Robotics & Automation Engineering Journal
ISSN: 2577-2899

Research Article
Volume 6 Issue 1 - May 2024
DOI: 10.19080/RAEJ.2024.06.555677

Robot Autom Eng J

The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs

Matthew Eaton[1]*, Hassan Tanveer M[1] and Farrukh Shahid M[2]

[1]*Department of Robotics & Mechatronics Engineering, Kennesaw State University, Atlanta, Georgia*

[2]*Department of AI and Data Science, National University of Computer and Emerging Sciences, Karachi, Pakistan*

Submission: May 09, 2024; **Published:** May 29, 2024

***Corresponding author:** Matthew Eaton, Department of Robotics & Mechatronics Engineering, Kennesaw State University, Atlanta, Georgia

Abstract

This paper introduces an open-source, modular, cost-effective robotics platform that includes both a robotic arm and a mobile robot chassis. This platform is designed to match the performance of expensive counterparts while maximizing affordability and simplicity allowing for the widespread adoption of modern ROS-based robotics in educational settings. The affordability of this robotics platform not only allows for greater student involvement, it's modularity provides students with a large number of exciting opportunities for Electrical, Software, and Mechanical Engineering labs covering concepts ranging from CAD design, Python, and C++ programming, even integration with frameworks like ROS. By developing an open- source robotics platform that emphasizes price, performance, and simplicity, students have the opportunity to engage in the entire engineering design process, including extensive opportunities for customization. The overall initiative of the paper aims to create a more inclusive learning environment by reducing financial constraints, enabling broader student participation, and contributing to the advancement of robotics education.

Keywords: ROS; Open Source; Mobile Robots; Manipulator; Engineering Education

Introduction

When it comes to educational robotics platforms, especially within k-12 and undergraduate environments, they tend to fall into two distinct categories. The first category is made up of affordable robots that are either of poor quality or lack any usable power or accuracy. While these platforms may be suitable for k-12 environments, they are rarely sufficient for undergraduate engineering environments. The second category is made up of highly capable platforms that offer many useful features but remain prohibitively expensive, limiting either the number of robots an institution is able to acquire or limiting the ability to acquire robots of this quality altogether. This paper proposes a novel robotics platform that offers a middle- ground solution. A modular robotics platform that maintains an affordability close to that of lower-quality robotics kits, while maintaining the performance and accuracy of more expensive robotics platforms. While there are countless design decisions to consider when designing a robotics platform, especially when the goal is to develop an educational plat- form, this project prioritizes a few key elements, affordability, modularity, performance, simplicity, and educational value (Figure 1).

Related Work

During the research and development phase of the project, a literature review was performed concerning the current state of research regarding open-source educational robotics platforms. During the literature review process, there were several interesting projects encountered, many of which contained aspects of what made up the research objectives of our proposed platform. One such project was that of a journal article published in IEEE Access entitled, ”An Inexpensive Autonomous Mobile Robot for Undergraduate Education: Integration of Arduino and Hokuyo Laser Range Finders”. In the article, a team of researchers from the Department of Mechanical Engineering at the National

Defense Academy of Japan Kanagawa outlines the development of an educational robot meant to serve as a platform to guide undergraduate mechanical engineering students through a 10-week course that covers topics such as control engineering, robot programming, and embedded systems [1]. While the system proposed no doubt serves the needs of the undergraduate course it was developed for, it's relatively high price to performance ratio, along with it's more niche audience represents a distinct departure from the more general-purpose platform proposed in this body of work.

In recent years, there have been many other attempts at creating general-purpose high-performance ROS-based robotics platforms for undergraduate education. One such project, published through The American Society for Engineering Education, entitled, ”Herbie: A Platform for Robotics Re- search with Undergraduate Students, Campus Engagement through Social Media, and Building Interest in STEM” [2] outlines the development of a ROS-based educational robotics platform for the use of computer science departments at California Polytechnic State University in San Luis Obispo. Their platform, nicknamed ”Herbie” was developed around a computing platform developed by the Nivida Corporation that they call the the Robotics Teaching Kit. This kit was con- structed around the Nvidia Jetson TX2 embedded computer, and was sent to various engineering institutions around the world to assist in developing robotics courses using the Nvidia Platform. While the Herbie platform achieved it's goals by serving as an exciting introduction to robotics, it's reliance on pre-built examples provided by Nvidia means that there are significant steps in the robotic system development process that are glossed over. One of the largest goals in the proposed development of this project is to instruct students in each and every step of the robotic systems development process, from CAD to high-level ROS-based manipulation.

One of the most impressive educational platforms encountered during the literature review portion of the research project was that of the PyRobot program. As outlined in the paper, ”PyRobot: An Open-source Robotics Framework for Research and Benchmarking” [3], a group of researchers from Carnegie Mellon University have developed the impressive PyRobot program. With a mobile base, stereo depth camera, and four DOF robotic manipulator, it is a well-documented and highly capable platform. Much like the proposed platform documented in this body of work, the PyRobot is ROS- based and fully compatible with common ROS-compatible frameworks such as MoveIt, RVIZ, and Gazebo. Additionally, this project is also completely open source, with all relevant documentation readily available. While the PyRobot platform is certainly impressive, there are two key differences between this platform and the platform outlined in this paper. The first of which is price, the average cost of the Pyrobot is close to $5000, close to 5x the target price of the platform discussed here. Secondly, the PyRobot is designed specifically for indoor operation. One of the hallmarks of the platform outlined in this paper is the proposed robotic systems' ability to function effectively in both the lab and classroom, as well as in outdoor environments. This was selected as a design criterion to ensure that the platform served as a viable option for the widest range of educational situations possible.

Research Objectives

In terms of affordability, the subjective nature of the term makes it difficult to define. For the purposes of this project, relative affordability can be defined as between $500 and $1000. While a $1000 price tag may seem excessive in certain contexts, when considering that high-level robotics platforms such as the turtlebot [4] platform without a manipulator is priced at $2500.00, the popular husarion rosBot2.0 [5] platform is similarly priced at $3049.00 the target price remains significantly lower than most available alternatives. Additionally, the modularity and open-source nature of the platform assist in the platform's affordability. Organizations have the ability to choose from a wide variety of alternatives for aspects of the platform such as using acrylic or plywood, stepper, and DC motors, as well as the choice of single board computer (SBC). Decisions regarding these aspects of the project can have a dramatic impact on the overall cost of the project.

Given the wide discrepancy of needs in educational institutions across the globe, achieving a high level of modularity was a key component of this project. One of the first challenges was developing a robotic arm and rolling mobile robot chassis that have the ability to operate both in tandem, as well as independently. With the goal of achieving maximum modularity, an additional design constraint was introduced, this constraint centered around the idea that all of the com- ponents in the design needed to be easily obtained through common educational vendors. This additional constraint meant that not only would parts be easily obtained, but that many alternatives would be available in case of increased prices or lower availability from a specific vendor. An additional design constraint observed during the research and development phase of the project was remaining hardware agnostic. By maintaining modularity at the computational level, the project would maintain usefulness for the widest range of educational institutions possible. This robotics platform is designed to have the ability to be fully controllable with a simple microcontroller. For testing purposes, this project was developed using the highly capable ESP32-S3 [6], chosen for its low cost, and high performance, as well as built it WiFi and Bluetooth. With modularity in mind, those dedicated to more popular platforms such as the Arduino ecosystem can easily adapt the project to be used with microcontrollers such as the Arduino Mega 25600 [7]. Not only is this project compatible with a wide range of microcontrollers, but it is also designed to work in tandem with popular SBCs such as the Raspberry Pi [8], or LattePanda Delta 3 [9]. This means that institutions that wish

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

to interface the robotics platform with software packages such as ROS [10], ROS2 [11], OpenCV [12], Open3D [13], and others, have the ability to conduct high-level computation operations on the SBC and communicate with the microcontroller, and educational institutions that cater to a younger audience can utilize a more basic microcontroller for motor control.

When considering the goal performance of the platform, there were a number of criteria to consider. With respect to the manipulator aspect of the project, one of the main goals was to achieve a repeatability of +- .05 in. and a level of accuracy of less than 1 degree. This meant that the manipulator would need to be powered by stepper motors as opposed to more simple and affordable servo motors. During the testing process, the manipulator for this project was powered by a set of Nema 17 Stepper motors, paired with 50:1 harmonic gearboxes. The use of harmonic gearboxes allowed the design to achieve a combination of accuracy, overall simplicity, and strength, previously unattainable with more traditional stepper motors. When determining the goal performance of the rolling mobile robot platform, one of the most important factors was the robot's ability to operate in a wide range of educational environments. One of the stretch goals of the project was to develop a platform that could operate efficiently in both indoor and outdoor environments. Achieving this meant that the torque and accuracy of the platform were more heavily favored over speed. The rolling mobile robot chassis is propelled by 12v 555 style DC gear motors with built-in encoders and a 150:1 ratio. This combination allowed us to maintain a balance of affordability, torque, and accuracy.

Simplicity was also a key criterion considered in the development of this modular robotics platform. By emphasizing simplicity, we were able to ensure that not only would the platform be simple enough to assemble and that the task could be completed by students using simple hand tools, but that by utilizing simple flat laser cut pieces, customization by students or institutions would be as simple as importing the open-source. DXF files into their CAD program of choice. This platform was designed to be able to be manufactured using standard makerspace equipment and be assembled, customized, and adapted easily by a wide range of student groups to meet the wide range of needs of educational institutions around the world.

Finally, the last criterion considered when designing this modular platform is arguably the most significant in developing an educational robotics platform, the educational opportunities the proposed robotics platform helps facilitate. Robotics education has the unique opportunity to offer stu- dents instruction in a wide range of fields within mechanical, electrical, and software engineering principles, and the robotics platform developed during the course of research for this project is no exception. Given the modularity of the platform, as well as the ability for the manipulator, and mobile base to be used independently, the platform as a whole has the ability to support as much as a semester's worth of content. From lessons in CAD, ROS, C++, Python, Matlab, Arduino, Strength of Materials, Kinematics and Dynamics, Circuity, and PCB design, the possibilities are almost endless (Figure 2).

Figure 1: An affordable educational robotics platform.

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

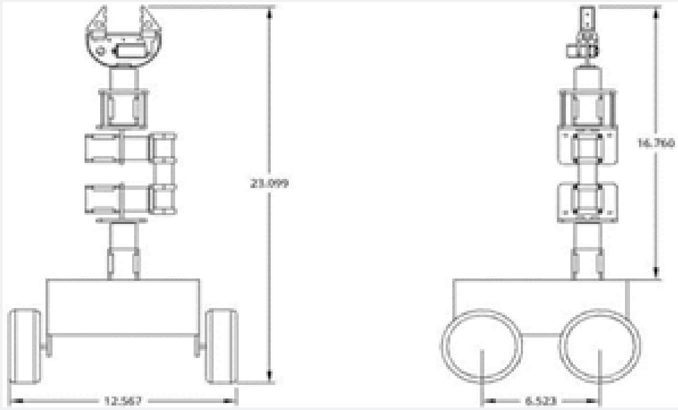

Figure 2: CAD model of educational robotics platform.

Physical Hardware

The robotics platform developed during the research phase of this project is made up of two independent, but connectable pieces. The first is a powerful and accurate stepper motor- based robotic arm, and the second is a mobile robot base, powered by highly capable and torque-focused DC motors. There were several important criteria observed when selecting the materials that make up the chassis of both the robotic arm and mobile base, such as affordability, ease of acquisition, strength, and accuracy.

The manipulator portion of the project is constructed from laser-cut 5mm acrylic sheets, braced with corner brackets and common brass standoffs, as well as a piece of 2020 aluminum extrusion. These materials were selected due to their rigidity, widespread availability, and relative affordability. The actuators for this project consist of standard Nema 17 stepper motors, which can be commonly found fitted with either planetary or harmonic gearboxes. The stepper motor drivers selected for this project are the popular TB6600 modules, chosen for their effective cost-to-performance ratio.

The mobile base portion of the project is also constructed primarily of laser-cut 5mm acrylic sheets and 2020 aluminum extrusion. The motors selected for the testing phase of this project were 12v 555 style DC Gear Motor with a 150:1 gear ratio, providing a velocity of 40 RPM at 12v, and a rated output torque of 5.6 kg-cm. The motor driver selected for this project is the Cytron MD20-A motor driver [14], with a max current of 20 A. It provides more than enough headroom to ac- commodate heavy load operations being performed by the DC gear motor. This particular motor driver represents a slightly more premium component and can be easily substituted for a more economical H-Bridge style motor driver. The batteries selected for this project consist of a bank of Lithium-ION 18650 batteries. These were selected for their excellent energy density to weight ratio, as well as their wide availability. The batteries are configured into 2 groups of 4 cells using a custom battery pack PCB. This provides an average of 16v per battery pack, which is dropped down to 12v through the use of one XL4015 buck converter per battery pack, allowing for power delivery of up to 4 Amp of current per battery pack.

Computing Hardware

One of the most important aspects of this design from a computational perspective was remaining hardware agnostic. There are two main components that make up the primary computational platform for this robotics platform. The first is an SBC, which could very easily be a Raspberry Pi or any other alternative. In this case, testing was completed using the Latte Panda Delta 3. While this SBC is considerably more expensive than a typical Raspberry Pi, it is proportionally more powerful. This SBC is powered by an x86- based 11th generation Intel N5105 Processor, with a base CPU frequency of 2.9GHz, and 8 GB of RAM. It offers tremendous performance, especially when processing detailed ROS-based simulations, computer vision applications, algorithmic tasks, and other computationally intensive operations. One of the more interesting features of the Latte Panda Delta 3 is its onboard Atemega32u4 co-processor, which is hardwired to the SBC circuit board, and designed to be communicated via UART. This creates a simple workflow for coordinating complex graphical and computationally heavy tasks that are best run on the x86-based CPU, and tasks such as PWM signal generation for motor controller applications that are perfect tasks for the AVR-based Atmega32u4 as seen in figure 3.

One downside to using the Latte Panda Delta 3's onboard Atmeaga32u4 microcontroller is the number of available PWM-capable GPIO pins. In this case, there are 7. When operating the drivetrain of the mobile base, or the manipulator independently, this is more than enough. However, in order to ensure that the platform is compatible with SBCs that may not have a built-in co-processor, as well as have the ability to wirelessly manipulate the system without the use of an SBC if desired, this project was designed to offer the option of fully controlling the platform using custom open-source co- processor based on

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

the ESP32-S3 microcontroller. There are many variations of the ESP32 microcontroller. For this project, the variation of choice was the ESP32-S3. ESP32-S3 is a dual- core XTensa LX7 MCU, with a maximum frequency of 240 MHz. Due to the ESP32-S3's built-in WiFi and Bluetooth capabilities, 45 available GPIO pins, up to 8 PWM capable pins, and multiple available communication protocols, the entire robotic system can be programmed and controlled using the co-processor as the primary computing device if desired. The PCB designed for the project is relatively simple. This is thanks in part to the fact that the ESP32-S3 has, for the first time on an ESP32-based device, native USB communication built-in, and does not need a secondary USB to UART communication chip as is commonly necessary on microcontroller development boards. A majority of the PCB is dedicated to creating efficient connections for the TB660 stepper motors, and Cytron DC motor drivers. It is worth mentioning as well that any standard ESP32-based development board or other microcontroller with sufficient PWM capable pins could also be used as the primary computing device for the platform, and the custom ESP32-S3 co-processor board is not essential to the project's success. Having the ability to support a wide range of computing platforms was a crucial design constraint observed over the course of the project (Figure 4).

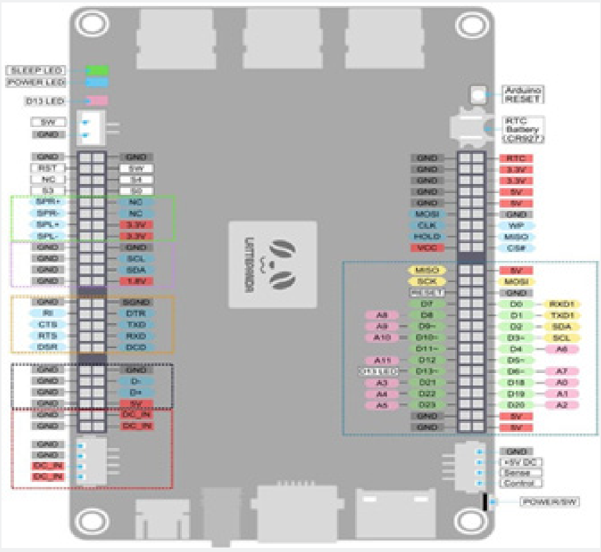

Figure 3: LattePanda Delta 3 Pinout.

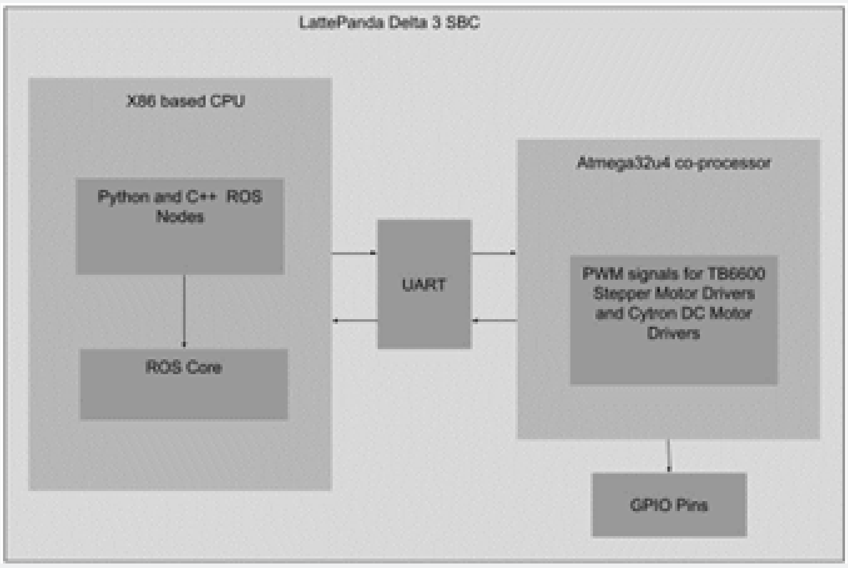

Figure 4: LattePanda Delta Based Control Flow Chart.

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

Gathered Data

During the testing portion of the project, both the mobile base and manipulator were evaluated while completing tasks in several indoor and outdoor environments. One of the primary goals of this project was the development of a robotics platform that serves as an affordable and effective educational and research tool. With that in mind, establishing consistent and successful operation of the robot in a wide variety of contexts was crucial (Figure 5).

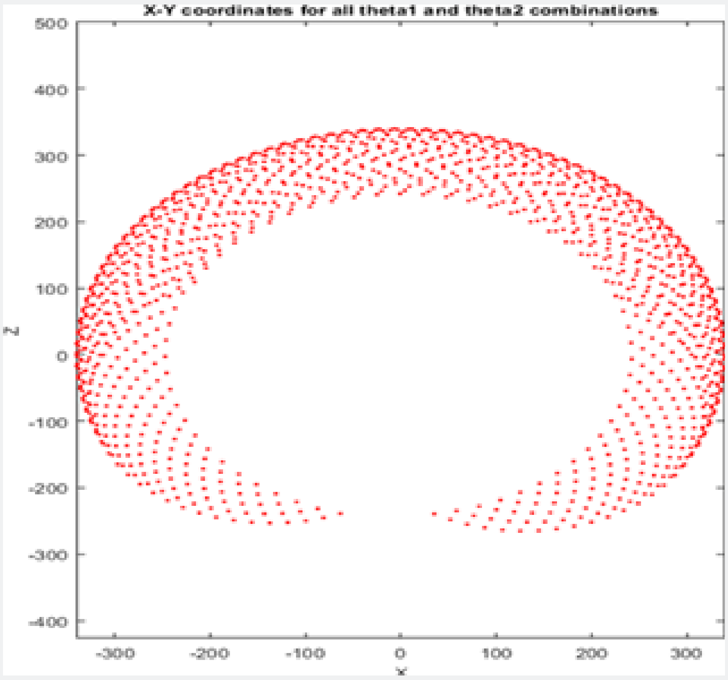

Figure 5: Range of Motion of the manipulator.

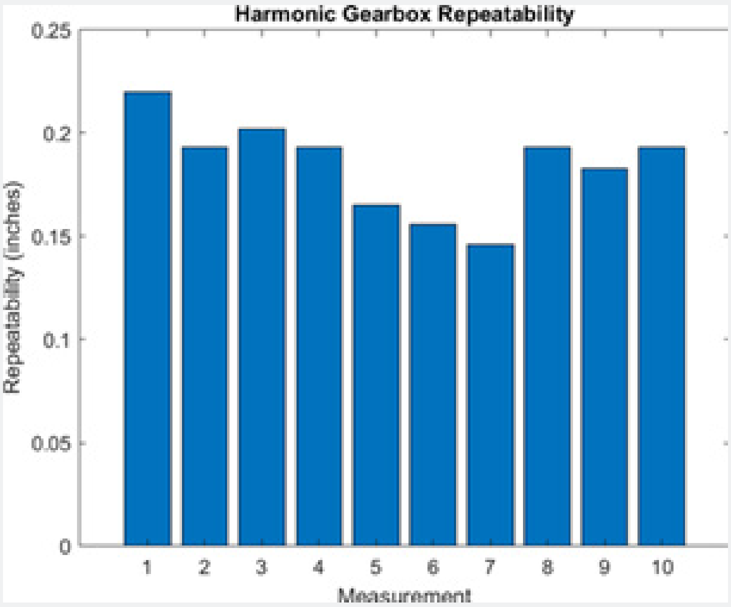

Figure 6: Harmonic Gearbox Stepper Motor Repeatability.

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

Choosing the right actuator is a crucial stage in the development of a robotics platform. In the testing of the manipulator portion of this project, both planetary and harmonic gearbox stepper motors were compared for use in the manipulator portion of the project and were compared by gearbox backlash, and mechanical repeatability. According to the manufacturer's specifications, the planetary gearbox stepper motors initially tested have a backlash tolerance of approximately 1.5 degrees or 5400 arcseconds, resulting in a larger displacement at the end effector than may be required in certain applications. This level of backlash, while negligible at short distances compounds at each robotic joint, resulting in a significant lack of accuracy for a fully constructed robotic arm with multiple joints. In contrast, the stepper motors equipped with harmonic gearboxes have a backlash tolerance of only 20 arcseconds, or 0.0055 degrees, representing a 27136.36% increase in accuracy. However, this increased accuracy comes at an in- creased cost of 2x to 3x, raising the price per actuator from approximately $50 to just under $150. The next item tested between the planetary and harmonic gearbox stepper motors was repeatability. This was tested using a digital dial indicator on a custom test jig to ensure stability and rigidity. The first motor tested was the harmonic gearbox motor. In a trial of 10 iterations, the stepper motor produced a repeatability of 0.053 inches. The planetary gearbox motor was tested using an identical testing environment and produced a repeatability of 0.133 inches. With this data in hand, the harmonic gearbox stepper motors were chosen as the primary actuators for the manipulator portion of the project (Figure 6).

The selection of proper motors for locomotion is in many ways just as important as stepper motor selection for a manipulator. With the inclusion of the manipulator, related electrical hardware including stepper motor drivers, and additional batteries, the entire weight of the system is not insignificant. With this in mind, the performance of the rolling chassis was tested in a variety of environments, completing navigation applications in both indoor and outdoor settings. The motors selected were biased toward torque rather than top speed. That being said, in an indoor environment, the mobile base with the manipulator attached was able to navigate linearly at a speed of 1 meter per second and rotate at a rate of approximately .5 meters per second. This data suggests that not only were the selected motors capable of propelling the system with enough torque, they were able to do so at a reasonable speed. During testing in an outdoor environment, similar performance was observed. The system as a whole was able to navigate on both grass and pavement environments of a variety of grades of steepness without issue.

Microcontroller Based Actuation

One of the most interesting aspects of this robotics platform is its hardware-agnostic approach to computing platforms. While the main purpose of this robot is to conduct student labs centered around ROS and its associated frameworks and software, the entire platform can be controlled without the use of a single board computer, relying solely on the ESP32-S3-based Co-Processor board. With the ESP32's integrated WiFi and Bluetooth module, the potential methods of input for the robotics platform are numerous. For testing purposes, while relying solely on the ESP32-S3 as the computational platform, the system was connected to a SONY DualShock 5 controller [15]. Despite being designed specifically for use with the Play station 5 video game console, the DualShock 5 is well equipped to act as a remote-control device for this robotics platform (Figure 7).

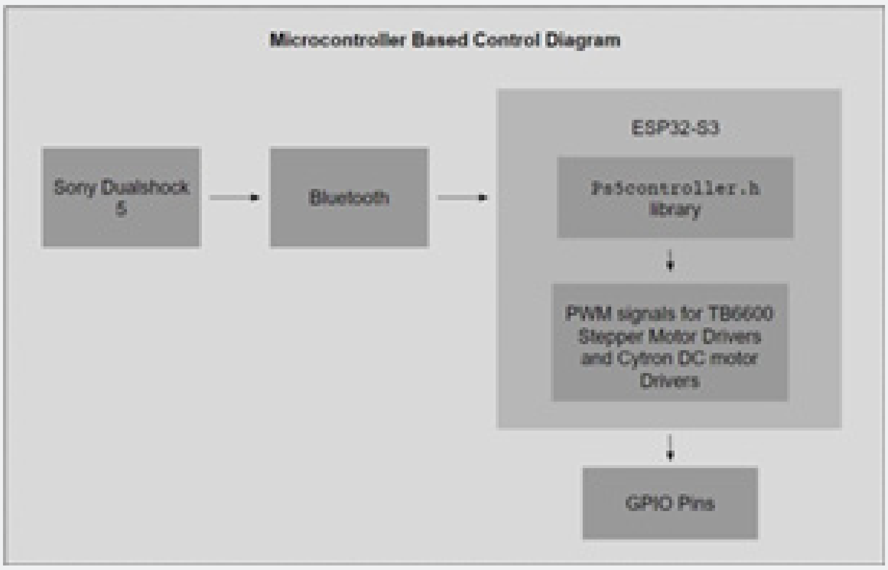

Figure 7: Microcontroller based actuation flowchart.

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

During the testing process, the ESP32-S3 was programmed using the Arduino IDE. Within the Arduino sketch for the system control, the first step in the process is parsing the joystick and button data using the ps5Controller.h ESP32 library [16]. Once the joystick data is properly analyzed, the control phase is implemented. During the control phase, joint position variables and velocity variables are incremented and decremented based on the position of the joystick data. Additionally, button data was mapped to values generated from button data, specifically the "R1" and "L1" buttons to open and close the gripper, and the "R2" and "L2" buttons to rotate the orientation of the gripper (Figure 8).

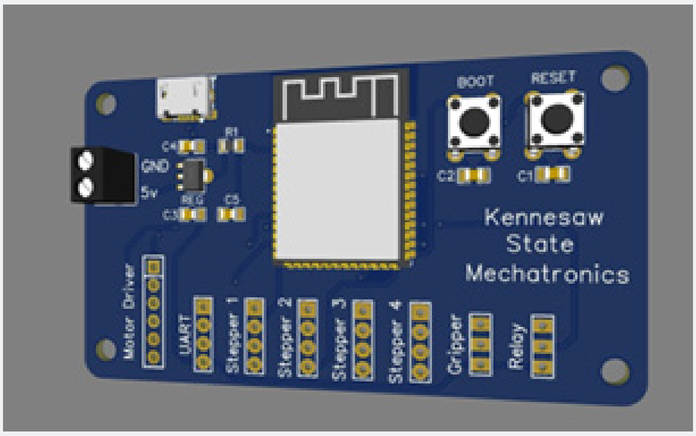

Figure 8: Custom ESP32-S3 Co-Processor.

ROS Integration

The use of a single board computer with this robotics system allows this platform to realize its full potential by manipulating the mobile base and manipulator using ROS. This opens the platform up to near-limitless research opportunities. The first step in the process of fully integrating the platform with ROS is the generation of a suitable URDF (Universal Robot Description) file. Given the goal of creating an educational robotics platform that instructs students regarding as much of the robotics engineering development process as possible, the instructional material for this project begins with CAD software. For this project, the CAD program selected was SolidWorks [17]. SolidWorks was selected for this project due to both its popularity in industry and engineering educational facilities, as well as its useful third-party swurdf exporter plugin. There are several steps required to generate a suitable URDF file from SolidWorks. The first step in the process is creating a SolidWorks assembly of the entire robotic system, complete with moving joints that behave exactly as they would for the physical robot. A URDF file uses the concepts of joints and links to allow various software to understand the physical properties of a given robotic system. With that in mind, the second step in the process is adding in reference geometry to the SolidWorks assembly such as points, axes of rotation, and coordinate systems for each relevant joint. Once the axes and coordinate systems are in place, the swurdf exporter plugin can be used to generate the URDF file that can then be imported into a ROS environment. Not only does the plugin generate a URDF file, it also generates a basic ROS package that can be used to verify that the URDF data, and 3D mesh files that make up the robot have been exported correctly.

The next step in the setup process is transferring the ROS package generated by SolidWorks to the Linux-based ROS environment being used for the robotics platform. For this project, the ROS environment used was configured within the Ubuntu 20.04 operating system installed directly onto the CPU of the LattePanda Dela 3 SBC used to support the project. Once the ROS package has been transferred to the ROS environment, the package must be compiled in the SRC folder of the catkin workspace being used by the ROS environment. The ROS package containing the URDF file for the system includes two built-in launch files used for testing purposes. The file "demo.launch" will open up RVIZ and display the robot. The file "display.launch" will also display the robot, this time in a Gazebo environment. While both are useful, it is worth noting that these launch files do not contain the code needed to control the joints of the robotics simulation, or the physics data needed to accurately represent the physical properties of the robot, and should be used primarily to verify the correct exportation of the robotic system model from SolidWorks.

Once the model has been verified, the URDF file must be modified to include software controllers for the various joints involved in the model. While it is often wise to write custom

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

controllers for more advanced robotic systems within a ROS environment, this platform was designed using two built-in controllers commonly utilized in ROS based robotic systems. The first is the general "libgazebo ros control.so" controller script for the manipulator's joints, and the second is the "libgazebo ros skid steer drive.so" controller to manage the four joints of the mobile base. This skid steer controller will allow the gazebo model to receive twist messages being published to the cmd vel topic, and accurately reflect the movement of the physical robot based on the received inputs (Figure 9).

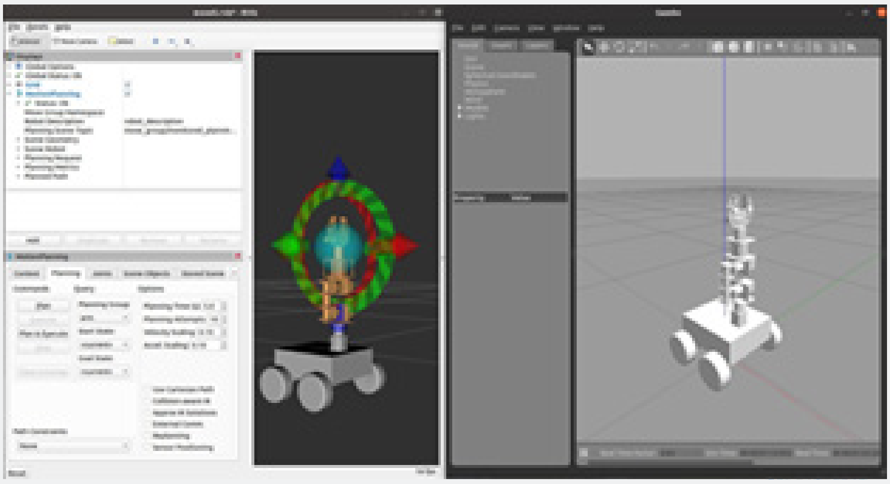

Figure 9: Full Robotic System within a ROS environment using RVIZ and the Gazebo Physics Simulator.

The next step in the process is passing the newly edited URDF file through the moveIt setup assistant package [18]. MoveIt is a popular ROS framework dedicated to advanced control techniques for robotic manipulators. Not only is it compatible with a wide range of popular commercial robotic arms, it is also fully capable of integrating custom robotic manipulators such as the one being used within this project. Instructions for the installation of MoveIt can be found on the software's wiki page. Once the setup assistant is launched there is a number of parameters that must be selected to complete the setup process. This includes adding additional joint controllers, calculating a self-colision matrix, establishing default poses for the manipulator, and others. Once the process is complete, a new ROS package is generated that contains a number of sample scripts, including a launch file that will open both an RVIZ window for controlling the pose and orientation of the manipulator, as well as a gazebo window that displays the full simulated robot in its entirety.

With the simulation complete, the final step is writing software to control the robot. During the testing phase of this project, several basic packages were written to demonstrate the basic manipulation of the mobile base and manipulator, as well as a sample package to achieve teleoperation of the system from any remote computer system a user might wish.

Teleoperation

Efficient teleoperation is a crucial capability of any modern robotic system. Thankfully, the use of a powerful single-board computer such as the Latte Panda Delta, allows this platform to serve as a capable teleoperation platform by manipulating the mobile base and manipulator from a remote computer. In fact, thanks in part to the onboard M.2 socket on the Latte Panda Delta 3, adding an additional 4G or even 5G module means that the teleoperation range of the system is not limited to areas with WiFi coverage. While remote communication can be achieved in a number of ways, one of the goals of this project is to allow the project to expand and grow as educational institutions conduct research. With that in mind, a sample package was written to allow the use of the AYN Loki handheld PC as a teleoperation device for the platform [19]. It is important to note however that teleoperation is also possible using any conventional desktop or laptop computer. This particular handheld PC was selected for testing due to its portability, screen size, and integrated control joysticks and buttons.

These features together make the device ideal for robot teleoperation. The communication platform for this phase of the project was achieved using the Windows RDP protocol (Remote Desktop Protocol). While the RDP protocol is not designed to transmit joystick data, it does offer robust support for transmitting keyboard and mouse data. To address this issue, a custom Python script was written to convert the system's embedded joystick data into keyboard presses, allowing the data to be easily transmitted to the SBC located on the robot. Additionally, by using the RDP protocol a user is able to remotely view important robot data such as camera feeds, LiDAR data, or any other relevant information received from ROS or other sensors (Figure 10).

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

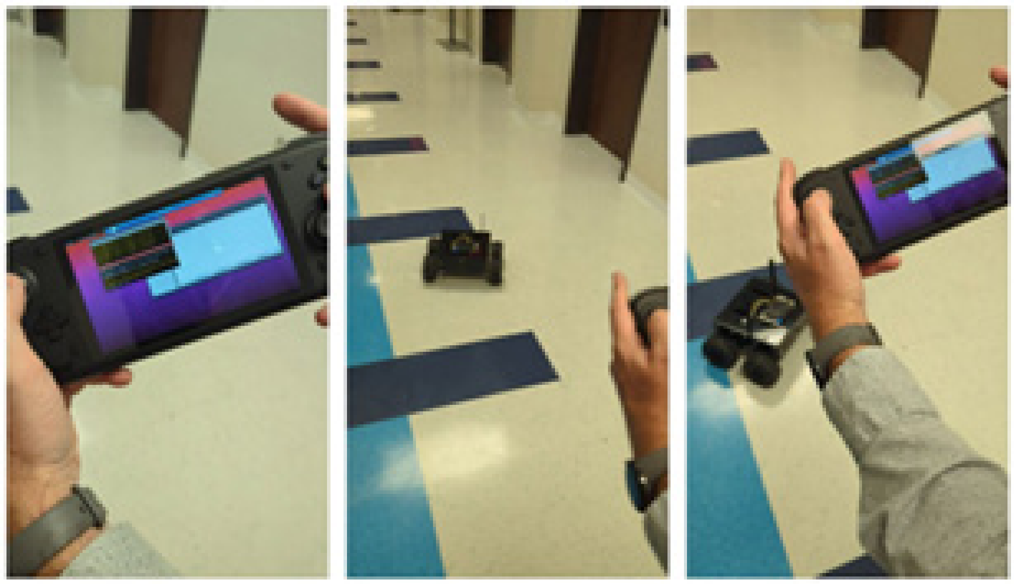

Figure 10: AYN Loki as a ROS Based Teleoperation Device.

Educational Material Developed

One of the primary goals of the project was the development of an educational robotics platform that supports undergraduate engineering education. With this goal in mind, a series of instructional videos have been developed that guide students through each and every step in the development process. These videos not only guide students through the steps necessary to set up the robotics platform outlined in this document but also provide them with the foundational information needed to develop future custom robotics systems as they progress through their engineering education.

In addition to the series of instructional videos, an open- source GitHub repository has been developed that provides students with all of the source files needed to re-create, customize, and adapt this system to any needs a student may have. The repository contains not only the appropriate ROS packages needed for simulation, but also the manufacturing files, SolidWorks assemblies, and wiring diagrams needed to complete the project https://github.com/ACBRrobotics/MTREbot.

Future Work

Given the educational nature of this project, the next steps in the development of this robotics platform center around further taking advantage of the vast number of resources available within the ROS environment. While teleoperation- based control was explored thoroughly during the scope of the project, a significant milestone will be achieving a higher level of autonomous capability. This will allow the platform to serve as a useful method of researching and demonstrating topics such as the efficacy of various planning algorithms such as A*, D*, Dijkstra's, and others. Alongside developing path planning capabilities, there exists a great deal of research potential regarding the integration of more advanced sensing devices such as LiDar. Additionally, while the platform currently contains a camera for teleoperation applications, there is a tremendous educational opportunity in developing undergraduate robotics labs centered around custom computer vision applications for the platform's camera.

Conclusion

Exploring robotics offers students a diverse array of skills and experiences, making it an excellent choice for engineering-based education settings. While there are numerous examples of small hobby-grade mobile robot platforms that lack usable levels of strength and accuracy when compared to industrial robots, as well as many impressive larger-scale professional robotics platforms that may offer high levels of performance but remain prohibitively expensive, there exists the need for the development of an intermediate solution. In an effort to address this issue, we have developed an open-source robotics platform with a reasonably affordable construction cost, boasting high accuracy and broad compatibility with various control systems-from basic microcontrollers like Arduino and ESP32 to sophisticated platforms like ROS and MoveIt. A robotics platform with the ability to lead educators through empowering students to engage fully in every aspect of the engineering design process, including CAD assembly, material selection, mathematical modeling, programming, and integration with industry-relevant software systems.

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677

References

1. Ueyama Y, Sago T, Kurihara T, Harada M (2022) An inexpensive autonomous mobile robot for undergraduate education: Integration of arduino and hokuyo laser range finders. IEEE Access 10: 79029-79040.
2. Bonilla A, Jones DC, Krysl A, Seng JS (2020) Full paper: Herbie: A platform for robotics research with undergraduate students, campus engagement through social media, and building interest in stem, in Proceedings of the 2020 ASEE PSW Section Conference, canceled, 2020.
3. Murali A, Chen T, Alwala KV, Gandhi D, Pinto L, et al. (2019) Pyrobot: An open-source robotics framework for research and benchmarking. arXiv preprint arXiv: 1906.08236.
4. Robotis - e manual (2024) Robotis.
5. Husarian - rosbot (2024) Husarian.
6. Espressif - esp32-s3 (2024) Epressif.
7. (2024) Arduino - arduino mega2560.
8. Home – raspberrypi (2024) The Raspberry Pi Foundation.
9. home – lattepanda (2024) Latte Panda.
10. Ros wiki - documentation (2024) Open Source Robotics Foundation
11. Ros2 wiki - documentation (2024) Open Source Robotics Foundation.
12. home - opencv (2024) OpenCV.
13. (2018) I. S. L. Org, "isl-org/open3d. GitHub, 2018.
14. home - cytron (2024) Cytron.
15. home - sony us (2024) Sony.
16. (2018) rodneybakiskan, "ps5-esp32." GitHub, 2018.
17. home - solidworks (2024) Dassault Syste`mes - SolidWorks Corporation.
18. home – moveit (2024) PickNik Robotics.
19. Ayn - loki (2024) AYNtec.

DOI: 10.19080/RAEJ.2024.06.555677

How to cite this article: Matthew E, Hassan Tanveer M, Farrukh Shahid M. The Design and Implementation of a Cost-effective Educational Robotics Platform for affordable ROS Based Mechatronics and Robotics Labs. Robot Autom Eng J. 2024; 6(1): 555677. DOI: 10.19080/RAEJ.2024.06.555677