Boston University
Electrical & Computer Engineering
EC463 Capstone Senior Design Project

Final Testing Plan & Report

The Pollux
by Team 22

Wenhao Cao wenhaoc@bu.edu
Karl Carisme karlc@bu.edu
Louis Cicala cicala@bu.edu
Noah Hathout nhathout@bu.edu
Caelan Wong caewong@bu.edu

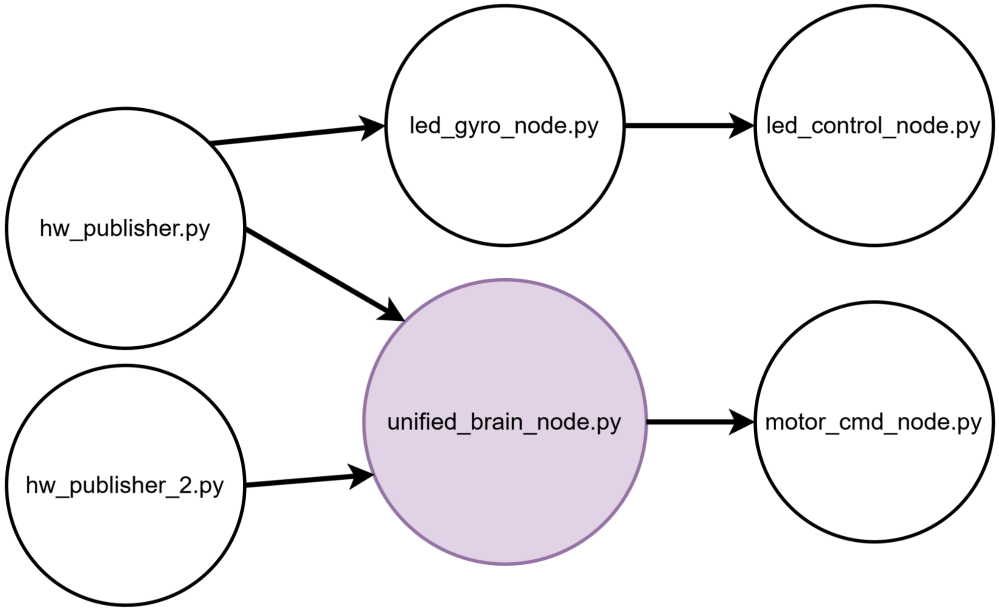
Equipment

The Pollux is an autonomous countertop cleaning robot with hardware and software components. Beginning with the hardware, the Pollux is built around a Raspberry Pi and is comprised of the following electronic components:

- Raspberry Pi 4B
- 5 x HR-SC04 Ultrasonic Sensors
- Adafruit MPU-6050 IMU
- 2 x IPSG Step Motor Driver Board + 5V Stepper Motor
- 1 x Green LED “ON/OFF”
- 3 x Red LED “Faux UVC”
- 1 x Orange LED “Indicator”
- 1 x Energizer UE20058 5V 20000mAh Battery Pack

Housing these components is a specially designed 3D-printed chassis, with mounts for the Raspberry Pi, motors, sensors, and the battery pack. Within the top of the chassis are magnets that keep the lid in place whilst also allowing for easy removal and access to the internals of the Pollux. Along with two wheels attached to the motors, on the bottom there is also a caster wheel to allow for stability and easy rotation.

On the software side, the robot’s movements are controlled using the Robotic Operating System (ROS Noetic) running on Ubuntu 20.04, with several ROS nodes dedicated to different tasks. Below is a flowchart detailing each of the ROS nodes, as well as what components each node subscribes to:



Below is a brief description of each node:

- **hw_publisher.py:** Publishes data from the IMU (on ROS topic /pollux/imu) and from the three downward-facing ultrasonic sensors (on /pollux/ultrasonic/hw).
- **hw_publisher_2.py:** Publishes data from the two forward-facing ultrasonic sensors (on /pollux/ultrasonic_2).
- **unified_brain_node.py:** Subscribes to the sensor topics published by the hardware nodes (bottom and/or front ultrasonic, plus any IMU data if needed) and decides the appropriate motor commands for navigation or avoidance maneuvers. This logic involves reacting to obstacles, cliffs, or other conditions to ensure safe and effective movement.
- **motor_cmd_node.py:** Subscribes to the Brain Node for motor command integers (e.g., forward, turn, stop). It then modulates the GPIO pins for each stepper motor, controlling the left and right motor drivers to execute these commands.
- **led_gyro_node.py:** Subscribes specifically to the IMU data (from the Hardware Nodes) to monitor the robot's tilt and angular velocity. If the robot's orientation is deemed unsafe (e.g., too much tilt), this node issues commands (published on /pollux/led_cmd) to turn off the UV-C LED strip for safety.
- **led_control_node.py:** Receives LED command codes (from the LED Gyro Node or other parts of the system) and drives the actual GPIO pins for the UV-C LEDs, indicator LEDs, and the general "robot on" LED. It serves as the low-level interface to physically turn LEDs on or off in response to higher-level logic.
- Web App: Although not yet fully implemented, reads commands from the web app and publishes them to the motor node

All of these nodes are initialized on boot and begin operation once the Raspberry Pi receives power, following a start-up delay of approximately one minute.

Setup

As just described, the Pollux begins operation on boot. Hence, the only setup required between the hardware and software is to power on the robot. Should our grader wish to see the internals of the robot during testing, the lid may first be removed. For bookkeeping purposes, here is are the GPIO pin connections of each hardware component:

Left Motor Driver		**Right Motor Driver**	
IN1	26	IN1	21
IN2	19	IN2	20
IN3	13	IN3	16

IN4	6	IN4	12
Top Left Ultrasonic Sensor		**Top Right Ultrasonic Sensor**	
Echo	4	Echo	27
Trig	17	Trig	22
Bottom Left Ultrasonic Sensor		**Bottom Middle Ultrasonic Sensor**	
Echo	18	Echo	24
Trig	23	Trig	25
Bottom Right Ultrasonic Sensor		**AMU Gyroscope / Accelerometer**	
Echo	8	SDA	2
Trig	7	SCL	3
LEDs		ON/OFF	9
Faux UVC	10	Indicator	11

Testing Procedure

In our final testing, we will demonstrate three capabilities of the Pollux in combination with its primary function of mimicked sanitization:

1. Cliff detection
2. Obstacle detection
3. Gyroscopic LED failsafe

Here is a compact testing procedure to analyze these capabilities:

1. Clear off a lab desk and place the Pollux on it
2. Power on the Pollux
3. Allow the Pollux to reach cliffs at least 10 times
    a. Measure whether the Pollux stops and turns (✓) or falls (✖)
4. Place obstacles on the desk in front of the Pollux at least 10 times
    a. Measure whether the Pollux stops and turns (✓) or collides (✖)
5. While moving, tilt the Pollux upwards and observe the underside at least 5 times
    a. Measure whether the bottom LEDs turn off (✓) or stay on (✖)

Score Sheet

Cliff Detection	1	2	3	4	5	6	7	8	9	10
Pass/Fail										
Obstacle Detection	1	2	3	4	5	6	7	8	9	10
Pass/Fail										
Gyroscopic Failsafe	1	2	3	4	5	6	7	8	9	10
Pass/Fail										

Testing Analysis

After some brief debugging prior to our final testing, we were able to successfully demonstrate the capabilities of the Pollux as described above.

Analysis of Functional Requirements

Our client, Jennefer, described the Pollux to us as follows:

> "A self-operating countertop cleaning robot that can automatically clean surfaces. It will be able to map out and navigate surfaces to determine an effective pattern to clean them. It should be able to sense when it's at the edge of a counter and turn in the other direction. It should also be able to avoid obstacles. It contains brushes and a vacuum, allowing it to clean up messes and it will sterilize using UV light, preventing the need for harsh chemicals. It will have a visual indication of what mode it's on (mapping, cleaning, disinfecting)."

We can break this description down and use it as a metric to analyze completion of the Pollux:

- "A self-operating countertop cleaning robot that can automatically clean surfaces"

The Pollux works without human interaction and traverses countertops to mimic UV sanitization.

- "Able to map out and navigate surfaces to determine an effective pattern to clean them"

Soon after we began working on the Pollux, we decided the ability to map out a surface was not only highly complicated, but also unnecessary. Instead, we opted for an adjacent, simpler solution. Currently implemented for navigation is a probabilistic algorithm with a degree of randomness to guide which direction the robot turns when it reaches a cliff or an obstacle. Our more permanent solution, though, is a fully-trained reinforcement learning algorithm designed to navigate more efficiently. While the algorithm is finished and ready to be implemented, we have refrained from implementing it yet due to time restraints. As such, for the final project testing, we instead use our strictly-probabilistic navigation algorithm.

➢ “Able to sense when it’s at the edge of a counter and turn in the other direction”
The Pollux detects cliffs and avoids them.

➢ “Able to avoid obstacles”
The Pollux detects obstacles and avoids them.

➢ “Contains brushes and a vacuum, allowing it to clean up messes”
We greatly de-emphasized this function of the Pollux over the duration of our work. Functionally, these components are important to ensure the cleaning capabilities of the Pollux, but from a technical standpoint, they are incredibly simplistic. Therefore, we opted to focus on the more technologically challenging requirements, rather than spend the time developing a custom cleaning system. For this reason, we have yet to implement them, and are considering doing so during customer installation.

➢ “Sterilize using UV light, preventing the need for harsh chemicals”
The Pollux contains an array of red LEDs on its underside, meant to mimic UVC LEDs capable of sanitization. Generally, UV sanitization can be slow when using low-wattage diodes. Thus, for testing purposes, we have increased the speed of the robot in order to make edge and obstacle detection easier to observe. As a note, the true speed required for 99% sanitization would be hundreds of times slower than we have implemented for testing. We will, however, reduce the robot’s speed back down for customer installation, such that it fits the client’s needs as opposed to what was more beneficial to us during testing.

➢ “It will have a visual indication of what mode it’s on”
While we did not develop any kind of modal system, as we saw it an unnecessary addition, we have included a green LED to indicate when the robot is on or off, as well as an orange LED to indicate when the faux UVC diodes are active (when the robot is on ‘sanitization mode’).

Feedback

Following our testing, we received a few key pieces of feedback. A minor comment was to include resistors between our LEDs and their voltage source, as they were likely not receiving enough current, dimming them.

A more significant component of our feedback, though, was to implement some form of motion tracking prior to ECE day during our customer installation period. This feedback was given in response to a question raised as to how we track how much of a surface the Pollux would be able to cover. Our solution to this will be to implement our existing reinforcement learning algorithm into our ROS system, which should allow the robot to cover spaces more efficiently.