Czech technical university in Prague
Faculty of Electrical Engineering

Department of computer science
Open informatics
Software engineering

Development of control system and handling interface for automated guided vehicle

MASTER THESIS

Author: Bc. Jan Holub
Supervisor: Ing. Jan Chudoba
Year: 2025

Acknowledgements

Many thanks to Ing. Jan Chudoba for the thesis supervision and Ing. Jan Novák for valuable lessons regarding the system development.

Bc. Jan Holub

Title:
Development of control system and handling interface for automated guided vehicle

Author: Bc. Jan Holub

Study program: Open informatics

Field of study: Software engineering

Thesis type: Master thesis

Supervisor: Ing. Jan Chudoba
CTU CIIRC

Consultant: Ing. Jan Novák
Leuze Engineering Czech s.r.o.

Abstract: A web-based AGV control application enables teleoperation in manual, automatic, and autonomous modes, leveraging ROS 2, .NET, and Docker. It integrates ROS 2 messages with a SignalR API for real-time communication and uses Docker to manage ROS 2 container lifecycles, ensuring modularity and rapid deployment. Featuring a two-hand control joystick device and other Blazor components, the intuitive interface supports diverse AGV control use cases. Built with clean architecture, the application is easily adaptable for future enhancements and varied robot con gurations, enhancing exibility and e ciency in industrial automation.

Key words: automated guided vehicle (AGV), web Application, Robot Operating System (ROS), Docker, clean architecture, microservices, SignalR, Blazor, programmable logic controller (PLC), two-hand control device (THCD), real-time communication

Abstract (CZ): Webová aplikace pro °ízení AGV umoº¬uje teleoperaci v manuálním, automatickém a autonomním ovládacím reºimu pomocí ROS 2, .NET a Dockeru. Aplikace integruje zprávy ROS 2 do SignalR API pro komunikaci v reálném £ase a vyuºívá Docker pro správu ºivotního cyklu ROS 2 kontejner·, £ímº zaji²´uje modularitu a rychlé nasazení. Obsahuje za°ízení s obouru£ním ovládáním pomocí joysticku a dal²í Blazor komponenty, £ímº je docíleno intuitivního ovládání v rozli£ných scéná°ích kontroly AGV. Aplikace je snadno p°izp·sobitelná vyuºitím clean architektury. Tím je docíleno exibilního a efektivního nasazení pro rozli£né kon gurace robot· v pr·myslové automatizaci.

Key words (CZ): automaticky navád¥né vozíky (AGV), webová aplikace, Robot Operating System (ROS), Docker, clean architektura, mikrosluºby, SignalR, Blazor, programovatelný logický automat (PLC), obouru£ní ovládací za°ízení (THCD), komunikace v reálném £ase

Contents

List of Abbreviations

AGV	Automated Guided Vehicle
API	Application Programming Interface
CA	Clean Architecture
CLI	Command Line Interface
CQRS	Command Query Responsibility Segregation
CRUD	Create, Read, Update, Delete
DB	Data Block
DB	Database
DDS	Data Distribution Service
ERP	Enterprise Resource Planning
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HW	Hardware
I2C	Inter-Integrated Circuit
IPC	Industrial PC
IPC	Intra-Process Communication
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JWT	JSON Web Token
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport
OPC UA	Open Platform Communications Uni ed Architecture

PLC	Programmable Logic Controller
PoC	Proof of Concept
QA	Quality Attributes
QoS	Quality of Service
RAM	Random Access Memory
RCL	Razor Class Library
REST	Representational State Transfer
ROS	Robot Operating System
S7	Siemens Step 7
SDK	Software Development Kit
SIL	Safety Integrity Level
SOA	Service-Oriented Architecture
SSH	Secure Shell
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
THCD	Two-Hand Control Device
TIA	Totally Integrated Automation
TLS	Transport Layer Security
UDP	User Datagram Protocol
UI	User Interface
XML	Extensible Markup Language

List of Figures

Introduction

Leuze Electronics GmbH o ers a wide range of optical sensors for industry usage, including industry safety devices. One of the agships is the lidar safety scanner RSL 400 and the new RSL 200.

This lidar scanner can be used to guard hazardous areas in factory environments, to restart warning and protection elds after detecting an obstacle, and to navigate automated guided vehicles [1].

As the need for AGVs with safety features has increased in recent years, Leuze aims to develop sensors with new features and con gurations speci cally for this industry.

The company`s goal is to develop a platform that would allow sensor testing in real world setups and environments. This platform is meant to work as a proof-of-value and proof-of-technology for the AGV applications of the RSL and other sensors.

This work aims to develop part of this platform, speci cally a vehicle handling application for robot operators.

There are already two proof-of-concept platforms and a prototype one. Platform systems are built mainly on top of ROS (Robot operating system) and industry Siemens PLC (programmable logic controller). The ROS based system acts as a high-level controller, while the PLC system accommodates the industry safety program, movement commissioning, and hardware-related drivers.

The main purpose of the handling application is to provide a reliable and e cient way to control the vehicle in the di erent operating modes while meeting industry standards for material handling applications. The application would e ectively bridge the gap between web, robotic, and industry technologies.

Chapter 1

System scope de nition

Given the company's commitment to PLC industry applications, making both hardware and software for Pro net, Pro safe, EtherCAT, and Ethernet/IP, the low-level systems are built with PLCs and PLC controlled sensors. High-level systems, such as autonomous guidance, automatic sequences, and line-following algorithms, are developed on Linux computers utilizing ROS.

The system in its current state has a PLC unit and an IPC with ROS. The missing part is a user application and a user interface.

There is also a need to control the ROS system runtime and to have information about the state of its components.

The problems were summarized into two main points:

1. UI: "We need an interface that centralizes the controls and state reading. We want a convenient robot control application on top of the current ROS system."
2. ROS process control: "The system must dynamically launch and terminate the actual ROS functions. Developers now use a command line to start the robot in di erent scenarios. This is inconvenient for an end user."

1.1 Market research

Why is there even a need for a custom solution? Why not use an existing platform instead? Also, half-way solutions exist, so why not use them as a base for this system?

1.1.1 SIMOVE

Siemens SIMOVE is a material handling software platform for flexible factory and warehouse environments. Its key features are [2]:

- Carrier control: Control through integration of industrial sensors with integrated communication.
- Navigation control: A navigation stack with pre-defined paths, laser-based navigation, and map management.
- Fleet control: Single system for controlling and orchestrating AGVs and AMRs from different vendors.
- Maintenance and diagnostics: Location, system metrics, alarms, and others.

Figure 1.1: SIMOVE web interface for path planning.

Figure 1.2: SIMOVE mobile app for manual control.

SIMOVE is integrated with PLCs and IPCs and works well in the Siemens ecosystem.

Provides means for automatic, as well as manual handling.

Leuze AGV uses SIMOVE as a base case implementation, but only for PR events and trade fairs.

A major drawback of the SIMOVE system is the fact that the sensors used must be certi ed. Thus, internal testing for AGV purposes of new pre-release versions of Leuze sensors is not possible.

Additionally, working with SIMOVE suggests that it uses dockerized ROS in the background.

1.1.2 Rosbridge suite

Rosbridge suite is a JSON ROS websocket API. It allows interaction with ROS nodes with the Rosbridge protocol:

```
{
    "op" : "subscribe",
    "topic"  : "/cmd_vel",
    "type"  : "geometry_msgs/Twist"
}
```

Figure 1.3: Rosbridge suite protocol usage [3].

Client side ROS libraries, such as JavaScript Roslibjs, use the Rosbridge layer to talk to the underlying ROS system [4].

Although it is convenient to link the web with a backend ROS, it has architectural drawbacks. Its biggest advantage, to connect to any ROS node in the system is also its biggest disadvantage. There is no gateway point where request authorization would be made. That means that the client has full control over the ROS.

A better solution would be to expose only speci c ROS nodes, or just topics and services. Having a custom API also allows for the creation of authorization levels for di erent user groups, e ectively securing the backend ROS.

1.1.3 ROS managed nodes

Managed nodes are de ned by applying a set of rules regarding their life cycle with a pre-de ned state machine [5].

This technique allows to control node activity state based on external conditions or other nodes. For example, a rule can be made to allow node startup only if its dependency node is also started.

Using a managed node, one can achieve mutually exclusive sets of nodes to be run. In that case, the manual mode could be run without interfering with autonomous or some other handling mode.