IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Intelligent Carrier

Boaz K Michael, Harsh T Krishna, Amaldev P M, Joyal Santhosh, Manjusha P,
B-Tech Student, B-Tech Student, B-Tech Student, B-Tech Student, Asst. Professor Mechatronics
Mechatronics,
Nehru College of Engineering and Research Centre, Thrissur, India

ABSTRACT: Logistics especially cargo handling and import-exports highly depend on internal and external transportation chains which are usually handled by human drivers and overseers or as in recent developments, by partially automated vehicles which has its disadvantage of free-thinking ability. Our project "intelligent carrier" is primarily aimed to make cargo and luggage handling in airports automated, with the help of implanting prefixed paths and sub-paths for intelligent path selection in the carrier, with added ability to avoid obstacles, using its camera module. This robot is completely developed to be controlled by an external processing center which can control multiple robot units. Its complete function is to receive positional data of an airplane on its arrival and allot a robot carrier which travels along the mapped path with intelligent obstacle avoidance to reach predestined position based on proximity to the anchor location (as in anchor-tag setup). This carrier can efficiently replace current systems due to its cost-effective (economical) approach.

I. INTRODUCTION:

This robotic carrier uses various software and hardware elements that make it suitable for its intelligent application. The main software element installed in the robotic system is microROS framework that performs the role of a communicating medium between the functional ROS in the processing unit and the robotic system. micro-ROS enables the development of robotic applications to be deployed onto microcontrollers that are typically limited in computational resources. It follows the ROS 2 architecture and uses DDS-XRCE (Data Distribution Service for eXtremely Resource-Constrained Environments) as its middleware. It replaces the Linux-based communication stack with POSIX-based RTOS (such as FreeRTOS, Zephyr, or NuttX) for microcontrollers. The layered and modular architecture ensures compatibility with ROS 2 components. The second important software element is ROS 2 installed in the processing unit, which is the heart of the complete operations. ROS 2 (Robot Operating System 2) is a collection of open-source software libraries and tools designed for developing and running robotics applications. It provides a flexible framework for building complex robotic systems. It supports multiple programming languages, making it accessible to a wide range of developers. Various hardware elements play the crucial role in implementing the software elements. The ESP32S2 is the main hardware element upon which microROS is embedded. For the crucial application of communicating the distance between the arrived airplane and the robotic carrier an FTM (Fine Time Measurement) communication protocol is used which is available in ESP32S2. To extract this ability of this microcontroller two such boards are used to transfer this data for the robotic carrier to reach to close proximity with the airplane. A camera module (ESP 32 CAM) is used connected to the main board (ESP32S2) to map the paths required for the carrier to traverse and for its intelligent obstacle avoidance. A motor driver with 2 connected motors along with a roller makes up the carrier's mechanical elements that help in its motion.

II. SETTING UP ENVIRONMENT:

2 Frameworks where initially found to be intended and required for development of the carrier which was the ESP-IDF and ROS2

The installation of ESP-IDF was completed using the guidelines stated by the Espressif ESP-IDF installation procedure for windows as in Standard Setup of Toolchain for Windows - ESP32 - — ESP-IDF Programming Guide latest documentation (espressif.com). After completion of which the IDF pwershell was embedded in VS Code for simpler coding, compiling and running of application

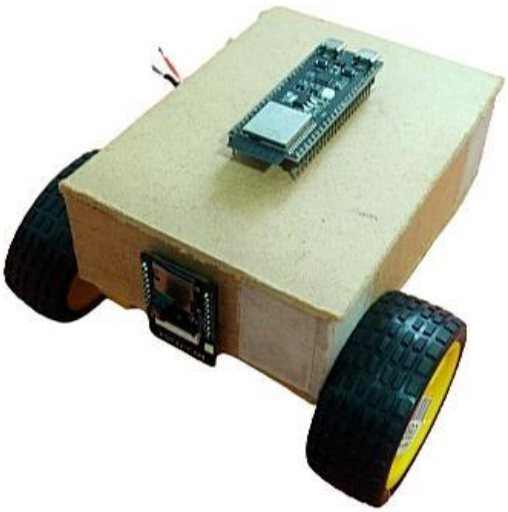
The ROS2 installation was done using Docker GUI as per guidelines set by Docker and ROS2 documentation

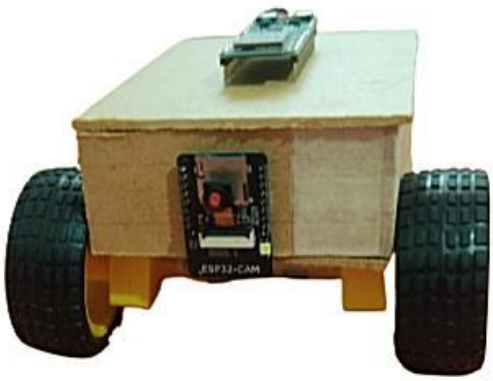
III. BACKGROUND:

The ROS2 platform needs a powerful operational processor for which a Laptop with high processing ability was selected. The installation with docker images and containers were done with assistance from git repositories. The docker GUI (Graphics User Interface) is a very powerful linux subsystem help that gives a very good UX and UI support for robot development. The microROS is a very powerful communication framework between microcontrollers and the processing units. It was built and run into esp32s2 using platform io. The espressif esp idf libraries were pulled from github repositories built by espressif team for ROS framework. These libraries were used to embed the position tracking ability of the carrier using FTM.

IV. WORKING:

The complete working of the carrier robot is done through the communication protocols set up within the microcontroller. The micro-ROS installed in the ESP32S2 communicates to the ROS2 installed in the processing center (Laptop). On receiving the positional initiated data to the processing unit from the airplane, a robotic carrier is allocated for being present at the location to carry the cargo/luggage from the airplane. The robotic carrier uses FTM to find the approximate location of the airplane. The camera module in the carrier gives real-time footage of its surroundings to assess whether any obstacles appear and require to be avoided or not. It also performs the function of initial mapping of the predestined path. The carrier on reaching close-proximity with the airplane, halts its motion. It then returns to its initial point of start on initiation from the processing center.





V. CONCLUSION:

This project “Intelligent Carrier” was an experimental project intended at making completely automated as well as highly economical carriers for assisting the moving of cargos from depot to airplanes or vice versa on point initiation. The economical aspects were met by using microcontroller instead of microcomputer, which are usually used in carrier robots (mostly Raspberry pi). The automation parts were developed using ROS2 Framework and communicated to microROS in ESP32S2. This Robot carrier can be used not only for airport logistics handling but also for numerous other application including inventory handling in factories to assisted carrier to help in storage handling. The primary aim of making carrier robots at an affordable rate with automation features was met making this project highly preferable in modern-day material and logistics handling.

VI. REFERNCES:

1. Borenstein, Johann, et al. "Mobile robot positioning: Sensors and techniques." Journal of robotic systems 14.4 (1997): 231-249.

2. Jo, Wonse, et al. "SMARTmBOT: A ROS2-based low-cost and open-source mobile robot platform." arXiv preprint arXiv:2203.08903 (2022).

3. Li, Bin, Zhengguang Ma, and Yongguo Zhao. "2D Mapping of mobile robot based on micro-ROS." Journal of Physics: Conference Series. Vol. 2402. No. 1. IOP Publishing, 2022.

4. Vales, Valentín Barral, Omar Campos Fernández, Tomás Domínguez-Bolaño, Carlos J. Escudero, and José A. García-Naya. "Fine time measurement for the Internet of Things: A practical approach using ESP32." IEEE Internet of Things Journal 9, no. 19 (2022): 18305-18318.