Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #46433

Novel Testbench and Controller for Teaching Python and Robotics in Mechatronics Engineering Education (Complete Paper)

Dr. Mohamed Gharib, Texas A&M University

Dr. Mohamed Gharib is an associate professor and program chair for the Mechatronics Engineering Technology and STEM Education programs at the School of Engineering at Texas A&M University. He is an expert in designing, prototyping, modeling, and simulation of robotic systems. Also, he is a STEM education specialist and program developer, including planning, developing, integrating, and teaching STEM programs for K-12 students through university outreach.

Joshua A Jalomo, Texas A&M University
Gabriel Tijerina, Texas A&M University

Graduated Masters of Science in Engineering Technology engineer. Background in mechatronics and autonomous robotics. Experienced in fast-paced and hands-on environments.

Novel Testbench and Controller for Teaching Python and Robotics in Mechatronics Engineering Education

Abstract

Python is extensively used in different engineering fields due to its versatility, simplicity, and a wide range of libraries tailored for engineering applications. Python's extensive ecosystem of libraries, combined with its ease of use and flexibility, makes it a popular choice for mechatronics education development across various domains, including industrial automation, autonomous vehicles, drones, and humanoid robots. In engineering colleges, the gap between teaching coding and implementing it on existing hardware affects the students' interest in advancing in programming and their understanding of the unlimited capabilities of Python. Integrating experimental sessions in engineering colleges can bridge this gap and align raw coding concepts with real-world applications.

This paper presents a novel low-cost robotics testbench (RTB) and Raspberry Pi-based robot controller (PRC) for teaching Python with application to robotics in mechatronics education. The robotic testbench, controller, and associated experiments enable the students to learn Python's foundation while operating several actuators and sensors. The testbench and controller were designed, built, and tested for use in a hands-on robotics course for sophomore engineering students, which requires extensive prototyping of robotic mechanisms. In the laboratory session associated with Python programming, the students learn how to operate and control different types of DC motors, stepper motors, servo motors, and linear actuators. Then, the students use the testbench to collect and export data using tracking and measurement sensors. The testbench consists of frames, a collection of motors and sensors. The controller was developed to operate six DC motors, five servo motors, and four sensors. Incorporating this testbench and experiment into mechatronics engineering courses will increase student engagement and facilitate a deeper understanding of coding using Python. The developed testbench and controller are low-cost, portable, and straightforward to replicate. This paper presents the testbench design, bill of materials, and case studies of experiments and measurements.

1. Introduction

Mechatronics engineering integrates mechanical, electrical, and computer engineering to design and develop sophisticated electromechanical systems. This multidisciplinary approach empowers engineers to create efficient, intelligent machines, robots, and automation systems that are transforming industries such as manufacturing, automotive, healthcare, and aerospace. As technology advances rapidly, mechatronics engineering is increasingly essential for addressing complex challenges and meeting the dynamic demands of the global market [1,2].

The high cost of training equipment in engineering education is a significant challenge, particularly in fields like mechatronics engineering. Advanced tools, including robotics kits and simulation software, demand substantial financial investment, which can limit access to quality training, especially in underfunded institutions or developing regions. Additionally, the ongoing maintenance and upgrade expenses place further strain on budgets. Despite these obstacles, investing in modern training equipment is vital for providing students with the required hands-on experience and industry-standard tools necessary for success in the workforce. Overcoming financial and safety challenges requires the innovative development of affordable, user-friendly training solutions that balance cost-efficiency with effective learning outcomes [3,4].

Developing in-house, low-cost engineering equipment for educational purposes offers several advantages, including significant cost savings, greater customization, and enhanced control over design and production processes. This approach fosters innovation, supports hands-on learning, and accelerates prototyping and iteration. Additionally, it allows for tailored solutions that address specific educational needs, optimize resource utilization, and enhance safety standards. While existing literature showcases many successful examples of low-cost engineering training equipment, there are limited studies specifically focused on the in-house development of training setups for mechatronics engineering [5,6].

In the work of [4], the authors made an interactive testbench to explore and demonstrate the field of mechatronics. The testbench contains components like actuators and sensors. The authors surveyed what students thought of the testbench and concluded that it was more interactive. The testbench is prominent in all dimensions, occupying one student’s table space. This limits the number of students that can use it at once and, therefore, the amount of interaction every student gets from it.

In the work of [7], the authors designed, built, and tested a low-cost mobile robot that uses 2 DC motors that were removed from battery-operated screwdrivers. The motivation behind this project was to make a simple yet effective robot to mass produce and lend to engineering students so they can program them and use them in their projects/classes/competitions. This project proves that there is a demand in the educational field to learn and participate in programming mobile robots to the extent of using any components available.

In the work of [8], the authors designed and 3D printed a small line following a mobile robot that uses low-cost electronic components like servo motors and an Arduino Uno. The robots use sensors to detect a line and follow it to the end of the path. The robot's structure was 3D printed, making it small and very limited in its capabilities. The electrical components are also placed freely on top of the robot. This project demonstrates that there is a desire to program robots and creates an opportunity to do that satisfyingly, but it can only be used for line following purposes.

In the work of [9], the authors examined three courses in the mechanical engineering degree plan and made one unified set of materials that can be used by the students for all three courses. They also created an online repository with helpful material and examples. A survey was taken for the students who had to buy a new kit for each class and the ones who just bought one kit and used it for all three classes, and the results show that the students who used the same kit for each class felt more confident in their electrical and programming capabilities. The only downside is that every student needs to buy the kits and learn how to use them independently.

In the work of [10], the authors design and fabricate a testbench for a mechatronic system. The lab demonstrates multiple PID control systems for a setup with a DC motor, 2 springs, and 2 pulleys. Using MATLAB and Simulink, the students could put their theorical knowledge into practice to learn what the actual experimental results look like. The testbench is primarily 3D printed, with the electronics hidden below and the experimental setup presented on top. They used an Arduino Due, a motor driver, and a custom board to be super compact and user-friendly. The majority of the code is given to the students so that the bulk of their work is to implement the control system and, therefore, learn the actual course content for the lab session.

This paper introduces an innovative, low-cost robotics testbench (RTB) and Raspberry Pi-based robot controller (PRC) designed for teaching Python, focusing on robotics in mechatronics education. The design of the RTB and PRC are described in detail to provide an accessible pathway for learners and educators. The RTB, PRC, and the accompanying experiments allow students to grasp the fundamentals of Python while interacting with various actuators and sensors.

The RTB and PRC were developed to support the laboratory of a new course, MXET 250 (Robotic Systems Design) for the Multidisciplinary Engineering Technology (MXET) Program in the Department of Engineering Technology and Industrial Distribution (ETID) at Texas A&M University.

The rest of the paper is organized as follows. In sections 2 and 3, the RTB and PRC design are described, respectively. Section 4 provides a brief overview of the developed lab manual. In section 5, samples of the students' projects are presented. Finally, conclusions and learning outcomes are provided in section 6.

2. The Robotics Testbench (RTB) Design

The motive behind the developed testbench is to provide a "static robot" for the students to enable them to learn programming using Python and test their codes without the need for extra space or moving in the lab. The developed testbench includes the same components to be mounted to a moving robot but mounted in a stationary training system. This will ensure the safety of the students while saving a significant amount of time compared to testing a robot on the ground.

The developed testbench is based on off-the-shelf components with minimum required fabrication. The design is modular, so it can be rearranged as needed for the target course and audience. A heavy-duty frame is designed and built using aluminum extrusion and c-channel structural elements from REV Robotics. A compatible assortment of brackets and fasteners are acquired from the same vendor. Custom-made 3D-printed holders are fabricated to carry the sensors while compatible with the framing system. PVC pipes and fittings are selected to provide housing and pathways for the wires. This ensures that the wires are protected and remain in place with continuous use of the testbench. Holes were drilled in the PVC pipes to allow the pipes to be mounted to the frame. A 3-mm acrylic sheet is placed on the top of the frame base to provide an area for mounting the controller. Figure 1 shows the frame assembly and its basic components.

Having assembled the stable frame, multiple DC motors, servo motors, and sensors were acquired and mounted to the frame. The DC motors come with power and encoder wires. Extension wires for the servo motors and sensors are acquired to fit the frame dimensions. A multiplexer (MUX) breakout is used to enable communication with multiple sensors (I2C), which have the same address. Figure 2 shows the final testbench with all motors and sensors mounted to the frame.

The cost of one setup of the testbench is about $500 and is suitable for 2 students. Table 1 lists the essential components of the testbench. The testbench can be customized as needed according to each course needs.

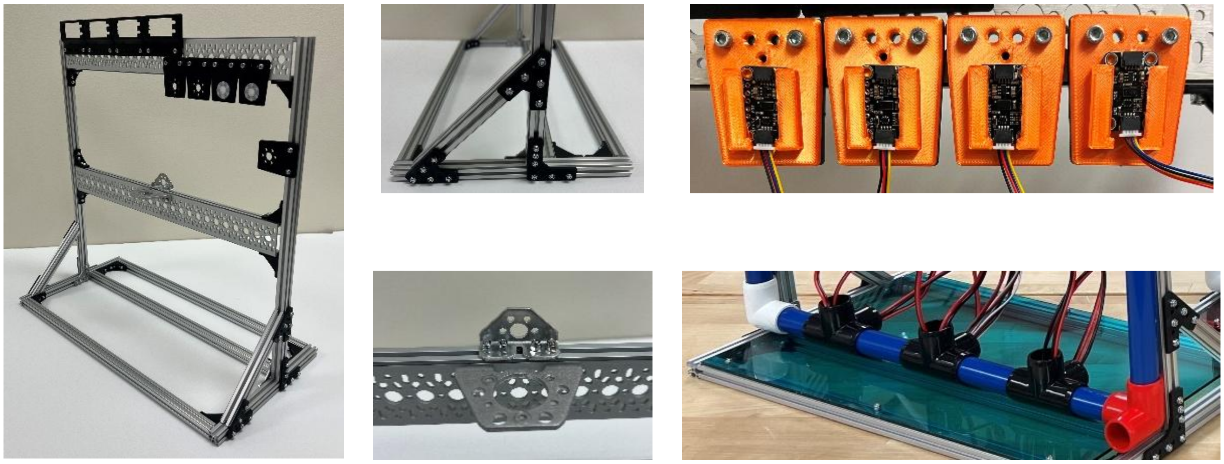


Figure 1: Testbench frame, custom sensor holders, and PVC fittings

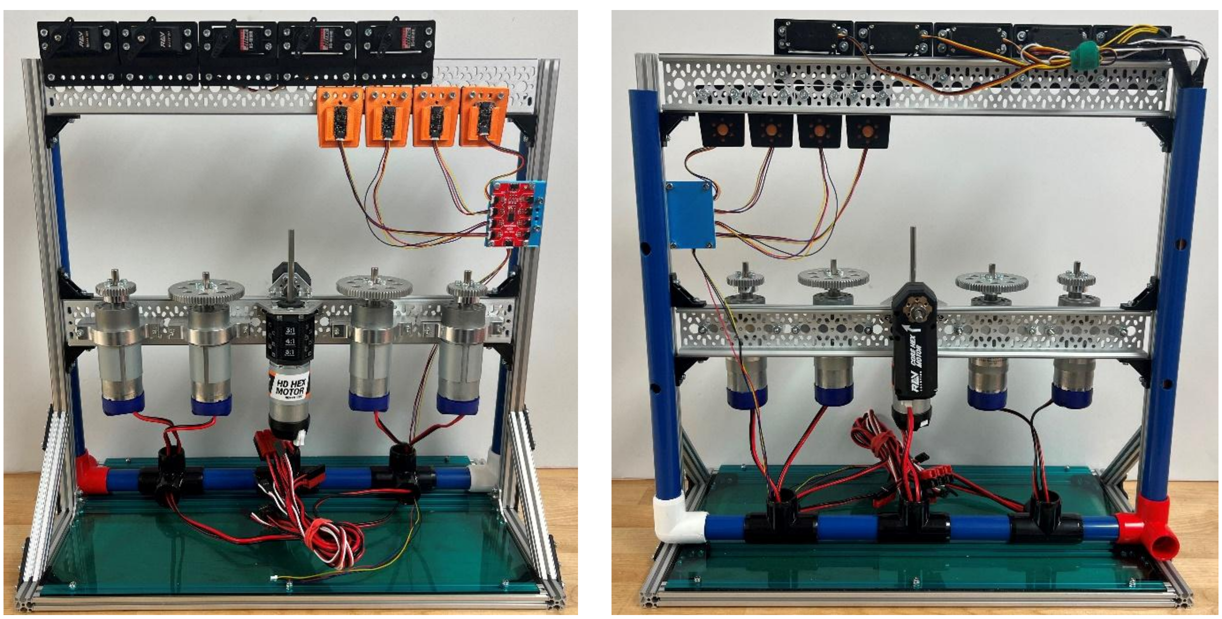


Figure 2: Front and back view of the complete testbench

Table 1: Bill of material and cost for the testbench

Item Name	Part Number	Cost	Quantity
Motors and Servos			
TETRIX MAX TorqueNADO Motor	44260	30.75	4
TETRIX Continuous Rotation Servo Motor	40379	19.75	3
UltraPlanetary Gearbox Kit & HD Hex Motor	REV-41-1600	45.00	1
Core Hex Motor	REV-41-1300	28.5	1
Smart Robot Servo	REV-41-1097	30.00	2
Structural Elements			
45mm x 15mm C Channel - 440mm	REV-41-1761	16.00	2
15mm Extrusion - 420mm - 90° Ends	REV-41-1432	6.25	5
15mm Extrusion - 225mm - 90° Ends	REV-41-1431	4.25	2
15mm Extrusion - 150mm - 45° Ends	REV-41-1430	3.50	2
Electronic Elements			
Adafruit Time of Flight Distance Sensor	5396	14.95	3
Adafruit 10-Channel Light / Color Sensor	4698	15.95	1
SparkFun Qwiic MUX Breakout	BOB-16784	12.95	1

3. Raspberry Pi-based Robot Controller (PRC) Design

Python is a widely used, high-level programming language renowned for its simplicity and readability. Its clean and intuitive syntax makes it beginner-friendly, enabling developers to focus on problem-solving rather than complex code. Python’s versatility spans numerous fields, including machine learning, automation, and robotics. With a robust ecosystem of libraries and frameworks, it accelerates development and enhances productivity. This blend of ease of use, power, and adaptability has made Python one of the most popular programming languages in the world. As a result, there is a growing demand in the industry for engineers proficient in Python programming skills. Python is considered the primary programming language used with the Raspberry Pi microcontroller.

Raspberry Pi is a versatile, affordable, and powerful single-board computer ideal for various projects. Its low cost makes it accessible to beginners, hobbyists, and educators, while its small form factor and low power consumption offer convenience for embedded systems. Raspberry Pi supports various operating systems, including Linux, and features GPIO pins for hardware interfacing. It’s used in educational settings to teach programming and electronics, in DIY projects for home automation, robotics, and media centers, and even in professional applications for prototyping. The Raspberry Pi community provides extensive resources, fostering innovation and collaboration.

Due to the importance of programming and hardware-software integration in mechatronics engineering education, the authors used Python and Raspberry Pi to develop a training controller for robotics education. The Raspberry Pi-based Robot Controller (PRC) includes additional electronic components that enable the students to control up to 6 DC motors at a time, while still having enough room to control 4 servo motors. Using less DC motors allows room to control more servo motors if desired. In addition to controlling individual motors, the PRC can simultaneously read measurements from 8 i2c sensors. While the students are not required to build the controller, they are introduced to the PRC components in an aesthetic assembly. Figure 3 shows the latest developed prototype of the PRC.

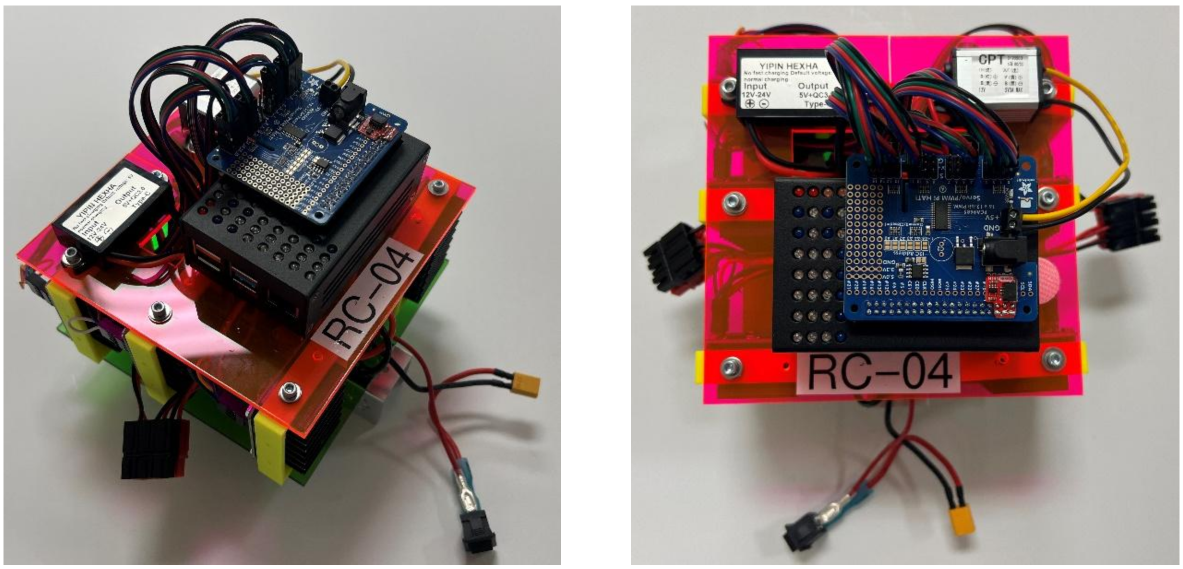


Figure 3: Views of the developed controller

The PRC consists of a Raspberry Pi, a 16-Channel PWM HAT that can drive up to 16 servos or 8 DC motors, 6 motor driver modules, 2 12V to 5V DC converters, terminal block distribution module, switch, and 12V battery. The cost of one PRC is about $300. Figures 3 and 4 show the PRC schematic and electric wiring diagram, respectively. Table 2 lists the essential components of the PRC.

Table 2: Bill of material and cost for the controller

Item Name	Part Number	Cost	Quantity
Electronic Elements			
CanaKit Raspberry Pi 4	Model B	74.99	1
Adafruit 16-Channel PWM HAT	2327	17.5	1
HiLetgo 43A High Power Motor Driver Module	BTS7960	10.99	6
Dorhea 12V to 5V DC Converter	C120503	8.99	1
YIPIN HEXHA 12V to 5V USB-C Converter	QC3.0	8.99	1
OONO Terminal Block Distribution Module	D1417	15.99	1
Battery and Charger			
12V Slim Battery	REV-31-1302	55	1
Battery Charger	REV-31-1299	36.5	1

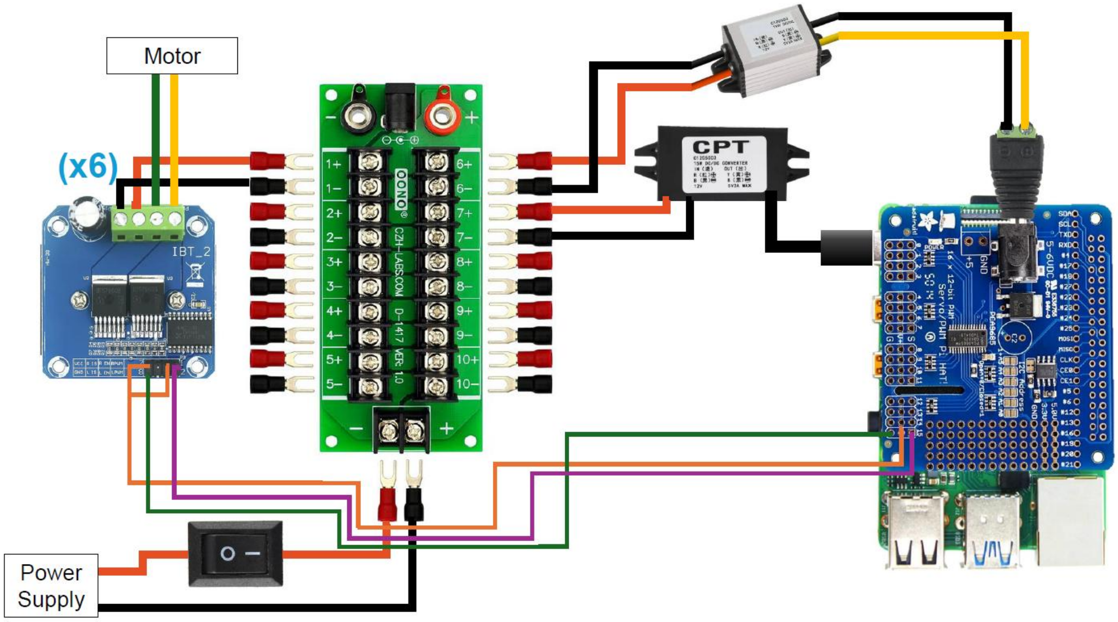


Figure 4: Schematic of the robot controller

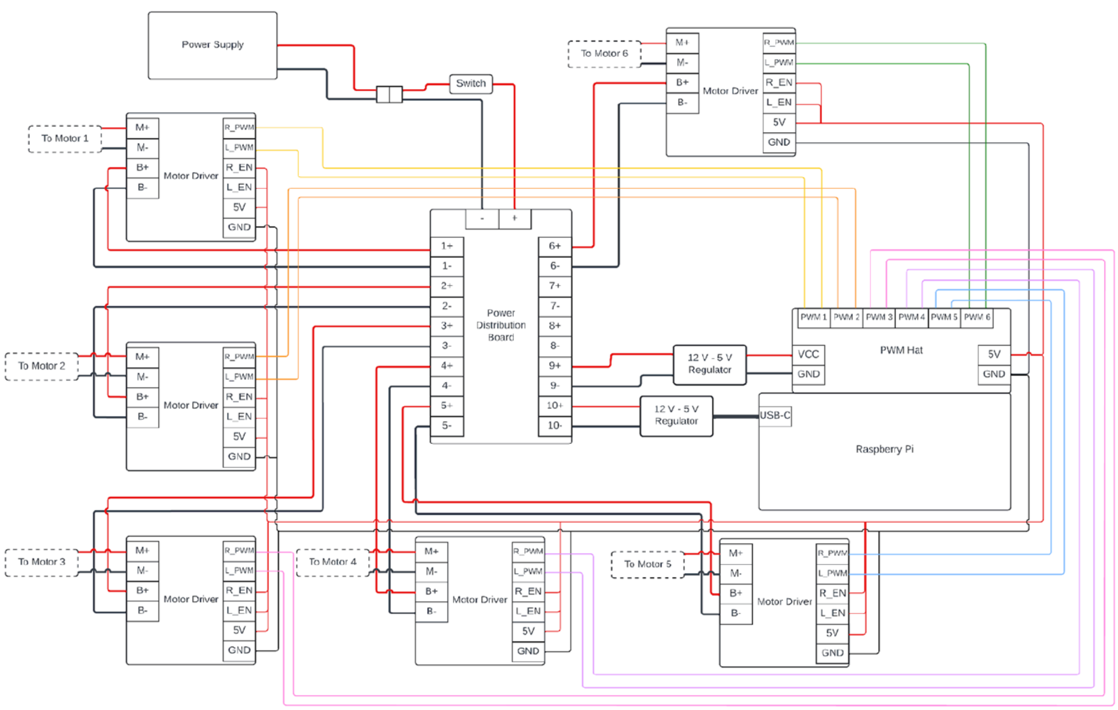


Figure 5: Electric diagrams of the robot controller

4. Laboratory Activities

The testbench and controller are used in the 6th lab session (Robotics Actuators) out of a total of 9 sessions before the final project, which is conducted in 5 sessions. In this section, a brief summary of the developed lab manual chapter is presented.

Lab Introduction: This lab teaches you how to use PWM (Pulse Width Modulation) to control the speed of DC motors, control servo motor angles, and measure sensors signals.

Objectives**:** Develop a code to operate 6 DC motors, 4 servo motors, and 8 sensors simultaneously using PRC.

Software Used**:** VSCode

Hardware Used**:** PRC (Raspberry Pi)

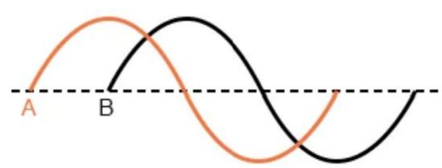
Lab Activity 1: I2C Device Connection

---

1. Create a new folder in the Documents directory called "Team#XX" (use your team's number.
2. Create a new folder in your team's folder called "Lab06".
3. Create a new file in this folder called "motorServo.py".
4. Copy and paste the code from the .txt file on Canvas into your Python file.
5. Navigate to the terminal. Type in ***i2cdetect -y 1***. You should see that there is an i2c device at address 40 or 43. These addresses are common for the PWM hat (remember the common address for the ToF sensors is 29?).
6. Modify the ***_PCA9685.__init__()*** function to have the correct i2c address that popped up for the pwm hat (*0x40* or *0x43* most likely).
7. Run the program to ensure that no errors pop up. Two motors and two servos should power on for a few seconds.
8. Create your own summary of this activity and include the i2c address that you used for the PWM hat in the lab report.

Lab Activity 2: Servo Motors

The task for this activity is to program two servos to power on based on the below sinusoidal looking inputs. Read through the example code to learn how to initialize, power on, and power off the servos. Be creative in how you make this code! There is much more than just one way to accomplish this task.



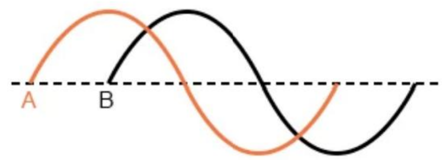
1. Create a new file called "Activity2.py"
2. Import the **motorServo.py** file by using *import motorServo as XXX*. You can decide what makes sense to you to use as your calling variable.
3. You can also import the time, math, or numpy libraries if you decide that they may be useful.
4. Servos can spin from 0 to 180 degrees, so your program should make them start at 90 degrees. That way, the "positive" wave input can make it go up to 180 degrees, and the "negative" wave input can make it go to 0 degrees.
5. The two servos should be 90 degrees out of phase from each other.
6. One period should be about 4 seconds.
7. Create an infinite loop for the servos until the keyboard interrupt is used "Ctrl + C".

---

8. Create your own summary of this activity and include the .txt file of your code and a video of the servos with the lab report.

Lab Activity # 3: DC Motors

Do the same thing as Activity #2 but using the DC motors. Read through the example code to learn how to initialize, power on, and power off the motors. Be creative in how you make this code! There is much more than just one way to accomplish this task. The maximum duty cycle for each motor should be 50%.



1. Create a new file called "**Activity3.py**".
2. Import the **motorServo.py** file by using *import* ***motorServo*** *as XXX*. You can decide what makes sense to you to use as your calling variable.
3. You can also import the time, math, or numpy libraries if you decide that they may be useful.
4. The maximum duty cycle for each motor, or the peak amplitude of your sinusoidal shaped input signal, should be 50%. These motors spin very fast so don't go above 50% for this activity.
5. DC motors use a percent duty cycle to determine their speeds, so 0% means not moving, and 100% means moving at maximum speed. A negative pwm does not mean moving in reverse; One PMW channel is for forward and the other is for reverse, so to make the motor spin in reverse, the forward channel needs to be set to 0% and the reverse channel is set to a positive number.
6. One channel must be set to zero at all times!
7. One period should be about 4 seconds.
8. Create an infinite loop for the servos until the keyboard interrupt is used "Ctrl + C". Remember to include code to stop the motors.
9. Create your own summary of this activity and include the .txt file of your code and a video of the motors with the lab report.

Lab Activity 4: ToF Sensor Integration

The goal of this task is to power your DC motors and servo motors based on the distance reading from your ToF sensor.

1. Restrict your sensor reading range between 0 and 100 cm. If a reading above 100 cm is detected, end the program and stop all motors and servos.
2. At 50 cm:
    a. Motor 1 is at 0% duty cycle
    b. Motor 2 is at 0% duty cycle
    c. Servo 1 is at 90 degrees
    d. Servo 2 is at 90 degrees
3. As the sensor readings get smaller than 50 cm:
    a. Motor 1 speeds up going CW
    b. Motor 2 speeds up going CCW
    c. At 0 cm, both motors are spinning at their maximum duty cycle for this activity (50%)
    d. Servo 1 decreases its angle
    e. Servo 2 increases its angle
    f. At 0 cm, Servo 1 is at 0 degrees and Servo 2 is at 180 degrees
4. As the sensor readings get larger than 50 cm, do the same as above but in opposite directions.

5. It’s not necessary to initialize the ToF sensor as a class like in Lab 04. You only have to use what you need from that code.
6. Create your own summary of this activity and include the .txt file of your code and a video of the motors with the lab report.

5. Examples of Students’ Outcomes

The students were assigned to work in teams to complete the tasks outlined in the lab manual within a 3-hour session. Introductory lectures and labs were given in Python and Raspberry Pi. In the Robotics Actuators Lab session, the testbench and PRC are used together to control the motors and collect measurements from the sensors.

As the students progressed through the tasks, the coding complexity increased, providing a greater challenge. The students were required to optimize the algorithm and clarify coding lines. In addition to applying their technical knowledge, students needed to incorporate critical thinking and effective teamwork to complete the final tasks successfully. The results varied across teams, showcasing unique approaches and solutions. Figure 6 illustrates students actively engaging in task completion during the lab session.

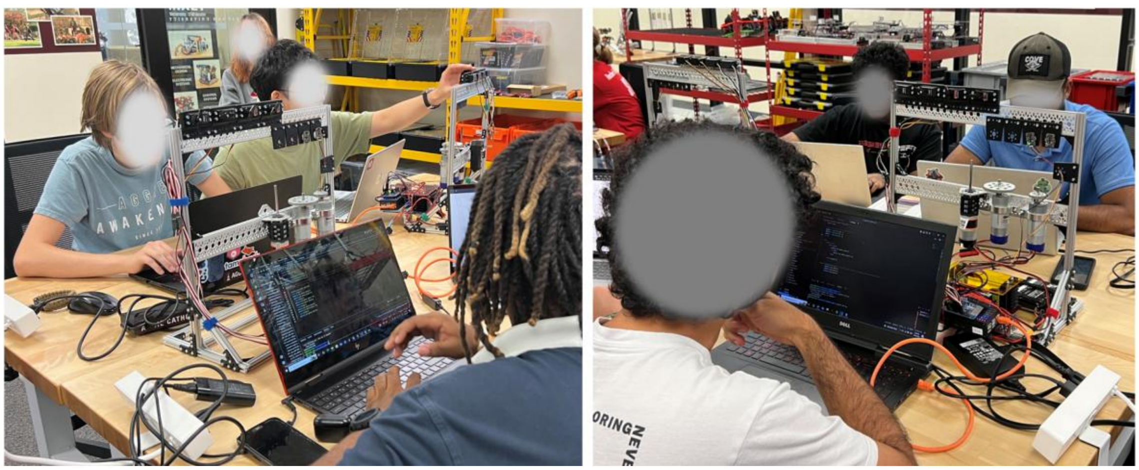


Figure 6: Students using the testbench and controller

In the final project lab sessions, the students receive a box of robotic structural elements, a set of motors, servos, and sensors similar to those in the testbench, and one PRC. The testbench stays on each team’s table so that the team members can work simultaneously on building and programming the final project robot. Figure 7 shows some of the developed robots in the final project.

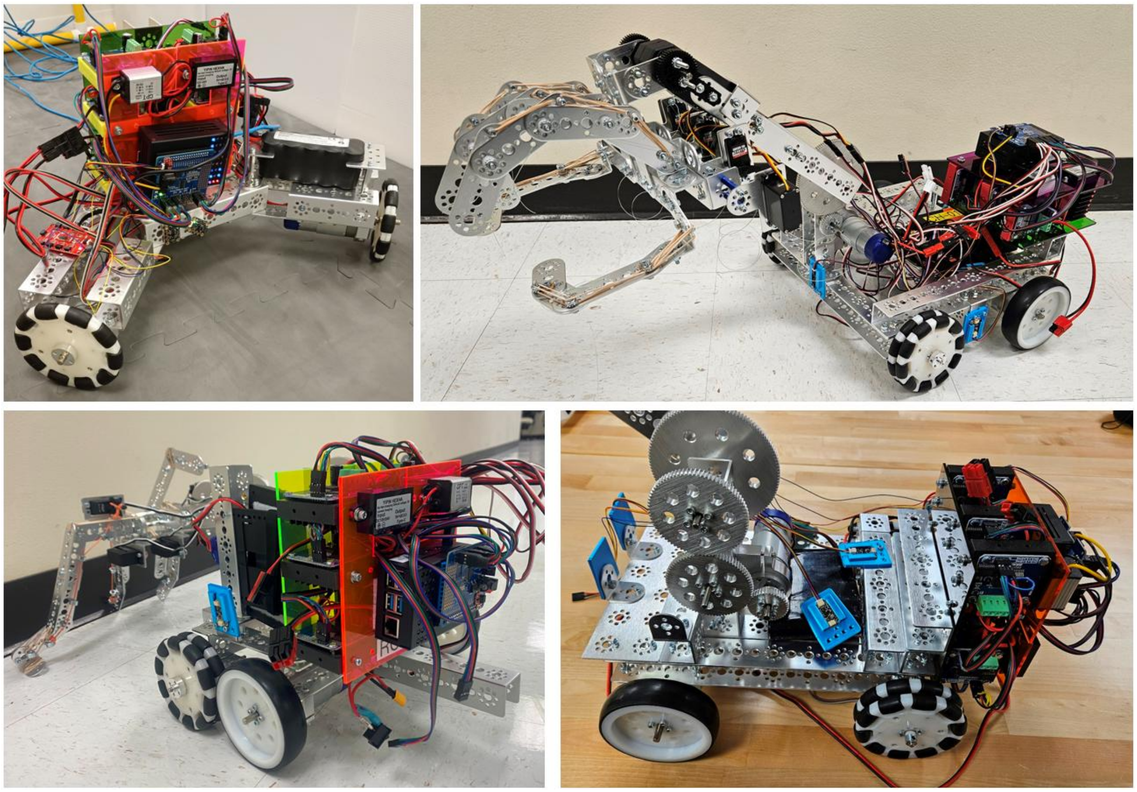


Figure 7: Final projects with the robot controller

The benefits for students from using the developed MTP testbench and associated lab chapter include:

- Proficiency in Python Programming: Students develop a strong foundation in Python, learning core programming concepts such as variables, loops, conditionals, functions, and object-oriented programming, with practical applications in robotics.
- Hands-On Experience with Robotics Components: Students gain practical experience in controlling various robotics components such as DC motors, servo motors, sensors (e.g., distance, color), and actuators, reinforcing theoretical knowledge with real-world applications.
- Understanding of Robotics Systems: By working with Raspberry Pi and robotic hardware, students develop a deeper understanding of how different robotics systems are designed, assembled, and controlled, including the integration of hardware and software.
- Development of Problem-Solving Skills: Students learn to approach complex robotics challenges, troubleshoot issues, and apply programming logic to solve real-world problems in robotics, fostering critical thinking and creativity.

- Prototyping and Experimentation: Using the Raspberry Pi as a platform, students learn how to prototype and iterate robotic mechanisms, testing and refining their designs while collecting data for analysis.
- Exposure to Industry-Relevant Tools: Students gain familiarity with widely used tools in the field of robotics, such as the Raspberry Pi, Python libraries, and sensor integration, making them more industry-ready.
- Collaboration and Teamwork: Robotics projects often require teamwork, helping students develop interpersonal skills, collaboration, and the ability to work effectively in a group setting to achieve project goals.
- Data Collection and Analysis: Students learn how to collect sensor data, analyze it, and use the results to inform decisions and improve system performance, enhancing their data analysis and interpretation skills.
- Enhanced Computational Thinking: Through hands-on programming and robotics projects, students refine their computational thinking abilities, learning to break down complex problems into smaller, manageable tasks and implement efficient solutions.
- Foundation for Advanced Robotics Concepts: Students are introduced to advanced robotics concepts such as autonomous control, real-time processing, and machine learning, providing a solid foundation for more advanced studies and projects in robotics and mechatronics.

These benefits prepare students for careers in robotics, automation, and mechatronics by equipping them with both the theoretical knowledge and practical skills needed to succeed in the field.

6. Conclusion

This paper introduces a novel, low-cost testbench and controller designed to teach Python programming with applications in robotics for mechatronics education. The testbench and accompanying experiments allow students to grasp Python fundamentals while interacting with a variety of actuators and sensors. Designed, built, and tested for a hands-on robotics course aimed at sophomore engineering students, the testbench supports extensive prototyping of robotic mechanisms. During the Python programming laboratory sessions, students learn how to control various DC motors and servo motors. They then use the testbench to collect and export data from distance and color sensors. The testbench consists of a frame, a range of motors and sensors, and a versatile controller based on a Raspberry Pi. This controller was developed to manage six DC motors, four servo motors, and eight sensors. Integrating this testbench into mechatronics engineering courses will significantly enhance student engagement and foster a deeper understanding of Python programming. Additionally, the testbench is low-cost, portable, and easy to replicate, making it an ideal tool for educational settings.

Acknowledgment

The authors would like to acknowledge the support of Dr. Reza Langari (Professor of Mechanical Engineering at Texas A&M University). Dr. Langari supported the development of the MXET 250 (Robotic Systems Design) course and associated laboratory space.

References

[1] Wikander, J., Torngren, M., & Hanson, M. (2001). The Science and Education of Mechatronics Engineering. IEEE Robotics & Automation Magazine, 8(2), pp. 20-26.

[2] Tantawi, K., Nasab, A., Potter, L., Tantawi, O., Wilson, N., Henrie, A., Sirinterlikci, A. & Kaplanoglu, E. (2023). Mechatronics Education-Current and Future Trends. In IEEE 11th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-5.

[3] Ceccarelli, M. (2003). Robotic Teachers' Assistants. IEEE Robotics & Automation Magazine, 10(3), pp. 37-45.

[4] Çeven, S., & Albayrak, A. (2020). Design and Implementation of Modular Test Equipment for Process Measurements in Mechatronics Education. Computer Applications in Engineering Education, 28(2), pp. 324-337.

[5] Fernandez, G.C., Borrego, R.C., Merino, P.P., Lopez, M.A.C., Ruiz, E.S., Gil, M.C., & Perez, F.M. (2015). Mechatronics and Robotics as Motivational Tools in Remote Laboratories. In 2015 IEEE Global Engineering Education Conference (EDUCON), pp. 118-123.

[6] Vasquez, H., & Fuentes, A.A. (2013). Integration of Sensors and Low-Cost Microcontrollers into the Undergraduate Mechanical Engineering Design Sequence. In 2013 ASEE Annual Conference & Exposition, pp. 23-789.

[7] Cardeira, C., & Da Costa, J.S. (2005). A Low Cost Mobile Robot for Engineering Education. In 31st Annual Conference of IEEE Industrial Electronics Society (IECON), pp. 2162

[8] Gonçalves, J., Silva, M.F., Costa, P.G.D., & Sousa, A. (2015). Proposal of a Low Cost Educational Mobile Robot Experiment: An Approach Based on Hardware and Simulation. In 6th Internation Conference on Robotics on Education, pp. 14375.

[9] Lu, Z., & Pangborn, H., & Fitzsimons, K. (2023). Developing an Affordable Multi-Course Electronics Kit for Increased Software and Mechatronic Literacy in Mechanical Engineering. In 2023 ASEE Zone 1 Conference, State College, Pennsylvania, pp. 45089

[10] Bianchi, G., & Cazzulani, G. (2024). Teaching Students to Control Mechanical Systems Through an Engaging Laboratory Activity. International Journal of Mechanical Engineering Education, OnlineFirst.