

**DEVELOPMENT OF INDOOR LOGISTIC AUTONOMOUS MOBILE
ROBOT USING BEHAVIOUR TREE IN ROS2**

By

Getrich Malvern Tannaka

11801012

BACHELOR'S DEGREE

in

MECHANICAL ENGINEERING – CONCENTRATION MECHATRONICS
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY



SWISS GERMAN UNIVERSITY

The Prominence Tower

Jalan Jalur Sutera Barat No. 15, Alam Sutera

Tangerang, Banten 15143 - Indonesia

January 2023

Revision after Thesis Defense on 18 January 2023

STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Getrich Malvern Tannaka

Student

Date

Approved by:

Dr. Rusman Rusyadi, B.Sc., M.Sc.

Thesis Advisor

Date

Dr. Maulahikmah Galinium, S. Kom., M.Sc.

Dean

Date

ABSTRACT

DEVELOPMENT OF INDOOR LOGISTIC AUTONOMOUS MOBILE ROBOT
USING BEHAVIOUR TREE IN ROS2

By

Getrich Malvern Tannaka

Dr. Rusman Rusyadi, B.Sc., M.Sc., Advisor

SWISS GERMAN UNIVERSITY

Behavior Tree (BT) was first introduced in video game Artificial Intelligence, and now become the world's leading way to structure a sequence where the program can switch between tasks in an autonomous agent such as Autonomous Mobile Robot (AMR). With the behavior tree readability and reusability, AMR could do much more complex tasks. AMR is a wheeled robot that is commonly used in industrial warehouses for logistics purposes, and it can path its way through a given map while avoiding obstacles. This thesis focuses on the use of BT, LIDAR, camera RealSense D435 for safety and Battery Swap. Using BT, three sequences were made to align with trolley, to drop trolley, and move away from trolley whilst moving to a position. LIDAR is more reliable than Navigation 2 for more precise positioning with an error of 0.014 rad compared to 0.126 rad. Camera RealSense has an error distance of 7% at 50cm and 8.26% at 100cm. The battery swap method has a downtime of more than 1 hour for every 5 batteries used. Hopefully, this project could contribute to a more complex use of BT in AMR

Keywords: Autonomous Mobile Robot, Behavior Tree, Navigation 2, LIDAR, Realsense D435, Battery Swap.

© Copyright 2023

by Getrich Malvern Tannaka

All rights reserved



DEDICATION

I dedicate this work to for my family, my friends, my university, my country:
Indonesia and the development of technology in Mechatronics field of study.



ACKNOWLEDGEMENTS

I would like to express my gratitude towards my family for believing in and for supporting me throughout my studies at the university.

I would like to thank Dr. Rusman Rusyadi, B. Sc., M. Sc. as my advisor for this thesis. He has been really helpful in guiding me throughout this thesis.

I would also thank Mr. Leonard P. Rusli, M. Sc., Ph. D. for guiding me in the mechanical design discussion, Mr. Fredhy for helping in printing the 3D design, Marcellius Jevin as the lab assistant who also help me when I had questions, Mr. Erikson Sinaga who guide me regarding electrical questions that I had. Moreover, I would like also to thank all the other lecturers for their guidance throughout all semester and also my friends.

I have found my coursework throughout the Curriculum and Instruction program to be stimulating and thoughtful, providing me with the tools with which to explore both past and present ideas and issues.

TABLE OF CONTENTS

	Page
STATEMENT BY THE AUTHOR.....	2
ABSTRACT.....	3
DEDICATION.....	5
ACKNOWLEDGEMENTS.....	6
TABLE OF CONTENTS.....	7
LIST OF FIGURES.....	11
LIST OF TABLES.....	14
1. CHAPTER 1 – INTRODUCTION.....	15
1.1. Background.....	15
1.2. Thesis Problem.....	16
1.3. Significances of Study.....	16
1.4. Research Questions.....	16
1.5. Hypothesis.....	16
1.6. Thesis Structure.....	17
2. CHAPTER 2 - LITERATURE REVIEW.....	18
2.1. Theoretical Perspectives.....	18
2.1.1. Autonomous Mobile Robot (AMR).....	18
2.1.2. Behavior Trees.....	18
2.1.3. Robot Operating System 2 (ROS2).....	19
2.1.3.1. ROS 2 Humble.....	21
2.1.4. Micro-ROS.....	22
2.1.5. Message Queuing Telemetry Transport (MQTT).....	22
2.1.6. Depth Camera.....	23
2.1.6.1. Stereo Depth.....	24

2.1.7.	Previous Research.....	24
2.1.7.1.	Development Of Indoor Logistic Autonomous Mobile Robot Based on Robot Operating System 2	24
3.	CHAPTER 3 – RESEARCH METHODS.....	26
3.1.	System Overview	26
3.2.	Components of Design.....	27
3.2.1.	Hardware Design	28
3.2.1.1.	RealSense D435 Housing.....	28
3.2.1.2.	Magnet Holder for Battery Swap Door.....	28
3.2.1.3.	Battery Swap Method.....	29
3.2.1.3.1.	Door Mechanism for Swap Method.....	29
3.2.1.3.2.	Battery Holder.....	30
3.2.2.	Electrical Components.....	31
3.2.2.1.	Lithium-Ion Battery Pack.....	31
3.2.2.1.1.	Battery Voltage Meter Electric Diagram	32
3.2.2.2.	Intel RealSense Depth Camera D435.....	32
3.2.2.3.	Stepper Motor	33
3.2.2.3.1.	Stepper Motor Driver	34
3.2.2.3.2.	Stepper Motor Electrical Design.....	36
3.2.2.4.	US–100 Ultrasonic Sensor	37
3.2.2.5.	Arduino	37
3.2.3.	Software Design.....	39
3.2.3.1.	Behavior Tree.....	39
3.2.3.1.1.	XML – Code Base Behavior Tree.....	46
3.2.3.2.	Intel RealSense SDK.....	47
3.2.4.	Performance Test	47

3.2.4.1.	Mechanical Design Assembly.....	47
3.2.4.2.	Behavior Tree and MQTT.....	48
3.2.4.3.	Navigation 2 and Modified Maneuvering with LIDAR Method to Move under the Trolley.....	48
3.2.4.4.	Battery Performances.....	48
3.2.4.4.1.	Discharge Time.....	48
3.2.4.4.2.	Charging Time.....	49
3.2.4.5.	Battery voltage measurement.....	50
3.2.4.6.	Camera Test.....	50
4.	CHAPTER 4 – RESULTS AND DISCUSSIONS.....	52
4.1.	Assembly and Mechanical Design Fitting.....	52
4.1.1.	3D Print Parts.....	52
4.1.1.1.	RealSense D435 Depth Camera Housing.....	52
4.1.1.2.	Magnet Holder.....	54
4.1.2.	Overall Robot Assembly.....	55
4.1.2.1.	RealSense D435 Mounting.....	55
4.1.2.2.	Battery Swap Method Door Mechanism.....	56
4.2.	Behavior Tree and MQTT.....	57
4.3.	Testing Nav2 and Modified Maneuvering with LIDAR Method to Move under the Trolley.....	62
4.3.1.	Nav2 Test.....	62
4.3.1.1.	Directly set Target Position to Desired Position.....	62
4.3.1.2.	Set Target Position step by step until to Desired Position.....	63
4.3.2.	Modified Maneuvering with LIDAR Test.....	66
4.4.	Battery Performance.....	69
4.4.1.	Discharge Time.....	69

4.4.2.	Charging Time	70
4.4.3.	Battery Voltage Measurement	71
4.5.	Camera Performance.....	71
4.6.	ROS2 Packages Made.....	78
4.6.1.	Modified Maneuvering with LIDAR Package.....	78
4.7.	ROS2 Humble and Navigation 2 Problem.....	81
5.	CHAPTER 5 – CONCLUSIONS AND RECCOMENDATIONS	84
6.	GLOSSARY	86
7.	REFERENCES	87
8.	APPENDIX A – MECHANICAL DESIGN	88
9.	APPENDIX B – DATASHEETS.....	89
10.	APPENDIX C – INSTALLATION PROCEDURES.....	92
11.	APPENDIX D – OPERATING PROCEDURE.....	94
	CURRICULUM VITAE.....	95

LIST OF FIGURES

Figures	Page
Figure 2.1: ROS 2 Humble Hawksbill Logo.....	21
Figure 2.2: Multiple nodes communicating with each other via publishing and subscribing data.	22
Figure 2.3: Example of MQTT Publish and Subscribe.....	23
Figure 2.4: Side-by-side image of RGB and depth.....	23
Figure 2.5	24
Figure 3.1: System Overview.....	27
Figure 3.2: RealSense D435 Housing.....	28
Figure 3.3: Magnet Holder.....	28
Figure 3.4: Door Mechanism for battery swap method (front view).....	29
Figure 3.5: Door Mechanism for battery swap method (back view)	29
Figure 3.6: Battery Holder	30
Figure 3.7: Top view.....	30
Figure 3.8: Lithium-Ion Battery Pack.....	31
Figure 3.9: Battery Voltage Meter Wiring Diagram.....	32
Figure 3.10: Intel RealSense Depth Camera D435	32
Figure 3.11: Stepper Motor NEMA 23	33
Figure 3.12: DM556 Stepper Driver	34
Figure 3.13: Microstep Resolution Selection.....	35
Figure 3.14: Current Setting	35
Figure 3.15: Electrical Diagram of DM556 and NEMA 23	36
Figure 3.16: Connection from Teensy to DM556 from previous thesis	36
Figure 3.17: US-100 Ultrasonic Sensor	37
Figure 3.18: Arduino Nano	37
Figure 3.19: MQTT and Behavior Tree System Overview	39
Figure 3.20: “Get Trolley” Behavior Tree	40
Figure 3.21: AMR align with Trolley	41

Figure 3.22: AMR under the Trolley	42
Figure 3.23: AMR locking to the Trolley	43
Figure 3.24: “Put Down Trolley” Behavior Tree.....	44
Figure 3.25: “Movement” Behavior Tree	45
Figure 3.26: Behavior Tree of “Get Trolley” in XML format.....	46
Figure 3.27: Behavior Tree of “Put Down Trolley” in XML format.....	46
Figure 3.28: Behavior Tree of “Movement” in XML format	47
Figure 4.1: First RealSense Design.....	52
Figure 4.2: Second RealSense Design (Left).....	53
Figure 4.3: Second RealSense Design (Front, Back).....	53
Figure 4.4: Magnet Holder (Front, Back)	54
Figure 4.5: Final RealSense Mounting	55
Figure 4.6: Camera View	55
Figure 4.7: Magnet Holder Mounting and Door Mechanism	56
Figure 4.8: Battery Holder	56
Figure 4.9: DockingEvent action node	57
Figure 4.10: DockingEvent action node pt2	57
Figure 4.11: DockingEvent action node pt3	57
Figure 4.12: DockingEvent action node pt4	58
Figure 4.13: MinimalPublisher action node (stepper action node).....	58
Figure 4.14: MinimalPublisher action node.....	59
Figure 4.15: CameraEvent action node.....	59
Figure 4.16: CameraEvent action Node pt2.....	60
Figure 4.17: Testing Action Nodes in Behavior Tree (top) and Testing of BT with MQTT (bottom)	61
Figure 4.18: Inflation Layer (The red circle on the left image indicating the trolley, and the red circle on the right image indicating the trolley’s inflation layer)	63
Figure 4.19: Modelling of position of AMR.....	64
Figure 4.20: Trolley location in terms of angle	67
Figure 4.21: AMR under Trolley	68
Figure 4.22: Trolley locker	69
Figure 4.23: Lithium-ion Battery Discharge Time	70

Figure 4.24: Voltage (coding) vs Actual Voltage	71
Figure 4.25: Data echo topic “/aligned_depth_to_color/image_raw”	72
Figure 4.26: cv_bridge	72
Figure 4.27: Before and After Hole Filling Filter	73
Figure 4.28: Object Type Detection(top) and Distance Reading(bottom).....	74
Figure 4.29: Distance of object and the AMR is 30 cm (before calibration).....	75
Figure 4.30: Distance of object and the AMR is 50 cm (before calibration).....	75
Figure 4.31: Distance of object and the AMR is 100 cm (before calibration).....	75
Figure 4.32: realsense-viewer on-chip calibration.....	76
Figure 4.33: Distance between object to AMR is 30cm (after calibration)	77
Figure 4.34: Distance between object to AMR is 50cm (after calibration)	77
Figure 4.35: Distance between object to AMR is 100cm (after calibration)	78
Figure 4.36: Publisher Subscriber	78
Figure 4.37: Subscription callback	79
Figure 4.38: If Statement for trolley alignment	80
Figure 4.39: If Statement for moving under trolley	80
Figure 4.40: If statement for moving away from trolley	81
Figure 4.41: Obstacle layer is not working	82
Figure 4.42: Obstacle Layer code (top) and Obstacle Layer (bottom)	83
Figure 8.1: Magnet Holder Isometric View	88
Figure 8.2: Intel RealSense Housing Isometric View	88
Figure 9.1: Driver DM556 Datasheet 1	89
Figure 9.2: Driver DM556 Datasheet 2	90
Figure 9.3: Intel RealSense D435 Datasheet	90
Figure 9.4: Arduino Nano Pinout.....	91

LIST OF TABLES

Table	Page
Table 4.1: W angular result using Nav2 with tolerance 0.25m and 0.25 rad.....	63
Table 4.2: Tolerance 0.15 m and 0.15 rad	65
Table 4.3: AMR position using MML	67

