3D printed Robotic Arm Project Report

Kinematics and Control of Robotic Systems (EEL4664.01)

Elliott Hanke and Kaitlyn Surovy

Introduction

In this project, we built the mechanical aspect of a robotic arm using 3D-printed parts and wired the electrical components. We also coded the robot to run autonomously using Arduino. The main objective was to create a robot capable of picking up a small plastic duck and being able to put it back down autonomously. Using the instructions provided by HowToMechatronics and applying our skills gained throughout this course, we were able to successfully build, wire, code, and fully control the robot.

Materials (Bill of Materials)

For this project, we used the provided 3D-printed robot arm parts and our own Arduino kit, which contains the servos, micro servos, wires, breadboard, and a USB cable. We also needed extra parts that were bought including M3 bolts, a wooden platform for the robot, screws, and command strips to ensure the robot arm was not only stable on the platform, but also can be easily moved around. We used our own 5V power supply to provide stable power without overloading the servos. Finally, we used the Arduino IDE using C++ for controlling the arm.

Methods

To begin, we were given the 3D printed robot parts including the base, links, joints, etc. With these parts, we were able to begin the construction of the robot arm. We started with the base and mounted the first servo motor and the servo horn inside the round base for the rotation of the base of the arm. We then finished the housing of the base by adding the cover and securing it with 2 screws. After this, we moved onto the shoulder of the arm, where we attached it to the waist (or base) we previously built, securing the shoulder with a screw. We attached a rubber band around part of the base and the shoulder to support the excess weight on the shoulder and prevent the servo from overheating. The next part was mounting the elbow servo to the elbow joint and securing it, which then connects back to the shoulder. We repeated this process with the wrist of the arm but used a micro servo instead. By using the parts of the gripper and the M3 bolts, we were able to assemble the gripper and mount the micro servo. We tested each joint to ensure the functionality and make sure the servos worked properly.

For the electronics, we started by connecting each control pin of the servo motors to a specific pin on the Arduino board, so that we can control what the servo does. The servos get their power from a separate external 5V source because it will overload the board if not as the Arduino alone did not provide enough power.

We were able to use the emulator code provided as a base for the robot arm. Using this, we were able to use the pins to control the servo positions. We used a for loop to iterate through an array of strings, where each string represented a specific servo and position for the robot to move through in the pick-and-place process. The board receives the servo ID and the angle and gradually moves the servo to the correct position. We also used a wait command for 1000ms, so that the robot was controlled and never overextended itself. Using the emulator, we were able to control the robot using manual control and a stored sequence, which is how we obtained autonomous movement.

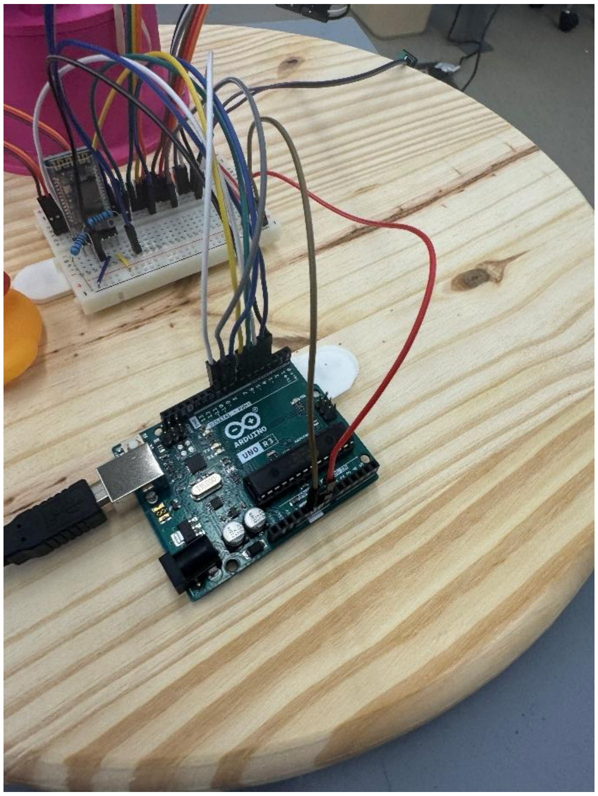

Figure 2 - Completed breadboard and Arduino board with the servos attached and working correctly with the external 5V power source connected

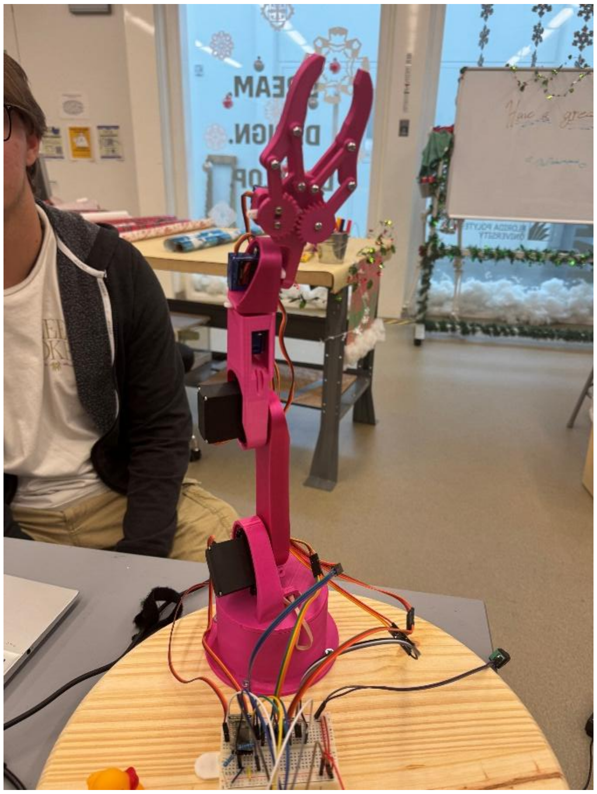

Figure 1 - Fully built and constructed robot arm at the starting position (straight up), which shows the different joints (waist, shoulder, elbow, wrist, gripper)

Results and Discussion

Once the robot arm was fully assembled, the robot arm was fully capable of performing the pick-and-place task with the small plastic duck using both autonomous and manual control with the emulator. However, we encountered multiple challenges throughout the project, including the servos burning out, the Bluetooth control not working correctly, and a slight jerky movement of the arm. One of the micro servo motors burnt out on us during the final testing of the arm due to the excessive operation. We switched out the micro servo with another to ensure it functioned correctly and put more operation onto the other joints to work better together and not burn out. The attempts we made at having Bluetooth functionality were not effective. We only had one Samsung device to use, but the program given seemed to be incompatible with the device. We tried updating the software to the current version, but they did not seem to be compatible. Instead, we switched to an emulator, which was able to send commands straight to the board. Using the emulator, we were able to successfully control the robot and ensure correct functionality. We experienced a bit of jerky motion throughout the project. By using smaller increments in the movement of the joints, it was able to run smoother and in a more controlled way. Although we did face many challenges, we were able to successfully complete this project, and the robot was able to meet the requirements set. By addressing the issues that went wrong, we were able to debug the arm and operate the arm correctly, while gaining knowledge and experience in managing problems in areas of power management and servo control.

Conclusion

Against many challenges, we produced a fully working Arduino-controlled 3D-printed robotic arm capable of basic automated processes. This project was able to provide our group with invaluable hands-on experience with the servo motor control, wiring and power considerations, and microcontroller programming. In the future, we would like even smoother movement between positions and the successful integration of the Bluetooth control for smartphone control.

GitHub Repo

https://github.com/ksurovy/P-ARM-Robotic-Arm-Project.git

Acknowledgement

We would like to thank Professor Ngo for the knowledge we learned throughout the semester to help with this project and supplying the hardware components. We also would like to acknowledge the Florida Poly Makerspace for allowing a space to work and supplies to use. We would like to thank the HowToMechatronics.com website for providing instructions and the STL file to aid in this project.

References

[1]

Dejan, "DIY Arduino Robot Arm with Smartphone Control," *HowToMechatronics*, Sep. 11, 2018. https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/

[2]

P. Corke, "Robotics Toolbox for MATLAB® release 10," *GitHub*, Nov. 08, 2023. https://github.com/petercorke/robotics-toolbox-matlab