Robot Project Design Document

2025-08-11

Robot Project Design Document

2025-08-11

Features

- Web dashboard: http://rpi-robot/status
  - Video feed. CPU temperature
  - System uptime
- Nginx: Web-server t/ v-in a Docker container
  - (video-dashboard) HTTP
- netstatus: runs in another container, exposes API endpoint on HTTP (/status) dashboard display
- Dashboard content is HTML, JanvaScrript
- Config: /home/pi/robot-project/

Planned Features

- Additional telemetry furfanclly for web dashboard
  - CPU usage, memory usage, disk usage
  - Disk usage
- Implementa HTTP over HTTPS

Table of Contents

1. Introduction

This document provides the design details of the Robot Project as of 2025-08-11. It covers the system architecture, hardware/software setup, features built to date, planned implementations, and detailed references to hostnames, ports, and file locations for development and maintenance.

2. System Overview

The robot is based on a Quantum J4 electric wheelchair power base with a RoboClaw motor controller and a Raspberry Pi 5 (RPi-Robot) running Ubuntu and Docker. The system is containerized for modularity and uses Nginx as a web server to host a video/telemetry dashboard. Planned upgrades include additional sensors, remote control integration, and AI/ML capabilities.

3. Current Features

- Web dashboard hosted at: http://rpi-robot/status
- Live video feed from onboard USB camera via `web-video-server` container
- CPU temperature and system uptime telemetry via `netstatus` container
- Nginx reverse proxy in `video-dashboard` container
- Static configuration stored in /home/pi/robot-project/
- Container stack built and managed with docker-compose

4. Planned Features

- Additional telemetry: CPU %, memory usage, disk usage
- Persistent Nginx configuration mounted from repo
- HTTPS support for dashboard
- RoboClaw motor control integration
- Remote control via FlySky FS-i6 receiver
- AI/ML processing for object recognition and autonomous navigation

5. System Specifications

Hostnames:

- RPi-Robot (primary robot controller)
- RPi-Frank (travel NAS, for later sync)

Primary OS: Ubuntu 64-bit on Raspberry Pi 5

Docker containers:

- video-dashboard (Nginx + frontend)
- web-video-server (ROS2 video streaming)
- netstatus (Python Flask API for telemetry)

Ports:

- 8080: web-video-server
- 8081: video-dashboard HTTP access
- 5000: netstatus API endpoint

6. Wiring Diagram

This section will include RoboClaw-to-motor, power, and FS-i6 receiver wiring diagrams. (To be inserted)

7. Container Stack Diagram

This section will include a diagram showing all containers, their interconnections, and port mappings. (To be inserted)