按钮和数据显示放在显眼位置。
2. 交互设计：实现流畅的交互效果，如按钮点击反馈、滑动操作等。
3. 数据可视化：采用图表、地图等可视化手段展示机器人的巡检数据和状态。
4. 对工厂进行建图，实时SLAM建图重合，确认机器人平面设备图纸绑定
5. 对工人进行感知，进行交互

### 3.3.2 后端设计

设计目标：构建一个稳定、安全、可扩展的后端系统，支持机器人控制和数据处理。

数据存储：选择合适的数据库系统，如MySQL、MongoDB等，存储机器人的巡检数据和状态信息。

业务逻辑：编写代码实现业务逻辑，如任务调度、数据分析、异常处理等。

通信接口：开发与机器人通信的接口，使数据稳定传达和数据的实时传输。

安全性：实施安全措施，如数据加密、访问控制、防止SQL注入等，保护系统和数据的安全。

### 3.3.3 系统架构

设计目标：构建一个模块化、可扩展的系统架构，支持功能扩展和维护。

模块化设计：将系统分为多个模块，如用户管理、任务调度、数据存储等，便于单独开发和维护。

服务化架构：采用微服务架构，将不同的业务逻辑部署为独立的服务，提高系统的灵活性和可扩展性。

容器化部署：使用Docker等容器技术，实现系统的快速部署和扩展。

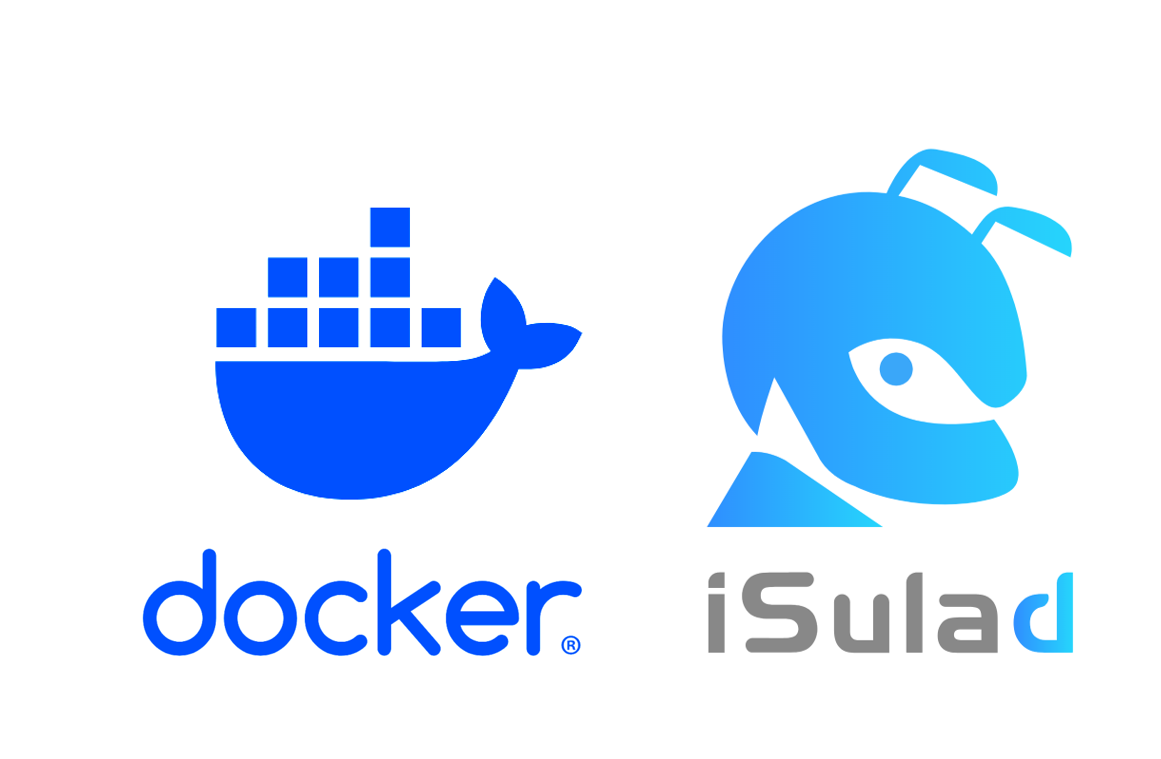


图6 Docker、iSulad等容器技术

# 系统实现

## 4.1物联网技术架构

### 4.1.1感知层

数据采集：

openMV\openCV视觉识别技术用于实时捕获巡检场地信息和人员位置

### 4.1.2传输层

5G移动基站通信技术  
通信模块：  
巡检小车配备5G通信模块，与数据网关建立高速、低延迟的数据传输通道  
搭建Nearlink抗干扰网络，使用开发板EB25进行BLE透传  
使用esp32搭建微型MQTT服务，多链路传输  
  
数据传输：  
将采集到的视频、图像、位置信息以及环境数据通过5G网络发送到数据网关  
USB串口连接上位机imx8-NXP,将部分数据封装后集成向AR502H-CN传输

1. 上位机IMX8-NXP的编译过程

1.配置本地虚拟机环境

cd ~ vim .bashrc

2.加入bitbake环境编译变量

1. $ export PATH=//home/darrenpig/yocto-poky/bitbake/bin:$PATH

2. $ bitbake --version

3. BitBake Build Tool Core version 1.50.0

3.保存bitbake 环境镜像的编译方式

1. //保存文件并退出 Vim 编辑器。

2. source ~/.bashrc

3. //注：如果不执行 source 命令，则需重启系统才能生效

4.测试bitbake工具是否可以编译

1. mkdir $HOME/hello

2. cd $HOME/hello

3. bitbake

4. BBPATH="$HOME/hello"

5. export BBPATH

6. bitbake

7. conf/bitbake.conf

5. bitbake根据找到的conf文件路径自动生成TOPDIR变量

1. // 临时变量，为了方便配置，也可以不设置

2. TMPDIR = "${TOPDIR}/tmp"

3. // 指定BitBake用于存储元数据缓存的目录，这样就不需要每次启动BitBake时都对其进行解析。

4. CACHE = "${TMPDIR}/cache"

5. // 指定用于创建配方戳文件的基本路径。实际戳文件的路径是通过对该字符串求值，然后附加附加信息来构建的。

6. STAMP = "${TMPDIR}/${PN}/stamps"

7. // 指向BitBake在构建特定配方时放置临时文件的目录，这些文件主要由任务日志和脚本组成。

8. T = "${TMPDIR}/${PN}/work"

5. bitbake 配置存储信息

1. for usage information.

2. //构建目标配方：绑定源

3. //下载“tree”查看结构

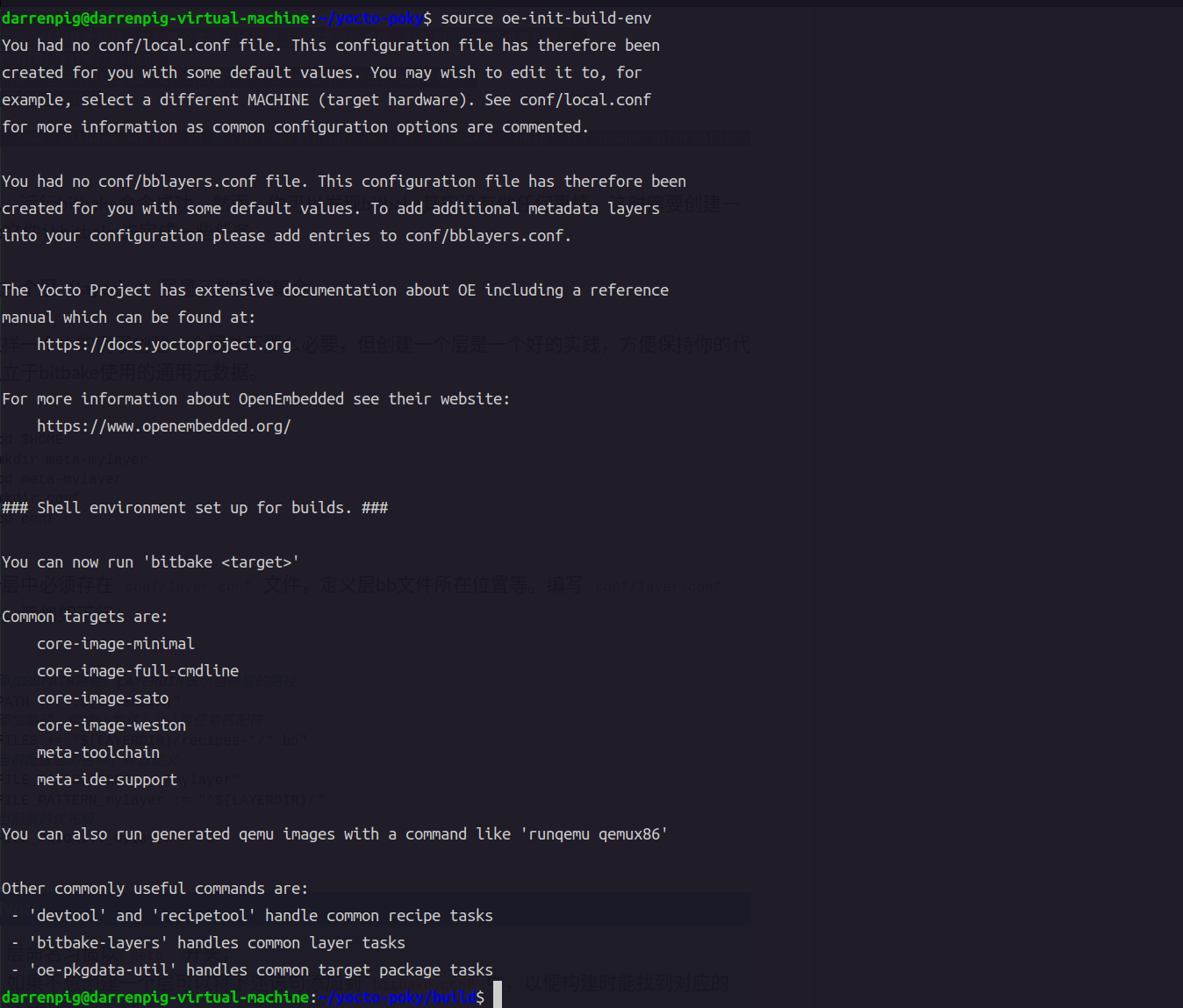


图7 虚拟机的bitbake编译环境配置

1. isulad 国产容器部署

1. /etc/isula/daemon.json

2. registry-mirros``docker.io``

3. $ isula image

4. $ isula run -it busybox sh 3

5. exit 退出容器

6. $ isula pull openEuler/openEuler? 23.04

7. $ isula run -it -net = host openEuler/openEuler :23.04 sh

8. sh2# dnf install ipuild

9. sh2$ ping www.baidu.com

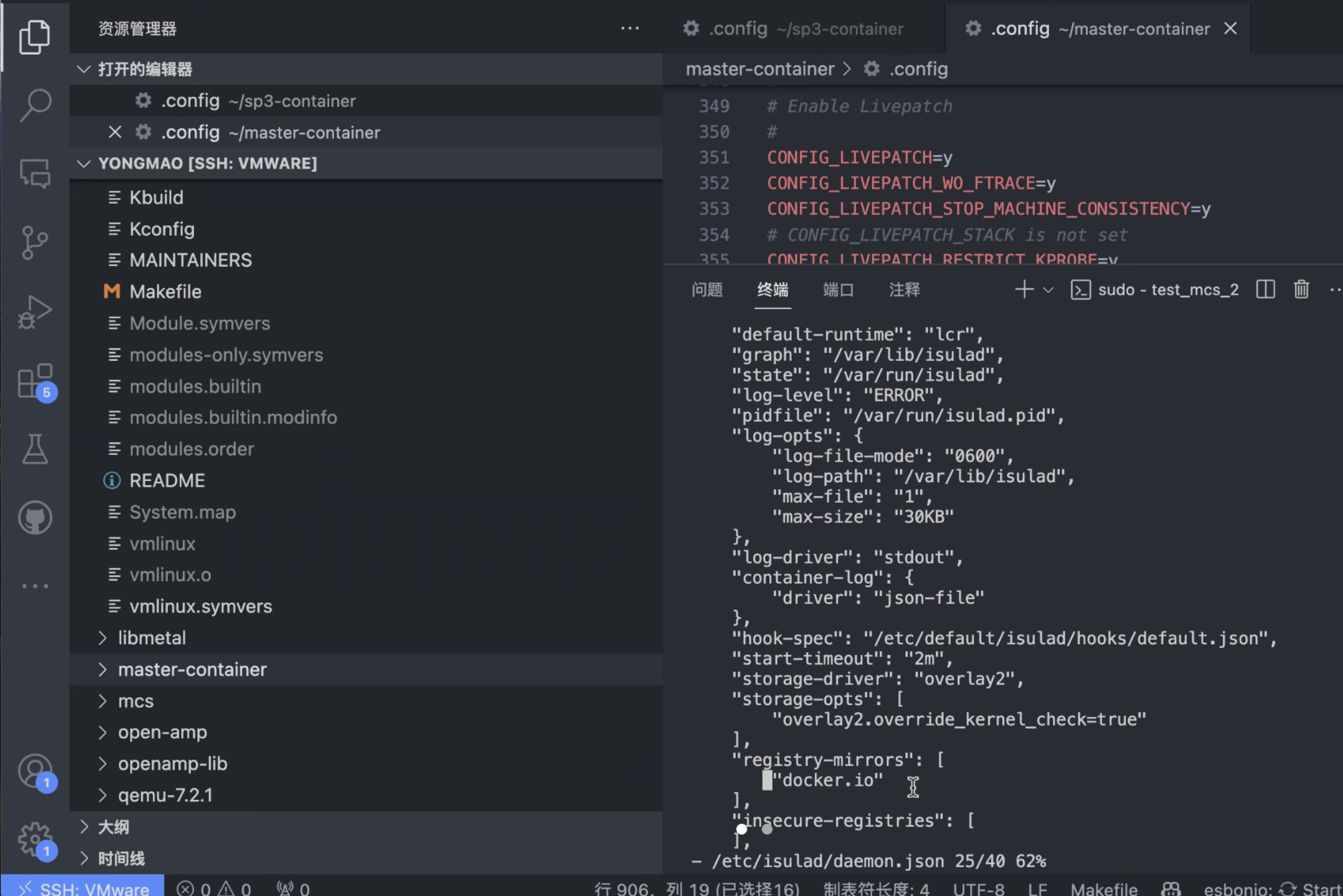


图8 虚拟机的isulad的配置

### 4.1.3控制层

综合管理系统  
  
 部署环境：  
 数据网关AR502H-CN容器部署的综合管理系统  
  
 功能实现：  
 接收来自巡检小车的实时数据  
 数据处理与分析（如人员位置跟踪、环境数据监测等）  
 巡检任务管理（如设置巡检路线、巡检质量要求、发布临时巡检任务等）  
 与其他系统（如ERP、MES等）集成，实现数据共享

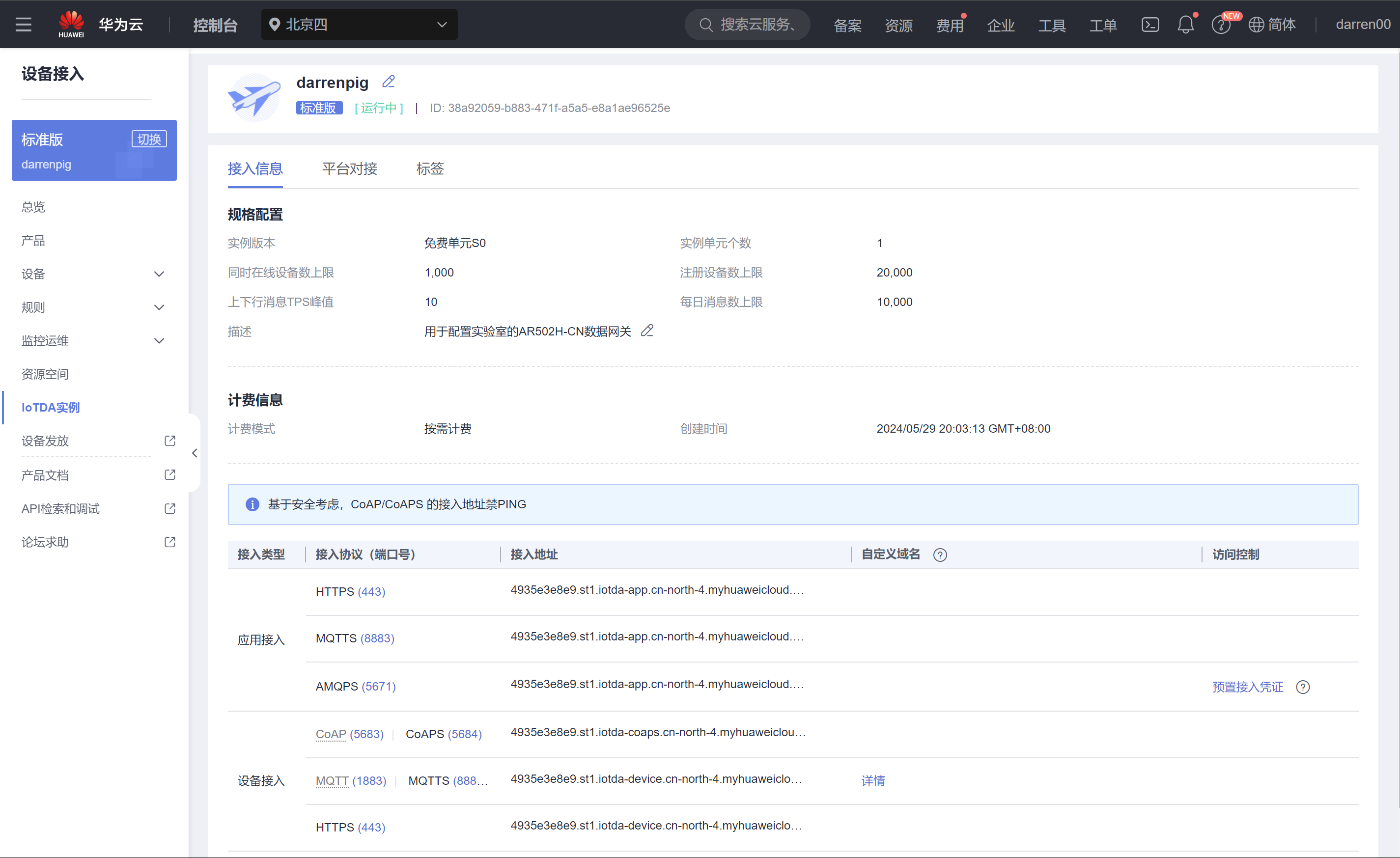


图9 华为云IOT应用接口使用

### 4.1.4软件开发技术

一．移动应用技术开发（APP反应监测数据）工具：Android Studio

　　 1. 定义网关AR502H-CN的数据接口

　　 协议：确定使用的通信协议（如HTTP、MQTT等）。

　　 数据格式：网关发送的数据格式（如JSON、XML等）。

　　 API文档：网关提供的API文档，包含数据获取的具体方法和参数。

　　 2. 设置Android项目

　　 a. 配置网络权限

AndroidManifest.xml中添加网络权限，以允许应用进行网络操作：

　<uses-permission android:name="android.permission.INTERNET"/>

b. 添加网络库依赖

建议使用OkHttp或Retrofit进行网络请求。以OkHttp为例，添加OkHttp依赖到build.gradle文件中：

1. dependencies {

2. 　　 implementation 'com.squareup.okhttp3:okhttp:4.9.3'

3. }

　　 3. 实现数据请求

　　 a. 发送网络请求

　　 使用OkHttp库发送HTTP请求获取数据：

b. 创建OkHttpClient实例：

　 OkHttpClient client = new OkHttpClient();

构建请求：

1. Request request = new Request.Builder()

2. 　　 .url("http://your-gateway-address/api/data") // 替换为实际的API地址

3. 　　 .build();

4. 　　执行请求并处理响应：

5.

6. 　　client.newCall(request).enqueue(new Callback() {

7. 　　 @Override

8. 　　 public void onFailure(Call call, IOException e) {

9. 　　 e.printStackTrace();

10. 　　 // 处理请求失败

11. 　　 }

12.

13. 　　 @Override

14. 　　 public void onResponse(Call call, Response response) throws IOException {

15. 　　 if (response.isSuccessful()) {

16. 　　 String responseData = response.body().string();

17. 　　 // 处理成功响应

18. 　　 processData(responseData);

19. 　　 } else {

20. 　　 // 处理请求错误

21. 　　 }

22. 　　 }

23. 　　});

24.

　　4. 处理和解析数据

假设网关返回的是JSON格式的数据，可以使用Gson库进行解析。首先，添加Gson依赖到build.gradle中：

1. 　dependencies {

2. 　　 implementation 'com.google.code.gson:gson:2.10'

3. 　　}

4. 　　定义数据模型：

5.

6. 　　public class DataModel {

7. 　　 private String field1;

8. 　　 private int field2;

9. 　　 // 其他字段及其getter和setter

10. 　　}

11. 　　解析数据：

12.

13. 　　private void processData(String jsonData) {

14. 　　 Gson gson = new Gson();

15. 　　 DataModel dataModel = gson.fromJson(jsonData, DataModel.class);

16. 　　 // 使用解析后的数据

17. 　　}

　　5. 更新UI

　　在onResponse方法中更新UI时，需要在主线程中执行，因为UI操作不能在子线程中进行。使用runOnUiThread方法来更新UI：

1. 　public void onResponse(Call call, Response response) throws IOException {

2. 　　 if (response.isSuccessful()) {

3. 　　 String responseData = response.body().string();

4. 　　 // 处理数据

5. runOnUiThread(() -> {

6. 　　　 processData(responseData);

7. 　　 // 更新UI组件，例如：

8. 　　 // textView.setText(dataModel.getField1());

9. 　　 });

10. 　　 } else {

11. 　　 // 处理请求错误

12. 　　 }

13. 　　}

　　6. 错误处理和优化

　　网络错误：处理网络请求错误，提供用户友好的错误信息。

　　数据处理：确保对接收到的数据进行验证和处理，避免数据格式不一致或损坏。

性能优化：优化网络请求和数据处理的性能，避免阻塞主线程，减少网络延迟。

　　7. 测试和调试

　　功能测试：确保应用能够正确地接收和处理网关的数据。

　　兼容性测试：测试不同的设备和网络条件下的应用表现。

错误日志：记录和分析错误日志，修复潜在的问题。

二．设置Android项目

　　1. 设置项目基本信息

　　 a. 创建或打开项目

　　创建项目：打开Android Studio，选择“Start a new Android Studio project”来创建一个新项目。

打开项目：选择“Open an existing Android Studio project”来打开一个已有项目。

　　 b. 配置项目文件

　 build.gradle（项目级别）：

　　 路径：<项目根目录>/build.gradle

　　 用于配置项目级别的Gradle设置，如插件和依赖项的版本。

1. 　buildscript {

2. 　　 repositories {

3. 　　 google()

4. 　　 mavenCentral()

5. 　　 }

6. 　　 dependencies {

7. 　　 classpath "com.android.tools.build:gradle:8.1.0"

8. 　　 }

9. 　　}　　build.gradle（模块级别）：

路径：<项目目录>/app/build.gradle

　　用于配置模块（如应用模块）的Gradle设置，如依赖项和插件。

示例：

1. plugins {

2. 　　 id 'com.android.application'

3. 　　}

4.

5. 　　android {

6. 　　 compileSdk 33

7. 　　 defaultConfig {

8. 　　 applicationId "com.example.myapp"

9. 　　 minSdk 21

10. 　　 targetSdk 33

11. 　　 versionCode 1

12. 　　 versionName "1.0"

13. 　　 }

14. 　　 buildTypes {

15. 　　 release {

16. 　　 minifyEnabled false

17. 　　 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'

18. 　　 }

19. 　　 }

20.

21. 　　　}

22.

23. 　　dependencies {

24. 　　 implementation 'androidx.core:core-ktx:1.9.0'

25. 　　 implementation 'androidx.appcompat:appcompat:1.6.1'

26. 　　 implementation 'com.google.android.material:material:1.9.0'

27. 　　 implementation 'androidx.constraintlayout:constraintlayout:2.1.4'

28. 　　 testImplementation 'junit:junit:4.13.2'

29. 　　 androidTestImplementation 'androidx.test.ext:junit:1.1.5'

30. 　　 androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

31. 　　 implementation 'com.squareup.okhttp3:okhttp:4.9.3'

32. 　　 implementation 'com.google.code.gson:gson:2.10'

33. 　　}

34.

2. 配置AndroidManifest.xml

　　路径：<项目目录>/app/src/main/AndroidManifest.xml

　　用于声明应用的组件（如活动、服务、广播接收器），权限请求，以及应用的基本设置。

示例：

1. <manifest xmlns:android="http://schemas.android.com/apk/res/android"

2. 　　 package="com.example.myapp">

3.

4. 　　 <application

5. 　　 android:allowBackup="true"

6. 　　 android:icon="@mipmap/ic\_launcher"

7. 　　 android:label="@string/app\_name"

8. 　　 android:roundIcon="@mipmap/ic\_launcher\_round"

9. 　　 android:supportsRtl="true"

10. 　　 android:theme="@style/Theme.MyApp">

11. 　　 <activity android:name=".MainActivity">

12. 　　 <intent-filter>

13. 　　 <action android:name="android.intent.action.MAIN" />

14. 　　 <category android:name="android.intent.category.LAUNCHER" />

15. 　　 </intent-filter>

16. 　　 </activity>

17. 　　 </application>

18.

19. 　　　　 <uses-permission android:name="android.permission.INTERNET"/>

20. 　　</manifest>

　　3. 配置Gradle插件和依赖

　　 在项目级别build.gradle中配置Gradle插件和存储库。

　 在模块级别build.gradle中配置项目的依赖项和插件，如OkHttp和Gson等网络库。

　　4. 设置网络权限

　　 位置：AndroidManifest.xml

　　 必须在AndroidManifest.xml中添加网络权限以允许应用进行网络操作。

　　<uses-permission android:name="android.permission.INTERNET"/>

　　5. 设置构建配置

构建类型：在模块级别build.gradle文件中配置构建类型（如debug、release）。

版本控制：配置versionCode和versionName，并设置minSdkVersion和targetSdkVersion。

6. 调试和测试

日志：使用Logcat来查看调试输出和日志信息。

模拟器/设备：在Android Studio中设置并运行模拟器，或连接实际设备进行测试。

7. 同步项目

　　在修改build.gradle文件后，需要点击“Sync Now”按钮同步项目，以应用新的Gradle设置。

三．设计网页反映监测数据 （前端后端）工具：visual studio

　　1. 设置开发环境

　　 a. 安装必要的软件

　　Visual Studio：首先安装Visual Studio

　　Java Development Kit (JDK)：安装JDK 11或更高版本。

　　Node.js 和 npm：安装Node.js（包含npm）用于管理JavaScript依赖。

　　Vue CLI：安装Vue CLI用于创建和管理Vue.js项目。

　　npm install -g @vue/cli // 终端安装Vue CLI

2. 创建后端服务（Java）

　　在Visual Studio中使用Java开发Spring Boot应用程序来处理后端逻辑。

　　 a. 安装Java插件

　　在Visual Studio中安装Java Extension Pack插件，这将提供Java开发所需的工具。

　　 b. 创建Spring Boot项目

　　在Visual Studio中创建一个新的Java项目，选择Spring Boot模板。

　　创建项目：

　　打开Visual Studio。

　　选择 File -> New -> Project。

　　搜索并选择 Spring Boot 项目模板。

　　添加依赖：

　　打开 pom.xml 文件，并添加必要的依赖，如下所示：

1. 　<dependencies>

2. 　　 <dependency>

3. 　　 <groupId>org.springframework.boot</groupId>

4. 　　 <artifactId>spring-boot-starter-web</artifactId>

5. 　　 </dependency>

6. 　　 <dependency>

7. 　　 <groupId>com.fasterxml.jackson.core</groupId>

8. 　　 <artifactId>jackson-databind</artifactId>

9. 　　 </dependency>

10. 　　</dependencies>

　　创建REST控制器：

　　在 src/main/java/com/example/demo 目录下创建一个控制器类：

1. import org.springframework.web.bind.annotation.GetMapping;

2. 　　import org.springframework.web.bind.annotation.RequestMapping;

3. 　　import org.springframework.web.bind.annotation.RestController;

4. 　　import org.springframework.web.client.RestTemplate;

5.

6. 　　@RestController

7. 　　@RequestMapping("/api/data")

8. 　　public class DataController {

9.

10. 　　 private final RestTemplate restTemplate = new RestTemplate();

11.

12. 　　 @GetMapping

13. 　　 public String getData() {

14. 　　 String url = "http://example.com/api/data"; // AR502H-CN 的数据接口 URL

15. 　　 return restTemplate.getForObject(url, String.class);

16. 　　 }

17. 　　}

　　运行Spring Boot应用：

　　使用Visual Studio的运行功能启动你的Spring Boot应用。

　　3. 创建前端应用（Vue.js）

　　在Visual Studio中可以使用外部工具来创建和管理Vue.js项目，但你可以在Visual Studio中进行集成。

　　 a. 创建Vue项目

　　打开终端：

　　在Visual Studio中打开终端窗口（可以使用 Terminal 或 Cmd）。

　　创建项目：

　　vue create my-vue-app

　　进入项目目录：

　　cd my-vue-app

　　安装Axios：

　　npm install axios

　　 b. 开发Vue组件

创建一个Vue组件：

在 src/components 目录下创建一个新组件 DataDisplay.vue：

1. 　　<template>

2. 　　 <div>

3. 　　 <h1>AR502H-CN 数据</h1>

4. 　　 <div v-if="data">

5. 　　 <p>数据：{{ data }}</p>

6. 　　 </div>

7. 　　 <div v-else>

8. 　　 <p>加载中...</p>

9. 　　 </div>

10. 　　 </div>

11. 　　</template>

12.

13. 　　<script>

14. 　　import axios from 'axios';

15.

16. 　　export default {

17. 　　 data() {

18. 　　 return {

19. 　　 data: null

20. 　　 };

21. 　　 },

22. 　　 mounted() {

23. 　　 this.fetchData();

24. 　　 },

25. 　　 methods: {

26. 　　 async fetchData() {

27. 　　 try {

28. 　　 const response = await axios.get('http://localhost:8080/api/data');

29. 　　 this.data = response.data;

30. 　　 } catch (error) {

31. 　　 console.error('获取数据失败', error);

32. 　　 }

33. 　　 }

34. 　　 }

35. 　　};

36. </script>

37. 　　</style>

在 vue.config.js 中配置代理以转发API请求（如果前端和后端在不同端口上运行）：

1. module.exports = {

2. 　　 devServer: {

3. 　　 proxy: 'http://localhost:8080'

4. 　　 }

5. };

运行Vue应用：

npm run serve

　　4. 整合前后端

　　确保后端服务正在运行，并且Vue应用能够正确地从后端获取数据。你可以在浏览器中访问Vue应用的地址（通常是 http://localhost:8080），检查是否正确显示数据。

　　5. 部署和发布

　　后端：将Spring Boot应用部署到一个服务器上，例如AWS、Heroku等。

前端：构建Vue应用并将其部署到静态文件服务器上，例如Netlify、Vercel　　后端：使用Spring Boot处理从AR502H-CN获取的数据，并将其提供给前端。

　　前端：使用Vue.js展示从后端获取的数据。

　　Visual Studio：可以用来开发Java项目，前端Vue.js项目可以在Visual Studio中管理，但主要开发工作可以在外部工具中完成。

　四．网关获取小车数据

　　1. 配置硬件设备

　　安装传感器：在小车上安装适当的传感器（例如温度传感器、湿度传感器、加速度计等）。

　　连接模块：将传感器通过相应的无线通讯模块（如ZigBee、蓝牙、NB-IoT模块等）连接到小车上。

　　电源供应：确保小车有稳定的电源供应，以供传感器和通讯模块工作。

　2. 配置小车通信

　　设置通信协议：确保小车上的无线模块配置为与AR502H-CN网关兼容的协议。例如，如果使用的是ZigBee协议，则确保小车上的ZigBee模块和AR502H-CN上的ZigBee接口都兼容相同的通讯标准。

　　地址分配：为小车上的每个无线模块分配唯一的地址。

　3. 配置网关AR502H-CN

　　接入网络：确保AR502H-CN网关已经接入互联网或本地企业网络。

无线网络配置：在AR502H-CN网关上配置相应的无线网络设置，以接收小车发送的数据。

通信设置：在AR502H-CN网关的配置界面上，选择正确的无线接入技术类型（如ZigBee、蓝牙、NB-IoT等），输入小车无线模块的地址，进行设备绑定和数据接收设置。

　4. 设置数据传输

　　数据格式：确保小车发送的数据格式符合AR502H-CN网关的数据接收格式。

数据通道：设置数据传输通道，比如通过MQTT协议将数据发送到云端或者本地服务器。

　5. 测试数据接收

　　连接确认：连接好小车和小车上的模块esp32、EB25，并U-boot启动系统。

数据捕捉：在AR502H-CN网关的监控界面观察数据接收情况，确认小车是否成功发送数据以及网关是否成功接收数据。

　6. 数据处理和存储

　　数据解析：对AR502H-CN边缘计算网关接受的数据进行解析，提取有用信息。

　　数据存储：将解析后的数据存储在本地数据库或云服务，进行后续分析。

　　注意事项：

　　确保所有模块和设备之间的兼容性，例如，使用互P2P连接协议或网状网络协议在设备间建立通信。

　　对于安全性的要求，可能需要在网络层进行额外配置，确保数据传输安全。

如果涉及跨地域或WAN环境下应用，可能还需要考虑网络延迟和带宽限制，优化数据的传输策略。

## 4.2数据挖掘和可视化

**技术描述**：数据挖掘和分析技术用于从大量巡检数据中提取有价值的信息，为决策提供支持。

**实现方式**：应用机器学习和数据挖掘算法，对采集到的数据进行深入分析，识别设备运行的规律和潜在问题，生成报告和建议。

### 4.2.1ModelArts

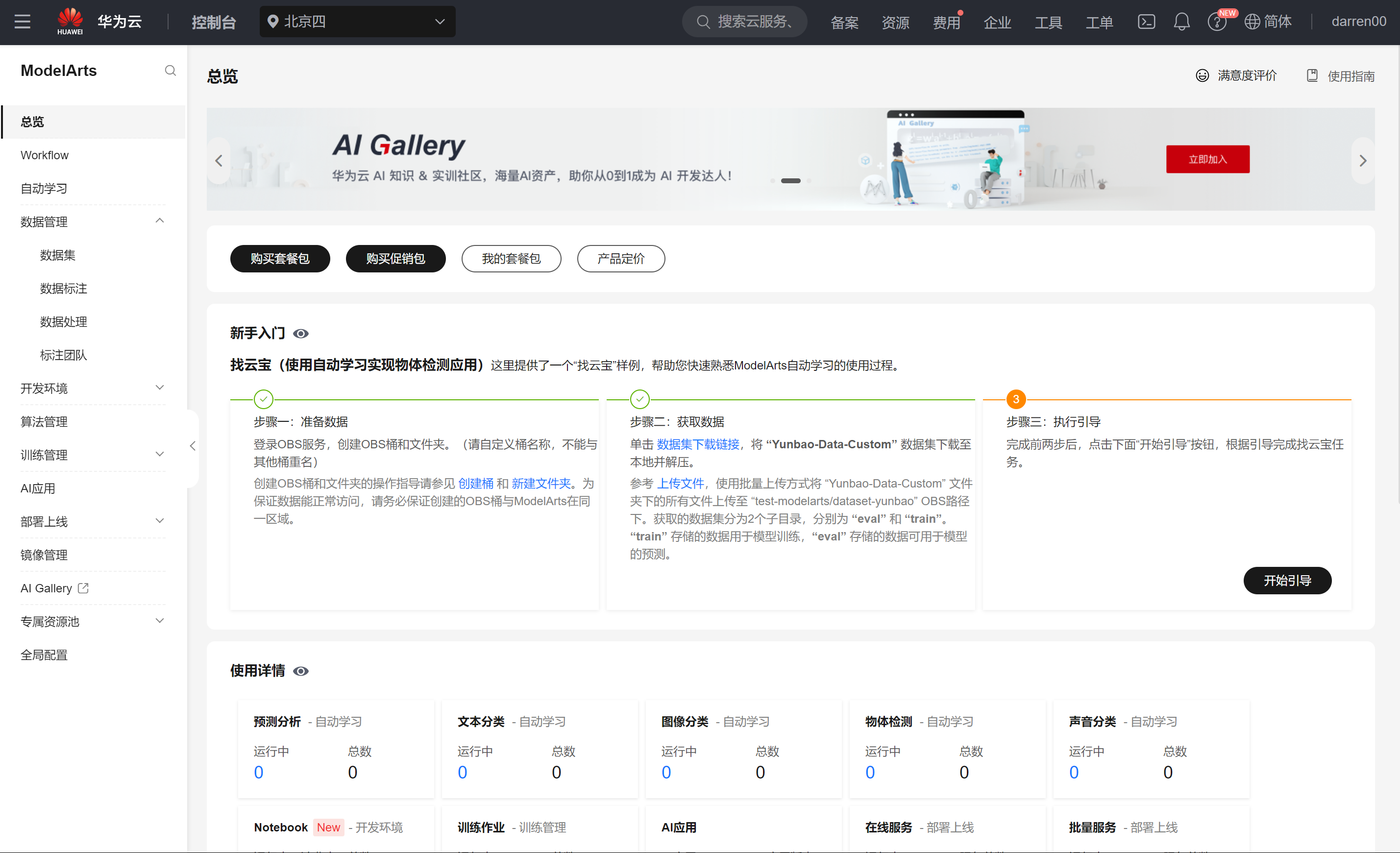
ModelArts是面向开发者的一站式AI开发平台，为机器学习与深度学习提供海量数据预处理及半自动化标注、大规模分布式Training、自动化模型生成，及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期AI工作流。

图10 华为云IOT的ModelArts应用界面

### 4.2.2Astro大屏应用 Astro Canvas

华为云Astro大屏应用Astro Canvas，以数据可视技术为核心，以屏幕轻松编排，多屏适配可视为基础，帮助非专业开发者通过图形化界面轻松搭建专业水准的数据可视化大屏应用，满足项目运营管理，业务监控，风险预警等多种业务场景下的一站式数据实时可视化大屏展示需求。

# 其他内容

## 5.1 工业设计与成本计算

### 5.1.1 工业设计

**设计理念**：智能巡检机器人的工业设计注重人体工程学、操作便捷性与环境适应性，确保机器人在各种工作条件下都能稳定运行。

**设计要素**：包括机器人的外形设计、色彩方案、材料选择等，以满足不同工业环境的审美和功能需求。

**用户交互**：工业设计考虑了用户与机器人交互的直观性和易用性，确保操作人员能够快速熟悉并有效使用机器人。

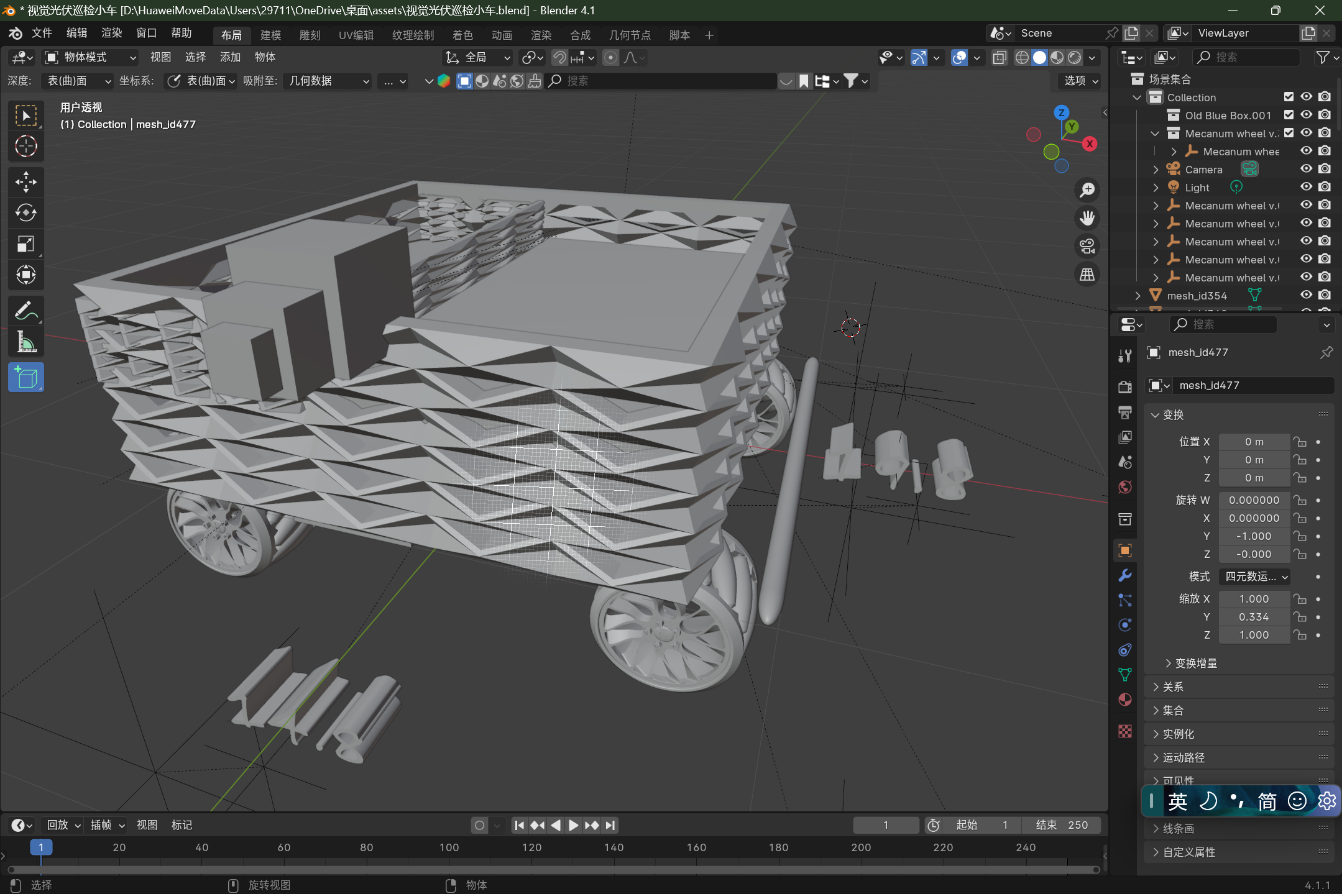


图11 小车搭载巡检系统和数据网关的部署工业设计

### 5.1.2 成本计算

**成本构成**：智能巡检机器人的成本计算包括研发成本、材料成本、生产成本、人力成本以及后期的维护和升级成本。

**成本控制**：通过优化设计、批量采购和提高生产效率等措施，有效控制成本，确保项目的经济效益。

**经济性分析**：对比传统人工巡检与智能巡检机器人的成本效益，评估机器人在长期使用中的成本节约和投资回报。

### 5.1.3 法规遵从性

**法规要求**：确保智能巡检机器人的设计和应用遵循相关行业标准和法律法规。

**合规性测试**：进行必要的测试和认证，以证明机器人满足安全、健康和环保等方面的规定。

### 5.1.4 市场推广策略

**目标市场**：明确智能巡检机器人的市场定位，识别潜在客户群体和市场需求。

**推广计划**：制定市场推广计划，包括宣传材料、展示活动、用户培训和售后服务等。

### 5.1.5 风险管理

**风险识别**：识别项目实施过程中可能遇到的风险，包括技术风险、市场风险、财务风险等。

**风险应对**：制定风险应对措施，包括风险预防、监控和缓解策略。

通过综合考虑工业设计、成本计算、环境影响、法规遵从性、市场推广和风险管理等其他实现要素，本项目能够确保智能巡检机器人的全面性和可行性，为项目的成功实施和市场推广打下坚实基础。

# 参考文献

1. 毛建旭,贺振宇,王耀南,张辉,钟杭,易俊飞,陶梓铭,陈诺天.电力巡检机器人路径规划技术及应用综述[J].控制与决策,2023,38(11):3009-3024
2. 高春艳,陶渊,吕晓玲,张明路.非结构化环境下巡检机器人环境感知技术研究综述[J].传感器与微系统,2023,42(4):10-1318
3. 陈现森,徐辰.基于openGauss的异构算子加速技术[J].华东师范大学学报:自然科学版,2023(5):90-99.
4. 邓军, 李鑫, 王凯, et al. 2024. 矿井火灾智能监测预警技术近20年研究进展及展望. 煤炭科学技术 [J], 52: 154-177.
5. 孙瑞江,池威威,李树荣等. 雄安新区的数字孪生变电站应用及探索[J]. 现代电力,2024,41(1):29-35. DOI:10.19725/j.cnki.1007-2322.2022.0215.(张明路, 2023)
6. GRZESIK P, MROZEK D. Combining Machine Learning and Edge Computing: Opportunities, Challenges, Platforms, Frameworks, and Use Cases [J/OL] 2024, 13(3):640