ACSO BahiaRT@Home 2024: Team Description Paper

1st Tamir A. Amorim, 2nd Gabrielle F. S. Carvalho, 3rd Vitória F. N. Matos, 4th Pedro H. O. dos Santos, 5th Luis J. S. Junior, 6th Samuel J. Cesar, 7th Reinan C. Amaral, 8th Pedro L. J. Santos, 9th João P. S. Santana, 10th Elias R. da Silva, 11th Filipe N. Silva, 12th Kaique W. S. da Silva, 13th Davi M. B. Barbosa, 14th Josemar R. de Souza, 15th Ivanoé J. Rodowanski, 16th José Grimaldo da Silva Filho, 17th Marco A. C. Simões, 18th Jorge A. Campos, 19th Ana P. F. M. Mascarenhas

Centro de Pesquisa e Arquitetura de Computadores, Sistemas Inteligentes e Robótica (ACSO).
Universidade do Estado da Bahia (UNEB), Salvador, BA, Brazil, teambahiart@gmail.com

***Abstract*— This paper presents the ACSO Bahia Robotics Team (BahiaRT) and describes an autonomous service robot named Bot Intelligent Large capacity Low cost (Bill) and its functions, such as navigation, manipulation, people and object recognition, human-robot interaction and decision making as well as the Bill's hardware and software systems. Furthermore, the paper highlights research interests and scientific contributions of BahiaRT from the Center of Computer Architecture and Operating Systems (ACSO) at the Bahia State University (UNEB).**

Keywords: Bill, service robotics, object detection, assistive robotics, speech recognition, RoboCup@Home.

I. Introduction

The Center of Computer Architecture, Intelligent Systems and Robotics (ACSO) at the State University of Bahia (Uneb) has been participating in RoboCup with the BahiaRT team since 2009 in leagues such as 2D Soccer Simulation, Mixed Reality, 3D Soccer Simulation, and @home, among others.

The service robot proposal of BahiaRT for the RoboCup@Home league is called Bill. It was born in 2014 as a result of research projects in assistive robotics. Its main functions are communicating with humans through natural language processing, recognizing objects and faces, and navigating through unknown environments. This paper presents the ACSO Bahia Robotics Team (BahiaRT) and describes an autonomous service robot BILL and its functions, such as navigation, people and object recognition, human-robot interaction and decision making as well as the BILL's hardware and software systems. Furthermore, the paper highlights this year-long BILL improvements.

Over the years BILL has been improved to better meet the @home challenges. This paper presents the third generation of BILL, named **BILL Estranho**, and its main improvements. Past year, a great part of the BahiaRT effort was dedicated to building BILL Estranho. This work involved re-engineering the hardware using new components, redefining the architecture, and changing the operating system to ROS2. As a consequence, this year, other functionalities, i.e. voice recognition, face and object recognition, navigation, and manipulation was the focus of update. This year, the team had two big releases: the first one, to improve object recognition, the BahiaRT added a system that combines it with text recognition Optic Character Recognition (OCR) to improve BILL's accuracy. This approach uses existing tools (YOLO [1] and Tesseract [2]). The combination allows robots to handle complex situations and is particularly relevant for service robotics where precise vision is crucial. The second release was in our speech recognition system that employed advanced machine learning techniques, specifically the GPT-2 [3] model for understanding and generating responses in natural language.

∗ This work is partially funded by CNPq/PIBIC, FAPESB/IC and UNEB/PICIN

II. BahiaRT's advances in innovative technology and research interests

The main research interests of BahiaRT involve robotics and artificial intelligence, specifically focusing on interaction and cooperation between human-robot, applications of artificial intelligence for services, standardization, and system integration, in addition to the construction of intelligent robots with low-cost components and open source. Developing an autonomous robot like BILL requires knowledge of computer and mechatronics technology to integrate systems and hardware solutions.

However, despite the importance of this type of research, there are still not several autonomous robots running in human daily lives. Furthermore, as well as in other sciences, autonomous robots can achieve different results in practice due to the occurrence of unpredictable events and environmental changes. Therefore, there is still a long road ahead, with much work to be done for spreading services robots in residences around the world.

To fill this gap, practitioners should provide feedback on robotic research, sharing experiences for reducing differences among countries and their laboratories. To allow replication of research and to contribute to the development of robotic research, BahiaRT will provide access to the codes, hardware, and software list in this paper and on Bill's website.

Fig. 1. BILL Estranho.

III. BILL Estranho

Based on the characteristics of robots built by ACSO, used in competition over the last 18 years, and observations on equipment from other parts of the world, we present BILL, base type 3, code name Estranho, whose main feature is 04 differential wheels (Mecanum Wheels).

IV. BILL's Architecture

One of special feature of robot BILL's is his architecture (Figure 2), that is basically divided into levels: a high level, containing the functions related to the robot's abilities; and the low level, containing the controllers and drivers that send command signals to the actuators and receive signal feedbacks from the sensors.

As shown in Figure (Figure 2), at the top of the architecture (in orange), is the interaction layer with the external world. BILL contains two monitors, one representing BILL's face (attached to its head) and another used for HMI (human-machine interface) (attached to his body). Also contains a microphone and speakers, Lidar, and webcam. The manual locomotion control can also be performed via a joystick.

The fault-tolerance Fox-Dog method is implemented using two ESP32 microcontrollers replicated (in red in Figure 2) with ROS2. They control the left/right motor front (LMF/RMF) and the left/right motor rear (LMR/RMR). So, if one microcontroller fails, the other takes over.

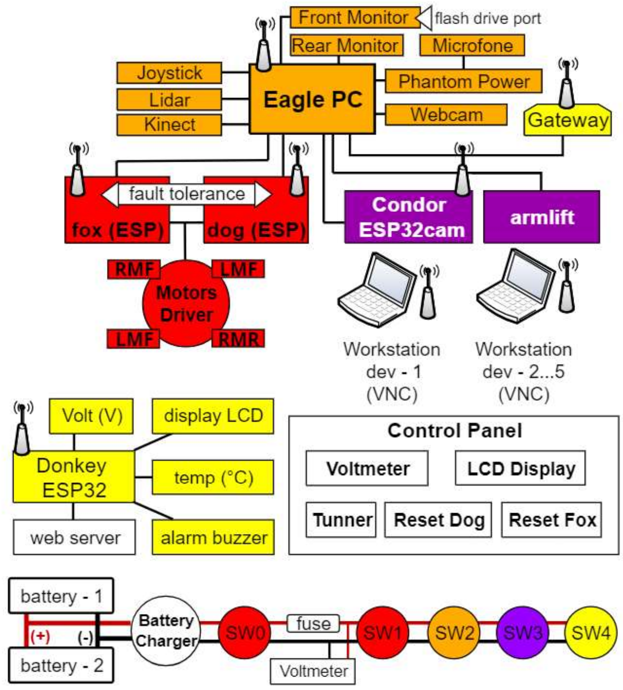

Fig. 2. Bill Estranho Architecture.

V. High-level process software

A. Navigation

Navigation is the keystone for efficient execution and environment interaction for robots. The components used by Bill for navigation are: encoders output, odometry, Slam_toolbox, behavior tree based navigation node called 'bt_navigator' (ROS2), Adaptive Monte Carlo Localization (AMCL) (ROS2), map_server (ROS2) and 360° laser scanner. The encoder data is used by odometry module to estimate the movements of the robot in space. Further, the odometry data is used to trace trajectory to a desired target by the Behavior-Tree Navigator. Once all data are published, the simultaneous mapping and localization using the AMCL [4] is activated integrating the 360° laser scan data. Simultaneous Localization And Mapping (SLAM) approach is used to map the environment and provide self-localization in this map. First, the map is built using the incremental mapping package, Slam_toolbox [5]. Then, the grid map generated is done by LIDAR sensor, which is able to get the 2D world data. The next step is creating the path planning based on the occupancy grid map that is updated based on the navfn_planner [6]. Then the shortest path to the goal is computed by means of the D* Lite algorithm [7] to ensure obstacle avoidance over incremental mapping. The motion planning in charge of getting the path planning and relating linear and angular motion is triggered, which applies the kinematics control law, and sends a message to low-level control.

B. Vision

This module handles the reception, processing, and response to external stimuli captured through image capture. It consists of two main sub-modules: facial recognition and object detection.

For object detection, BILL utilizes YOLO (you only look once) version 8 [1], a cutting-edge technology in object recognition capable of real-time detection across a broad spectrum of objects. Training for BILL involved 45 distinct objects, each represented by 20 photos taken in various locations and orientations. This selection was designed to ensure a diverse testing environment. We employed images containing varying numbers of objects per frame to assess the impact of parameters and train photo quantity on recognition accuracy. The labeling process was facilitated by RoboFlow [8] software.

Training required substantial computational resources, prompting us to utilize the Google Colab [9] environment for its processing capabilities, resulting in significant time efficiency gains.

The image below illustrates a prediction generated from our custom dataset, demonstrating successful training outcomes aligned with our project objectives.

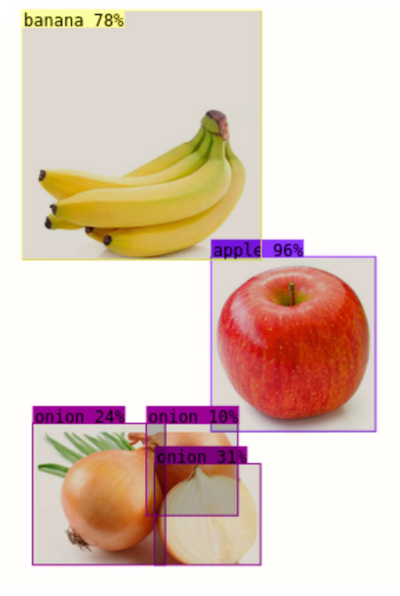

Fig. 3. Yolo's prediction.

To improve the accuracy of our system, we are integrating object recognition with optical character recognition OCR. To do this, we use the YOLO tools for object recognition and Tesseract [2] for OCR. This approach has already proven to be effective in several sectors. For example, in industry, as highlighted by Monteiro [10], in traffic management, according to Nandhakumar [11], in banking applications, such as instant invoice recognition [12], in healthcare, and also in the field of robotics, as described by Punde [13] In addition, integrating these technologies enables the creation of more robust and versatile solutions, capable of dealing with various scenarios and challenges.

This experiment stands out as innovative and highly relevant for the academic community, especially in service robotics, where the accuracy of robot vision is essential for correct object recognition. The research is currently in the integration and evaluation phase, to improve the accuracy of object recognition in the BILL system. The importance of this study lies in its ability to address practical challenges and provide solutions that can be directly applied in real-world scenarios, from home care to complex industrial applications. We expect that the integration of object recognition and OCR technologies will not only significantly improve the accuracy of BILL, but will also contribute to significant advances in the efficiency and functionality of service robots, setting a standard of excellence in the field. Furthermore, this work has the potential to serve as a solid foundation for future research and development, encouraging continued innovation and exploration of new possibilities within service robotics.

Inspired by BILL's emphasis on achieving high capacity at a low cost, we designed a two-stage system for efficient facial recognition. The first stage focuses on facial detection, a crucial step in isolating the face from the rest of the image. Here, we leveraged the power of the Haar cascade algorithm [14]. This open-source technique, readily available within the OpenCV library [15], excels at swiftly pinpointing faces in an image. It achieves this by identifying specific rectangular patterns, essentially acting like a digital fingerprint scanner for facial features. By employing Haar cascade [14], we can effectively isolate the face of interest for further processing.

Following successful detection, the Dlib library [16] takes center stage. Dlib [16] implements a sophisticated landmark detection technique. This technique goes beyond simply locating the face; it meticulously pinpoints 68 key coordinates across the face, creating a detailed map of its unique characteristics. This information, encompassing the relative positions and shapes of facial features like eyes, nose, and mouth, is then extracted and mapped to train the facial recognition model.

To gauge the system's effectiveness, we conducted recognition tests with a group of 55 students from Bahia State University. After a brief training period, the algorithm surpassed our expectations. Not only did it achieve individual student recognition with a high degree of accuracy, but it also demonstrated the remarkable ability to identify specific individuals within a group setting. This success highlights the potential of this two-stage approach, combining the efficiency of Haar cascade [14] with the detailed analysis of Dlib [16] for robust facial recognition.

C. Speech Recognition and Voice

Voice is the most utilized form of human-machine interaction to issue commands to BILL, either through command lines or natural language. Due to BILL's upgrade to ROS2, we are currently working on adopting Google's Speech Recognition software to enable voice interaction.

Our speech recognition system employs advanced machine learning techniques, specifically the GPT-2 [3] model for understanding and generating responses in natural language. Developed by OpenAI, GPT-2 [3] is a transformer-based language model known for its ability to generate coherent and contextually relevant text. For our purposes, GPT-2 [3] has been fine-tuned with a custom dataset consisting of various command phrases and natural language interactions relevant to BILL.

The model training process involves several steps. Initially, we compiled a comprehensive dataset of voice commands and corresponding actions. Subsequently, the data was cleaned and normalized to ensure consistency and quality. Training of the GPT-2 [3] model was conducted using TensorFlow and PyTorch libraries, wherein the model was fine-tuned on this dataset to learn the specific context and commands used in our application. Additionally, we utilized the Speech Recognition library to interface with Google's Speech Recognition API.

During the inference phase, the system captures voice commands through BILL's array of microphones. The captured audio is pre-processed to enhance clarity and reduce background noise. The processed audio is then transcribed into text using Google's Speech Recognition API. The transcribed text is fed into the fine-tuned GPT-2 [3] model, which interprets the command and generates an appropriate response or action.

Audio capture is a critical component of our speech recognition system. BILL is equipped with high-quality microphones strategically positioned to optimize sound capture. Audio processing includes noise reduction to filter background noise and improve signal-to-noise ratio, along with real-time processing to ensure audio capture and processing occur in real-time, enabling quick response to voice commands.

This comprehensive approach to speech recognition and voice interaction aims to create a robust and reliable system that enhances human-BILL interaction, making it more intuitive and efficient.

VI. Low-level process software

The low level is composed of a proportional control running on ESP32 boards. Four different embedded software, micro ROS [17] based, was developed with specific functions, running in different Esp32 Devices.

On the first, Dog device, the software communicates with Eagle to get instructions to activate motor drive for robot motion.

On the second, the Foxy device, software gets encoder data to process odometry to get data about the amount of motor motion realized for close-loop locomotion control.

On the third device, Donkey, the software receives battery voltage and temperature data, displays information, and shows the menu where BILL's operation mode is selected.

Fourth, the Condor device, embedded in the software, is dedicated to getting information from one camera used in the vision system to detect objects.

In addition, a fault-tolerant system was designed to integrate Dog and Foxy Devices, checking each other to verify if both controllers are correctly running, "the dog pursuit the fox, and the fox pursuit the dog".

VII. BILL'S VISION HARDWARE

BILL's vision system is based on two different subsystems, first system is composed of a LIDAR (Light Detection and Ranging) sensor used for distance measurements for environment mapping, and second system uses two cameras, getting images, one for face recognition and another for object recognition.

A. *Laser Scanning System*

For this year, LIDAR device was updated, for model RS Lidar N10, figure 4. According to the manufacturer [18] based on ToF (time of flight), the N10 PLUS LIDAR, can conduct 360° two-dimensional scans and detection of the environment. The LIDAR uses a wireless power supply and wireless communication internally, and the pulse repetition frequency (PRF) is 5.4 KHz. The LIDAR reaches a measurement accuracy of ±3 cm with a maximum range of 15 m. With such high performance, the LIDAR mainly applies to scenarios that require precise location and obstacle avoidance, including indoor service robots, AGVs, cleaning and sterilization robots, drones, and more.

Fig. 4. RSLIDAR model N10.

B. *Object Recognition Camera*

To BILL is equipped with an ESP32-CAM module, , a low-cost ESP32-based development board with an onboard camera, small in size. It is an ideal solution for IoT applications, prototype constructions, and DIY projects [19].

VIII. ELECTRIC AND ELECTRONIC CIRCUITS AND DEVICES

A. *Motor drives*

To move, BILL is equipped with a circuit to drive 4 different geared motors attached to mecanum wheels. Instructions from core to ESP32 actuate in 2 circuits drive based on chip L298 dual bridges.

B. *Circuit Sensor Position Feed Back*

For closed-loop positioning control, Bill uses 4 encoders based on hall sensors, one on each axis of the locomotion motors. The angular position signal is read and processed by an ESP32 microcontroller, which makes it available to the central computer.

C. Power Supply

The electrical system is supplied by two packs of PowerTek lead-acid batteries with 12 volts and 7A/h capacity. This type of battery is usually found in embedded systems, suck as security systems. Because of this, it is easy and affordable to obtain in local stores. Its negative aspect is that it takes a long time to recharge, almost 10 hours to recharge completely. BILL has a hanging charging dock where the batteries cam be charged when they are not running. This battery packs supply enough power to @Home competition needs, however for real domestic application its necessary improve the charge capacity.

D. Low Level Processing Devices

An essential component of Bill's electronic circuit is the use of ESP32 devices, Figure 5 [20], because it is popular and accessible, there is a large community developing applications and libraries for it, and it is easy to integrate with ROS2 system using Micro ROS system [17] dedicated for these type of devices. These devices are used to get access, in real-time, to digital and analogic data from sensors and send commands for actuators and interfaces, for example.

Fig. 5. Circuit board of ESP 32 device.

BILL uses four different ESP 32 devices with different functions on the system. ESP#1 (called Fox) activates the motion motors, ESP#2 (called Dog) gets data from motion encoders, and ESP#3 (called Donkey) gets data from temperature and battery voltage sensors and sends texts to LCD display and run a web server. ESP#4 (called Condor) is a specific module designed to be used with the camera to get data from one camera for a vision system.

E. BILL's CPU (Eagle)

A central processing unit (called Eagle) is a portable computer mini PC Bee Link, Figure , equipped with an Intel 11th generation N5095 processor, 16GB DDR4 RAM, and 512GB SSD memory capacity, were high-level programs run.

This computer is a good solution for these applications because it is designed for lower power consumption and good processing capacity.

IX. BILL'S CONTROLS AND INTERFACE

A mode of remote controlling uses a wireless gamepad model ThrustMaster Controller.

BILL also contains a control panel, shown in figure 6, with an LCD that allows monitoring voltage, temperature, connection and generates alarms in some situations. In figure 6, #1 is battery charger connector, #2 is general power switch, the set of buttons #3 to #6 power switch different circuit systems isolated. In LCD menu, turning and pushing knob #7, users can operate some pre-programmed functions, to execute simple tasks and start some operation modes. Buttons #8 and #9 reset Foxy and Dog subsystems.

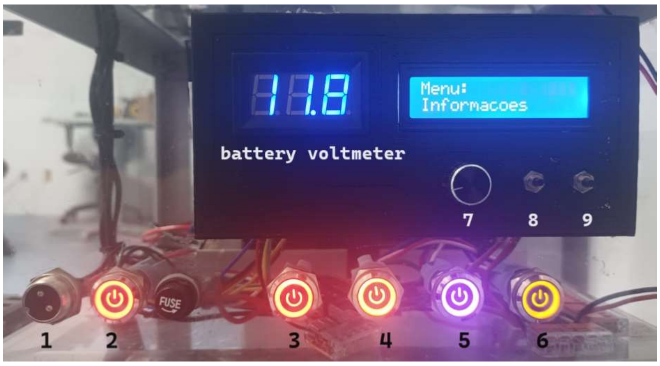

Fig. 6. BILLs control panel.

X. BILL´S MECHANICAL HARDWARE

A. Chassi and Boddy

BILL has a chassis of aluminum extruded L bars, usually accessible in many local metal stores. A square frame 2" L is attached to the robot base, where the locomotion system (motors with gearboxes) and batteries were attached to keep a low center of mass. Others modular squares frames, made by 1" L aluminum bars also, are stacked for the upper body, are designed to receive other parts of hardware, parts like as LCD, communication system and CPU, this way, transportation, maintenance and upgrades are easier to do. Clear acrylic sheets cover the body to protect internal components.

The locomotion system is based on 4 powered mecanum wheels for omnidirectional movements. This is a very important improvement from the last version. This type of locomotion system is much more enjoyable for service robot applications because it can move in 8 different directions without turning in your instant rotating center and is easy to turn if needed.

B. Five Degree of Freedom Objects Manipulator

Object manipulation plays an important role to interact with a home environment. To meet that requirement, BILL is equipped with a 5 DoF TurtleBot Robotic Arm, as shown in Figure 7 [21], with clamp, built from a set of 5 Dynamixel AX-12+ model servo motors, 4 of which are for the joints and 1 for the claw. Furthermore, the system has a joint with a vertical prismatic degree of freedom, driven by a stepper motor, to arm leveling when handling objects.

This system was controlled by an Arduino Arbotix-M, but for this year, it was upgraded for a new controller based on Arduino MKR Zero board with Robotis Dynamixel Shield designed for this purpose.

This improvement was motivated by the fact that the new board is compatible with the Micro-Ros system.

Fig. 7. Turtlebot Robotic Arm

In addition, we explore the Gazebo [22] simulation environment. However, there are no much documentation using this device for ROS2 applications. The next step is, trying to fund some way to transfer the knowledge acquired in the simulator to BILL's physical arm.

XI. Conclusions

We can mention that the improvements observed are notorious; especially those related to the use of ROS2 and micro-ROS due to the ease and interaction between the various parts of the robot and also the system that improve object recognition using Tesseract[2] and Yolo[1] together. Another highlight was the results obtained with the use of Esp32, for low level processing, a robust and versatile microcontroller that certainly "came to stay". So far, we are satisfied with what has been observed and intend to continue investing in this new hardware and software architecture.

References

[1] "Yolo," https://pjreddie.com/darknet/yolo/, accessed: 2023-07-21.

[2] "Tesseract," https://github.com/tesseract-ocr/tesseract, accessed: 202-06-25.

[3] "Chatgpt," https://chatgpt.com/, accessed: 2024-07-03.

[4] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.

[5] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 155–160.

[6] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of euclidean distance maps and voronoi diagrams," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 281–286.

[7] J. Neufeld, M. Sokolsky, J. Roberts, A. Milstein, S. Walsh, and M. Bowling, "Autonomous geocaching: Navigation and goal finding in outdoor domains," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 47–54.

[8] "Roboflow," https://app.roboflow.com/, accessed: 2024-06-19.

[9] "Googlecollab," https://colab.google/, accessed: 2024-07-03.

[10] G. Monteiro, L. Camelo, G. Aquino, R. d. A. Fernandes, R. Gomes, A. Printes, I. Torné, H. Silva, J. Oliveira, and C. Figueiredo, "A comprehensive framework for industrial sticker information recognition using advanced ocr and object detection techniques," *Applied Sciences*, vol. 13, no. 12, p. 7320, 2023.

[11] R. G. Nandhakumar, K. N. Devi, N. Krishnamoorthy, S. Shanthi, V. R. Pranesh, and S. Nikhalyaa, "Yolo based license plate detection of triple riders and violators," in *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 2023, pp. 1–6.

[12] A.-S. Bulzan, C. Cernazanu-Glavan, and M.-G. Marcu, "Exploring the potential of ocr integration for object detection in invoices," in *2023 27th International Conference on System Theory, Control and Computing (ICSTCC)*, 2023, pp. 149–154.

[13] S. Punde, V. Lomte, P. Pawale, H. Narule, and N. Aijaz, "Survey on object detection techniques based on gpu for marathi character recognition," *IJRAR*, vol. 7, no. 2, pp. 46–56, 2020.

[14] "Haar," https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html, accessed: 2023-07-21.

[15] "Opencv," https://opencv.org/, accessed: 2024-07-03.

[16] "Dlib algorithm," https://medium.com/brasil-ai/mapeamento-facial-landmarks-com-dlib-python-3a200bb35b87, accessed: 2023-07-21.

[17] Micro-ROS., "micro-ros puts ros 2 onto microcontrollers," accessed: 2024-03-10.

[18] LS-LIDAR, "N10 plus, user manual v. 2.0.2," 2022, accessed: 2024-03-10.

[19] DF-Robot., "Esp32-cam development board," accessed: 2024-03-14.

[20] Espressif, "Esp32 wroom32 datasheet."v3.4." 2023, accessed: 2024-03-12.

[21] INSEL, "Turtlebot arm to turtlebot2, catalog product. st. petersburg," accessed: 2024-03-15.

[22] "Gazebo: 3d robot simulation made easy," http://gazebosim.org/, accessed: 2023-06-29.