Memo

To: Professor Pisano

From: Linden Adamson, Jason Calalang, Tara Gill, Zhilang Gui, Cole Resurreccion, Raina Yin

Team: Team 7 - Bike for Blind

Date: 02/22/2025

Subject: Second Prototype Testing Plan

**Boston University**

**Electrical & Computer Engineering**

**EC464 Senior Design Project**

**Second Prototype Testing Plan Report**

**Blind Bike**

****

**By**

**Team 7**

**Bike for Blind**

**Team Members**

**Linden Adamson** [lindena@bu.edu](mailto:lindena@bu.edu)

**Jason Calalang** [jasonc16@bu.edu](mailto:jasonc16@bu.edu)

**Tara Gill** [taragill@bu.edu](mailto:taragill@bu.edu)

**Zhilang Gui** [zgui@bu.edu](mailto:zgui@bu.edu)

**Cole Resurreccion** [coler@bu.edu](mailto:coler@bu.edu)

**Raina Yin** [ryin2@bu.edu](mailto:ryin2@bu.edu)

**Required Materials/Equipment**

Hardware:

* Jetson Orin Nano Super Developer Kit
* OAK-D RGBD Camera
* LED Strips - Brake Lights (5V, 8W)
* 2x DS5160 60 kg\*cm Servo Motor
* PCA9685 Motor Driver
* NFP-36GP-555-EN 24V Geared Motor
* DC Motor Driver L298 Dual H Bridge Motor
* 19V 4.2A Barrel Jack Power Supply
* Arduino Uno R3
* Arduino Nano
* Keysight E36318 Triple Output Power Supply
  + 6V for braking
  + 24V for steering

Software:

* Arduino Code
  + Steering script - flashed to Arduino Nano
  + Brake and Key Fob script- flashed to Arduino Uno
* Emergency Stop with Stereo Camera Demo
* Manual Stop Button
* Orin Nano ROS2 Nodes
  + Keyboard Operation (for control)
  + Braking Command
  + Steering Command
  + OAK-D ROS2 Nodes
  + Vectornav

**Set-Up**

First, every component must be wired according to the diagram below.

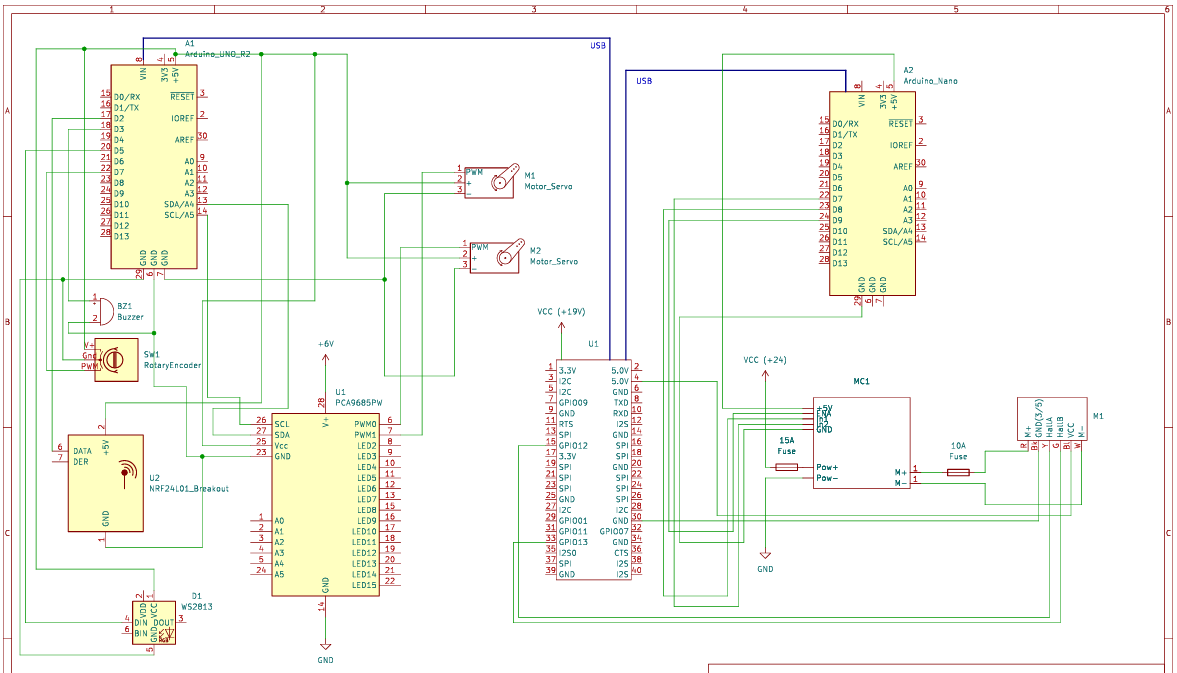


Figure 1: Circuit Schematic of Prototype System

Once all the components are appropriately connected, ensure no exposed wires are making contact with the aluminum frame to cause a short. After confirming connections, provide power to the necessary components.

* Arduino Nano and Uno should be getting power from the USBs provided by the Orin.
* Connect the 6V output from the power supply to the PCA9685 servo driver.
* Connect the higher voltage output from the power supply to the motor controller and set the output to 24V.
* Power the Orin with the 19V barrel jack plug.
* The LED strip should be powered from the 5V pin on the Arduino.
* The OAK-D should be powered from an Orin USB port.

For proper braking tests, at least one of the DS5160 servo motors should be mounted and attached to the trike.

For the braking system test, the Arduino Uno will be connected to the Jetson Orin. The LED strips and PCA9685 motor driver will be connected to the Arduino Uno. The DS5160 servo motors will both be connected to the PCA9685 motor driver, with one servo motor mounted onto the trike’s brake pad.

For the key fob, a 433MHz RF receiver and a 96dB Piezo buzzer will be connected to the Arduino Uno.

For the steering system, a belt loop should be connected to the main steering joint and the motor output shaft. The NFP should be plugged into the controller and the encoder output should be attached to the Arduino.

For object detection, an OAK-D camera will be mounted onto the trike and connected to the Orin.

**Pre-testing Setup Procedure:**

On Orin Nano:

1. Open four terminals on the Orin Nano. Run the following commands on those terminals.
   1. *ros2 run tester key\_op*
   2. *ros2 launch trike main\_launch.py*
   3. *ros2 run tester emergency\_stop*
   4. *rqt*

Power Supply

1. Set the +/-24 V Output of the power supply to +24 V.
2. Set the 6V Output of the power supply to 6V

**Testing Procedure:**

The testing procedure should mimic a person using the trike.

1. Stand 10m away from the trike and use the keyfob by pressing RF button transmitter and hearing the sound from the buzzer to find the trike.
2. On the keyboard operation terminal:
   1. Type the character *s*. The brakes should engage and the LED brake lights should turn on.
   2. Type the character *w*. The brakes should disengage and the LED brake lights should turn off.
   3. Type the character *j*. The steering motor should turn left.
   4. Type the character *k*. The steering motor should stop.
   5. Type the character *l*. The steering motor should turn right.
3. Activate emergency braking software
   1. Clear OAK-D camera’s sightline of any object 2m or closer. The brakes should not be engaged.
   2. Walk to within 2m of the OAK-D camera. The brakes should engage.
   3. Leave 2m sightline of OAK-D camera. The brakes should disengage.
4. Manual brake
   1. Press the manual brake button on the inside of the vehicle. The brakes should engage.
5. Repeat each of the above testing procedures 5 times to prove consistent accuracy of our system.

**Measurable Criteria**

1. Key fob will activate the buzzer at least 10 meters away from the trike with 80% accuracy.
2. Brake lights and brakes engage within 0.1s of receiving brake command from the Orin Nano with 100% accuracy.
3. Brakes will engage when an object is within 2m of OAK-D camera sightline, and disengage when there is no object within 2m of OAK-D camera sightline with 90% accuracy.
4. Manual brake button will engage the brakes with 100% accuracy. The system should confirm full brake engagement within 0.2 second. It should also override the braking command from the emergency brake.
5. Brake servo motors rotate enough to engage the brake pad with 100% accuracy. The response time should not exceed from command issuance to brake engagement must not exceed 0.2 second.
6. The steering motor will rotate in response to the motor command from the Orin, and the rotation angle must match the command input within the tolerance of 2 degrees to ensure accurate steering.
7. The steering motor restricts itself to rotate within a 60-degree range.

**Summary**

Our equipment was successfully set up as described in the test plan. The NVIDIA Jetson Orin Nano was seated outside of the trike plugged into the wall. It was also connected to a monitor with a mouse and keyboard. We opted to use a display instead of SSH in order to visualize the output from our camera. Plugged into the Orin, we had an Arduino Nano used to control steering, an Arduino Uno R3 used to control braking, and an OAK-D stereo depth camera. We noticed if the Nano was plugged in before the Uno, the ROS2 nodes would connect to the incorrect serial ports. Once everything was plugged in, four terminals were successfully opened with the terminal commands above. The ROS2 logs confirmed a successful connection to the camera and the Arduinos. The ROS rqt tool was also used to visualize the camera footage, validating the depth and color images. The steering and braking motors were powered by a Keysight E36318 Triple Output Power Supply with a 24V and 6V connection respectively. The wiring for the rest of the system was also completed successfully. To confirm the wiring was done correctly, we activated steering and braking. As expected, the brakes activated with the brake lights and the steering motor turned within the range. One issue we noticed was the lack of compatibility between the RF signal and LEDs. We found the LED library disables interrupts. To fix this, we moved the RF signal to its own Arduino.

**Measurement Results**

|  | Test: | Result: |
| --- | --- | --- |
| 1. | Key Fob | Test the key fob in and out of the room with the trike. The pass rate is 100%. Tests outside of the room included distances over 10 meters and did not affect accuracy in this range. |
| 4. | Manual Brakes | Manual brake button successfully activated the brakes within 0.2 seconds with 100% accuracy. It was also tested to have critical priority (Always overrides to meet user needs) |
| 5. | Brake Pad Engagement | When physically pushing the trike at a constant but low velocity (~0.3 m/s), the brakes engaged from the servo caused the trike to come to a full stop within 2 seconds. |
| 3. | Camera Brake Control | Brakes automatically (< .2 seconds) engage when an object placed within 2 (configurable) meters of OAK-D camera sightline with 100% accuracy. |
| 2. | Brake Lights | Brake light system compatible with the ML model reacts within 0.1 second. |
| 6. | Steering Control | Actuation from the motor was seen each time commands were sent from the Orin. These commands included a variety of direction and pulse length (how long to spin motor). |
| 7. | Steering Limitation | Due to the Arduino logic onboard, the steering motor was limited to the angle we determined safe for controlling the steering. This was ensured by attempting to over rotate the motor through the signal from the Orin. |

**Conclusions**

The second prototype demonstrated significant progress in terms of functionality, accuracy, and responsiveness across all key systems. The key fob performed well, with a 80% success rate even at distances exceeding 10 meters, allowing easy location identification of the trike. In order to achieve a higher success rate, we need to find a way to mount the RF receiver without interference from the metal frame of the trike. The manual brake system also functioned perfectly, engaging the brakes within 0.2 seconds and ensuring full stop within 2 seconds when the trike was gently pushed. The automated brake control based on the OAK-D camera worked as intended, activating the brakes when objects entered within 2 meters of the camera’s sightline, with consistent performance at the target 90% accuracy. The manual brakes also overrode the automated system which shows our success in prioritizing safety.

The brake lights responded within 0.1 seconds of receiving a brake command from the Orin Nano, ensuring timely signaling. Furthermore, the steering motor reliably responded to directional commands from the Orin, with the motor rotating accurately and within the prescribed 60-degree range.

A key takeaway from this phase of testing was the issue with RF signal interference with the LEDs, which was resolved by isolating the RF signal on a separate Arduino, ensuring that both systems operated without conflict. The steering motor’s range limitation was successfully managed by the onboard Arduino logic, preventing any over-rotation and ensuring that the steering remained safe and controlled.

In conclusion, the prototype is performing at a high level of accuracy and reliability across all tested functions. This is important to ensure our trike is safe with redundant braking systems. The next steps will be to integrate a battery and clean up our wiring to allow our trike to be mobile.