Robert Murrer
Project 2 Report
EML6805
November 8, 2022

AR4 Robotic Arm Gripper & Testing

Description

This report will detail the finishing of the Annin Robotics AR4 robot arm. In the first installment the arm was completely assembled as well as the enclosure being fully wired up. The robot was then functional in movement only and was able to calibrate itself. The second portion of this project was the attachment of a gripper, finishing the enclosure, and to perform some tests.

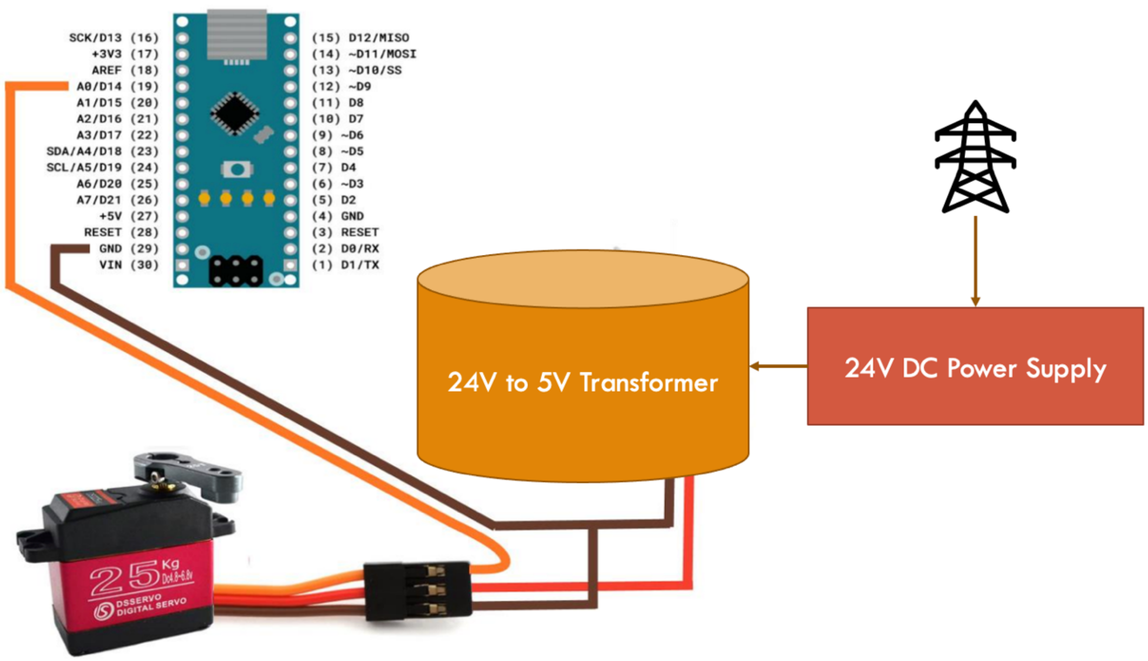
Gripper System

The original design of the AR4 arm is tool agnostic. The Teensy microcontroller that is included in original build is only used for kinematics. To add a gripper the Arduino Nano compatible microcontroller is used to control the servo. The 3D printed components are made of the same Onyx material used for the rest of the robot build. Although they are printed with higher density for improved strength.

Bill of Materials

- Arduino Nano Compatible Microcontroller
- 3D Printed Components from [1]
- 25Kg Servo Gripper DS3225
- Various Lengths of 3mm Steel Rods
- #6 Self Tapping Panhead Screws
- M3x8 and M3x10 with Locking nuts
- 24v to 5v Transformer

System Diagram

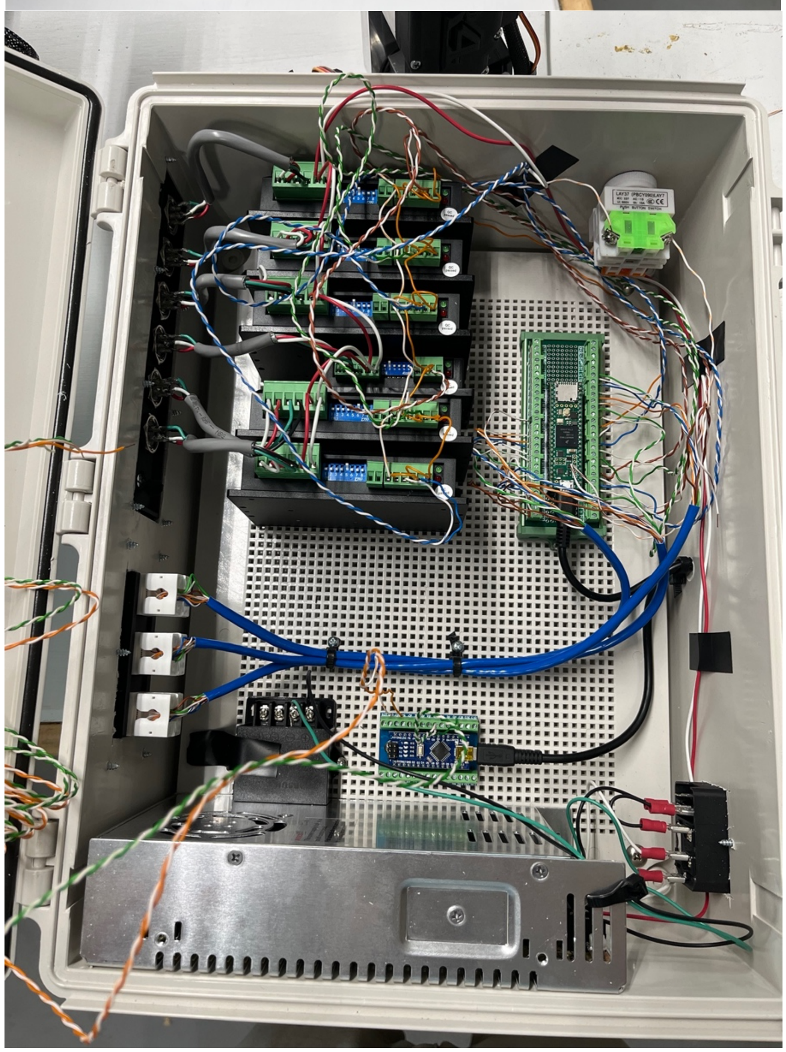


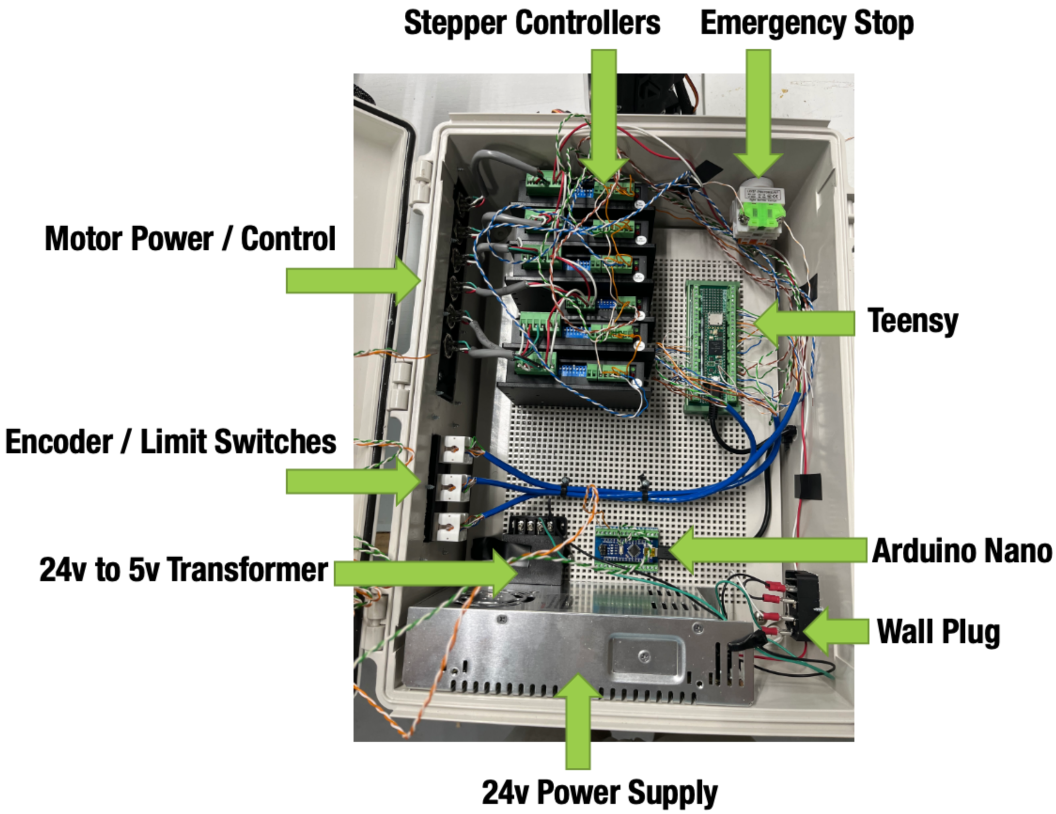
Adapted from [1]

Finalizing the Enclosure & Base

After electrical testing and debugging any connectivity issues that occurred on initial assembly, all the motor stepper controllers needed to be mounted to the grid. The enclosure also needed to have some ventilation so when fully closed the components do not overheat. Finally, the robot base was mounted to 31"x31" piece of ¾" plywood so that the robot can be displayed easily on tables without mounting screws.

There were also some parts that needed to be reprinted or attached to assist in enclosing joint 5 belt system and assist with cable management. All these parts were printed at Sea3D lab.





Software & Code

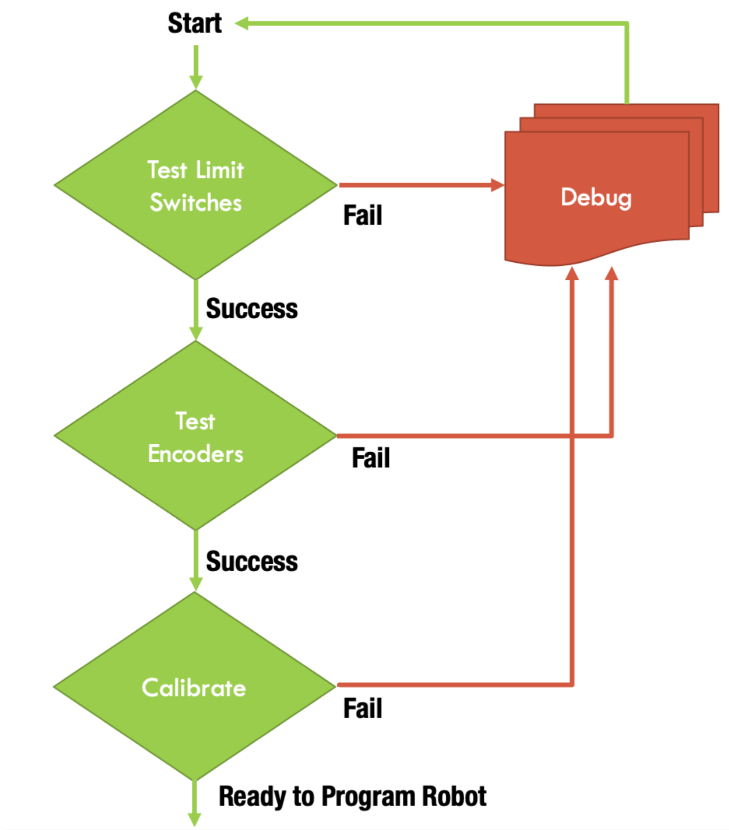
There are 3 main pieces of software to run the arm. The Teensy firmware which was presented in the past paper will be omitted from this document. The Arduino Nano firmware can be found in the appendix. [3]

The AR4 Human Machine Interface (HMI) which is the main control software for the robot will be discussed in more detail in the following sections.

First Time Setup

There are several test programs included with the AR4 HMI software.

- "test limit switches" – Allows you to test each limit switch manually to determine if they are wired properly and the limit switches in fact work. This is important as if the limit switches do not work the robot cannot calibrate itself and can damage itself.
- "test encoders" – This program when run the power should be turned off from the stepper motors. Then each joint is articulated manually, and the software display confirms the orientation and that the encoders are functioning properly.

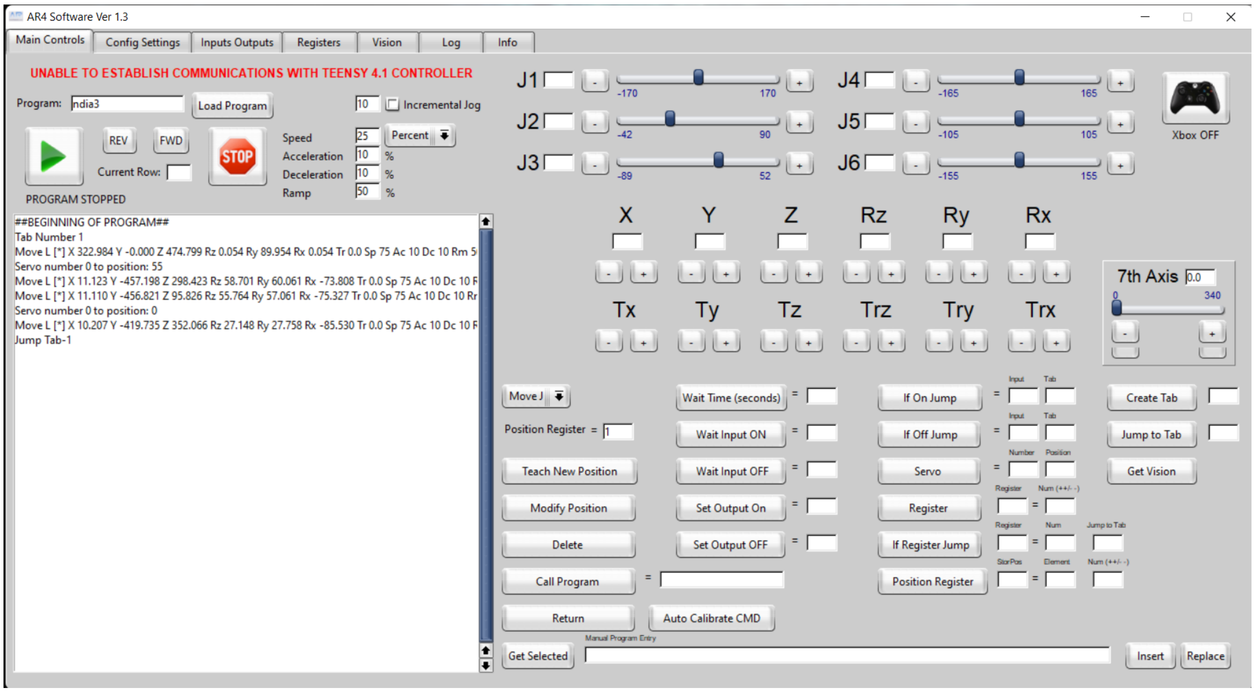


Calibration

After confirming that both the limit switches and encoders are functioning properly, the robot can run calibration. The calibration program jogs each joint until it hits a limit switch and then rotates each joint into its rest position. On inspection of this rest position each joint should be at 90-degree angles. A digital level can be used to measure and ensure this is the case. If there is a discrepancy each joint can be offset in the configuration tab of the HMI.

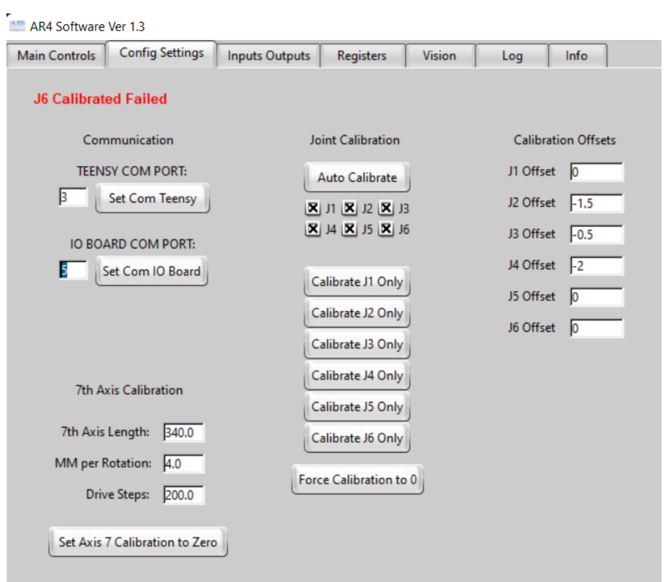
AR4 Human Machine Interface (HMI)

This is the main control software of the AR4 robotic arm. All calibration, testing, and programming is done through this tool. The version used is 1.3. Version 1.4 was released shortly before the finalization of this robot, but it was ignored to keep errors minimized.



Setting Up

Since there are now two microcontrollers hooked up to the computer through USB it is important to identify each of the COM port used for serial communication and assign it in the GUI. You can see IO BOARD below is the Arduino Nano. Also note the calibration offsets.



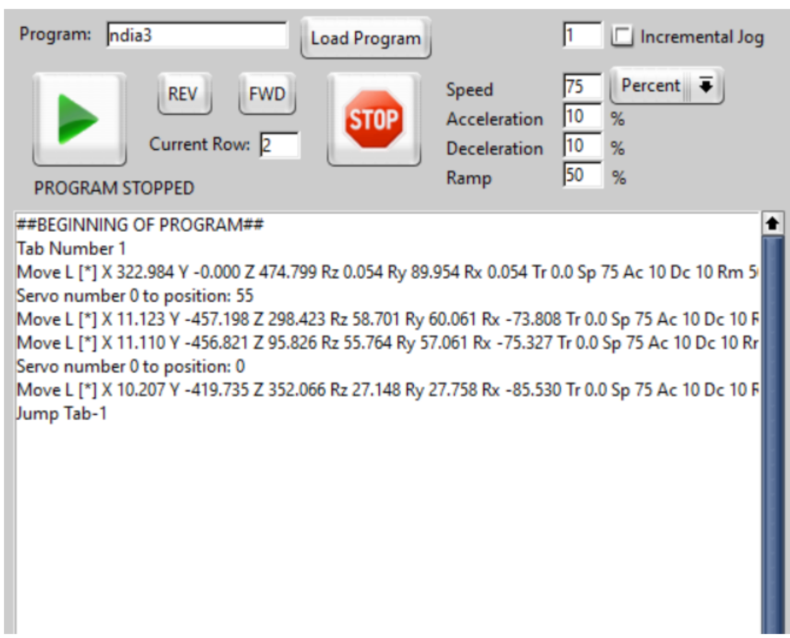
Programming the Robot

To program the robot, it is quite intuitive. After the robot has been calibrated and is at rest position, you may start teaching positions and programming the robot. There are a few move commands, but I will only focus on "Move L" for this portion of the project. This is a linear move from one position to the next.

The procedure for programming the robot is as follows.

1. Create Tab 1 which is the starting point of the robot
2. Teach the rest position of the robot by selecting "Move L" and Teach New Position
3. Move the robot using the Joint controls or Cartesian controls
4. Teach the position to the robot through "Move L"
5. When teaching robot to pick something up with gripper it is important to set way point above object rather than directly at pickup point. This will ensure the trajectory doesn't hit the object it is moving to the place for picking up.
6. If desired jump to Tab 1 to repeat program.

Sample Program for Pick and Place



Note that the gripper is connected to Servo 0 and a zero value is its closed position and 55 is its open position. So, in the above program the robot will move almost directly to its right and pickup and object, move back to its rest position and open servo. Then it repeats.

Results & Analysis

With the enclosure vented, all the components inside secured, and the robot mounted to a base it was time to move the robot to a public forum for demonstration. The robot was moved to the Destin Convention Center on October 31st and was on display November 1st performing a pick and place program dropping key chains into a container.

Repeatability Tests

Originally, I planned to do a more formal analysis of the repeatability, but in the end, I settled for running the pick and place program in a loop for few hours to see if there was any drift in placement. There was no apparent drift.

I will be analyzing the repeatability further when I can 3D Print a penholder and draw lines and measure the difference with a microscope in future semesters.

Load Testing

The robot creators claim is that it can carry a 4lb payload. Unfortunately, the gripper was not able to hold much weight at all. Originally this robot is designed with a pneumatic gripper. In an impromptu load test, we could move in total 3 carabiners linked together before dropping.

Orientation of the gripper could have been changed to have the load perpendicular to the gripper jaw, but on trying to reprogram the robot, one of the limit switches bent…

Challenges & Troubleshooting Notes

As with most robots, issues arise from repeated use. After running the robot successfully for about 4 hours some weaknesses started to appear.

- Joint 5 Limit failed on a subsequent calibration causing the switch to become bent. I'm not sure if this was a mechanical or electrical failure.
- Joint 4 servo motor became very hot. This is the motor that is visible attached to a belt in the above picture. I am not sure if this is normal and will investigate further before operating robot for extended periods of time.
- Following up with Joint 4, there is some "slop" in the triggering of the limit switch do to what I think the part not being fully pressed on. Again, this will be investigated. I have had to replace this limit switch 2 times now because the aluminum chassis is bending it on some calibration runs.
- When connecting all the servo controllers to the grid, it was quite easy to dislodge the control wires. Looking for red lights on the controllers isn't always enough to catch the floating wires. If a motor error occurs in the software but all the encoders and cables work. Double check the connection to controller, chances are power cable isn't attached fully.
- On hookup of the 24v to 5v transformer for the servo behaved normally at first but after a while there was significant jitter in the servo. I'm not sure if I need a digital ground or some other configuration. In the end the USB 5v Source from the Nano was enough to power the servo.
- The joint stop pictured below for Joint 2 failed on a calibration and caused the limit switch to break. This part was reprinted in a different orientation and with higher density to ensure mechanical stoppage.

You can never have too many limit switches!

Appendix

References

1) AR4 Gripper Manual - https://drive.google.com/file/d/1EmDsH71n0chPxTI5fSjBHcfw5GjuttP7/view?usp=sharing
2) AR4 Manual - https://drive.google.com/file/d/1RTrN32WDTpDZGJapVFAvZOyV8Nq2nmqc/view?usp=sharing
3) Arduino Nano Firmware for Gripper - https://drive.google.com/file/d/1NWg7z-nHZPYCWo470vDmVOci5omuaV_Z/view?usp=sharing

Acknowledgements

- Thank you to Murilo Basso at Sea3D lab for printing the parts for the gripper and reprinting parts that failed
- Thank you to Dr. Sevil for providing funding to attend the NDIA symposium where the robot was part of the HAAS Center's booth.