Prototype II and Customer Feedback

Andrew Bettin

Andrew Walisser

Connor Rennie

Wesley Savage

Vidvath Tanjore

University of Ottawa

GNG 1103 - Introduction to Engineering Design

Mr. David A. Knox

Sunday, March 9th, 2025

Table of Contents

Introduction

The purpose of this report is to develop the second prototype and create a detailed test plan for it. As outlined in previous reports, the primary objective is to design a device that will enable the EMED team at Canadian Nuclear Laboratories to safely collect a 30-80 milligram sample of metal from inside a fuel channel supplying uranium fuel to a CANDU reactor.

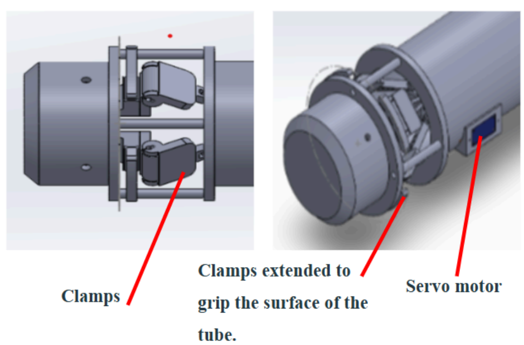
Earlier reports have already identified and categorized the project needs based on their importance. Design criteria were established for each of these needs, and the first prototype was created, along with a comprehensive list of materials required for its construction, including the cost of each item. Additionally, a set of tests to analyse the functioning of each part of the first prototype was designed.

The goal of this report is to incorporate the feedback provided by the client on the first prototype, making the necessary changes to the design based on the inputs provided. Furthermore, this report will outline the test plan for Prototype II, detailing the tests to be conducted, and documenting the results of these tests.

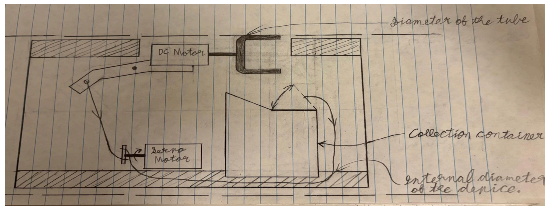
Client Feedback

The table below outlines the feedback provided by the client regarding Prototype I:

Table 1: Feedback Provided by the Client.

Part of the Design.	Description of the Part.	Feedback provided regarding the part.
Drivetrain.	 **Figure 1: Clamping mechanism.**  **Figure 2: Drive mechanism.**	The client approved this design. However, during the presentation, the client did mention to another group that the use of the rack and pinion system may be slow. Based on this input, changes may be made to the design to ensure that movement in and out of the tube is fast and efficient.

	- The clamps in Figure 1 extend outwards to grip the surface of the tube. - With this side of the tube fixed, the middle and the other end of the tube are driven forward by a rack and pinion system (located inside the middle portion) driven by a stepper motor. - Once the movement of the rack and pinion system is complete and the device is extended to its maximum length, the clamps at the other end engage to allow the free end to move.	
Communication and power supply.	The communication system consists of two ESP32 microprocessors. One ESP32 is mounted inside the device, while the other is placed at the end of the tube. These two microprocessors are wirelessly connected and exchange signals, which will be used to determine the distance. The electricity required for the functioning of the various components in the device is provided by a 11 V LiPo battery, through a circuit.	The client did not provide any negative feedback regarding the communication or power supply systems. However, it is important to note that a circuit diagram detailing the connections to the motors, sensors, and other electrical components was not presented to the client, so no feedback was received. As a result, the circuit will

		need to be tested and approved by the team itself.
Sample collection and storage.	 **Figure 3: Collection and storage mechanism.** The scraping tool, which consists of a scraping blade and a DC motor, is mounted on a support. This support is attached to the device's circumference using a steel rod. The support can rotate along the rod’s axis, which is perpendicular to the device’s axis. The collection container is positioned directly below the scraping tool to catch the sample. A servo motor, mounted on the bottom surface of the tube, controls cables connected to both the container’s lid and the scraping tool support. When the motor applies tension to the cables, the support holds the scraping tool against the tube’s	The client has approved the scraping system. However, there is a concern about the collection container’s ability to collect the sample when the device operates in a vertically oriented tube. The issue arises because the tip of the container is positioned far from the scraping tool's blades. As a result, the sample will not fall into the container but instead into the device itself. This prevents proper sample collection and leads to contamination of the device with sample particles. The collection container will have to be redesigned in order to rectify this discrepancy.

	surface, and the container's lid remains open for sample collection. When the motor reduces tension, the support moves inward, pulling the scraping tool away from the tube's surface to prevent unnecessary contact. At the same time, the container's lid closes, ensuring the sample does not come into contact with the device operator.	

Updated Design of the Collection and Storage mechanism

Based on the feedback provided by the client regarding the sample collection and scraping mechanisms, two new design concepts were developed. They are as follows:

1. **Design concept 1**

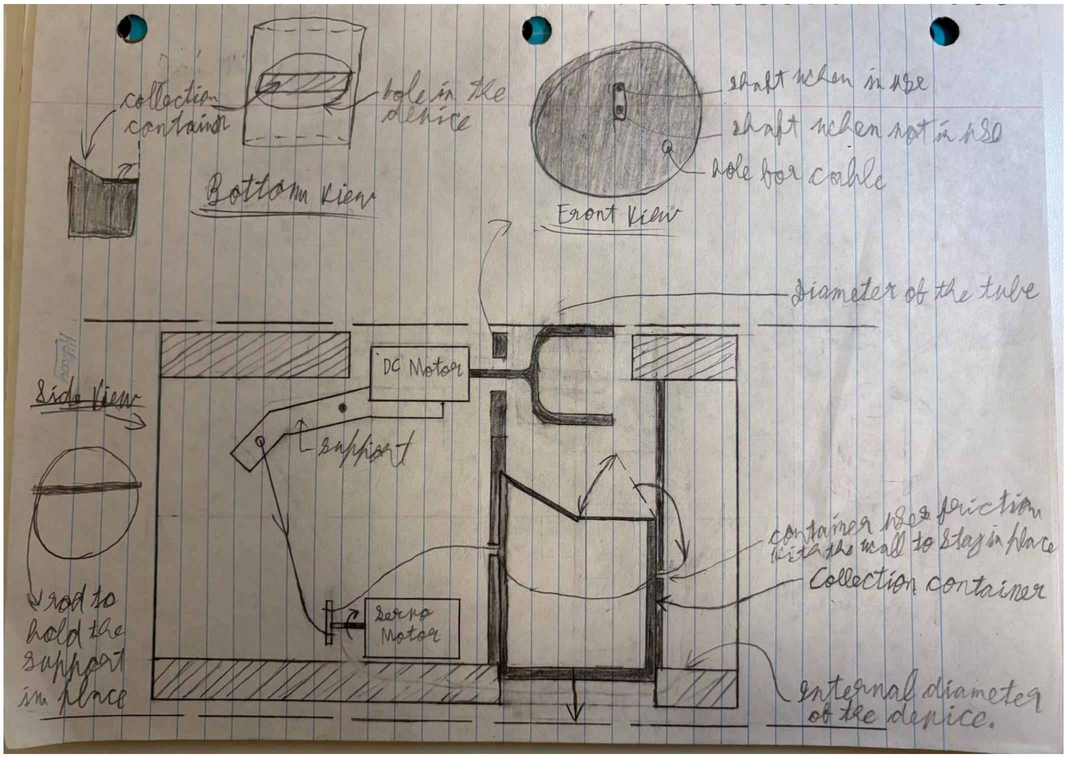


Figure 4: Design concept 1.

The collection and storage mechanism shown in Figure 4 is similar to the one previously presented to the client. However, a few modifications have been made to ensure smooth and efficient sample collection.

The first major change is the addition of two side walls on either side of the collection container. As shown in the front view, these side walls enclose the entire circumference of the device's interior while still allowing the DC motor shaft and the cable for opening and closing the lid to pass through designated holes. The primary function of these walls is to prevent the sample from falling into other parts of the device while ensuring that it is directed into the collection container.

An important consideration in a vertically oriented setup is that the sample will initially rest on the side walls rather than falling directly into the container. However, once the device is removed from the vertical tube and placed horizontally, gravity will naturally cause the sample to fall into the container.

Another key modification is the addition of a circular hole at the bottom of the device. This hole allows the collection container to be inserted into and removed from the tube efficiently. In terms of positioning, the collection container will remain in constant contact with the side walls, and the friction generated by this contact will help keep it securely in place.

2. **Design concept 2**

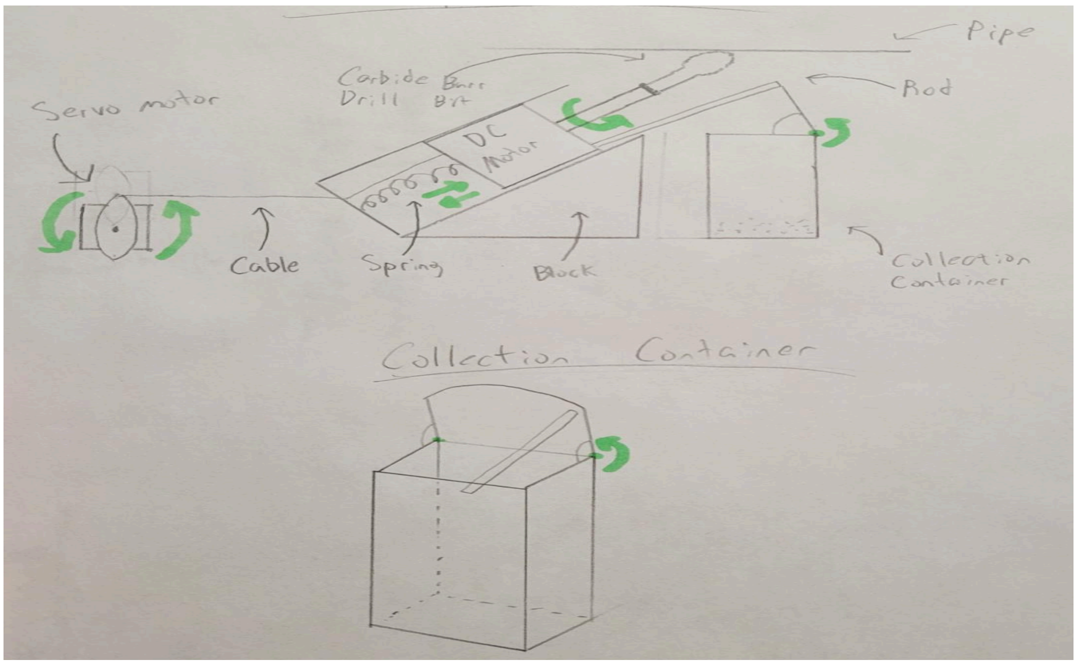


Figure 5: Design concept 2.

Here is how the concept represented in Figure 5 works: The servo motor actuates, releasing a spring that allows the drill bit to come into contact with the pipe while the DC motor begins spinning allowing the drill bit to scrape off a sample. The rod between the DC motor and the collection system allows it to be opened as the drill bit extends towards the tube and closes when the drill bit is retracted.

Circuit Diagram and Components

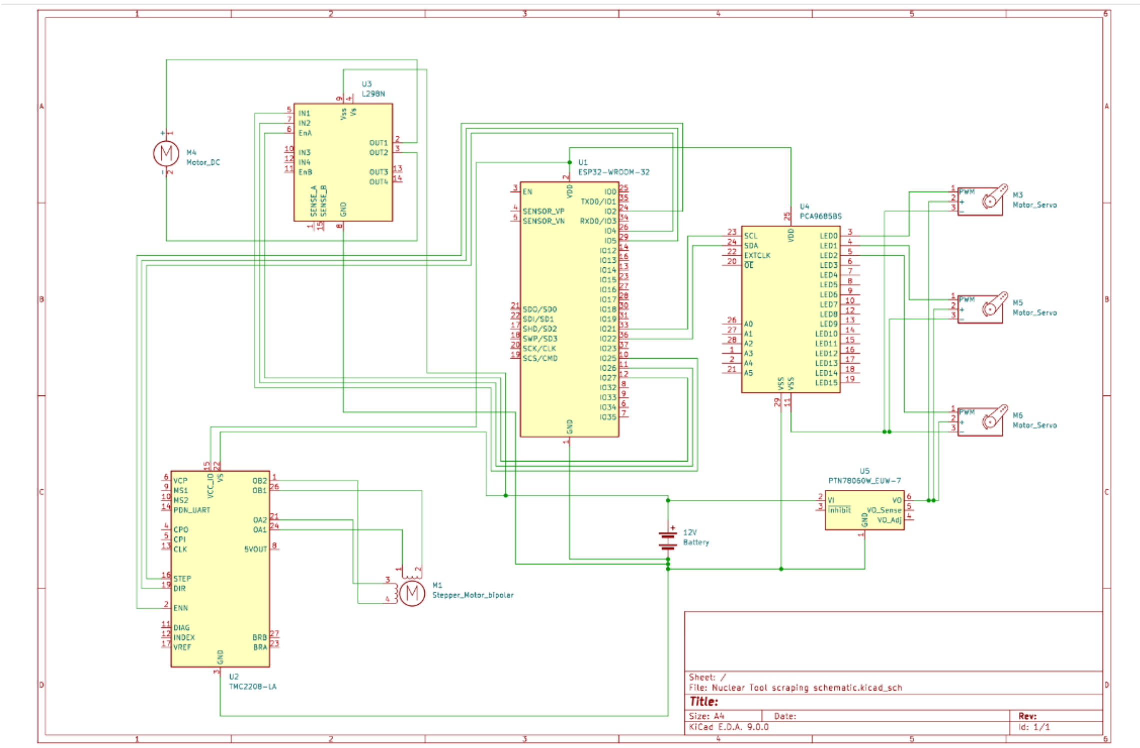


Figure 6: Diagram representing the circuit connecting all the electrical components of the device.

In addition to the various motors and microprocessors used in the device, the circuit shown in Figure 6 consists of the following components:

1. 11 V LiPo battery as the power source.
2. 6V buck convertor to act as a step down module for the servo motors.
3. 1 uF and 100 uF capacitors to store energy and prevent voltage spikes.
4. L298N driver for the DC motor.
5. PCA9865 driver for the servo motor.

All of these electrical components will be connected together using wires with connectors at the ends.

Updated Bill of Materials

The table below represents the updated bill of materials for the device:

Table 1: Table representing the updated bill of materials for the device.

Item No.	Component	Description	Qty.	Cost (CAD)
1	ESP32 Dev Board	Main Microcontroller (Wi-Fi, Bluetooth)	2	$23.95
2	TMC2208 Stepper Motor Driver	Stepper driver (for rack and pinion movement control)	1	$10.19
3	L298N DC motor Driver	Motor driver for scraper rotation movement	1	$2.51
4	PCA9685 Servo Driver	16 channel PWM servo driver	1	$8.33
5	6V Buck converter	Step down module for servo motors	1	$2.89
6	NEMA 17 Stepper motor	Rack and Pinion Movement	1	$14.07
7	DC Motor	Scraping rotation	1	$8.99
8	MG995 Servo Motor	High torque servo	3	$15.36
9	3S 11.1V LiPo Battery	Power Source	1	$16.20
10	Wires and Connectors	Set of 40		$2.95
11	1 uF Capacitor		1	$1.33
11	100 uF Capacitor		1	$0.12
14	Pinion Gear	Carbon Fiber Mix Filament (3d printed)	1	$0.77
15	Rack	Carbon Fiber Mix Filament (3D printed)	1	$1.34
16	Carriage support	Polycarbonate Filament (3d printed)	1	$0.52
17	Clamp Carriage	Polycarbonate Filament (3d printed)	4	$5.40
18	Central Arm	Polycarbonate Filament (3d printed)	4	$1.80
19	Central Arm Connector	Polycarbonate Filament (3d printed)	1	$0.38
20	Servo Guide	PLA (1 roll)	1	$30.00
21	Servo Guide Rail	PLA	1	|
22	End Caps	PLA	1	|
23	Back Section	PLA	1	|
24	Front Section	PLA	1	|
25	Middle Section	PLA	1	|
26	SG-5 Carbide Bur	Scraping Tool	1	$18.23
27	M5x0.8mm Screws	Fastener	8	$15.00
28	#12-28 Screws	Fastener	8	$5.18
				Total Cost: $185.53

Objectives

The objective of this prototype is to test and refine the code that interfaces with the microprocessor. This code controls the motors responsible for moving the device, operating the scraping tool, and controlling the lid of the collection container. Any bugs that arise during testing will be identified and fixed.

In addition to testing the code, several other components will be evaluated. These include the clamps that grip the tube's surface, the scraping tool, and the collection process in both vertical and horizontal orientations. The performance of the motors, the rack-and-pinion system for movement, the cables operating the scraping mechanism and lid, and the microprocessors measuring the distance traveled will also be evaluated.

Analysis of Critical Components and Systems

The critical components and systems of this prototype are similar to those in Prototype I. However, certain new components have been added as critical. Below is an updated list of the critical components identified in this project:

1. **Clamping Mechanism** – The clamping mechanism is essential because, without a secure grip on the tube, the rest of the device will not be able to function properly.

2. **Rack and Pinion System** – This system must operate smoothly to enable forward and backward movement. In addition to ensuring smooth operation, the speed of the system is also critical, as it directly affects the efficiency of the device.

3. **Scraping System** – The scraping mechanism must efficiently collect between 30 and 80 grams of the sample. To achieve this, the scraping tool and motor must be able to make contact with the tube’s surface when collecting samples and retract when collection is not in progress.

4. **Collection System** – The collection system must gather all scraped samples while preventing contamination by ensuring that no material falls into other parts of the device. Additionally, the system must be effective in both horizontal and vertical orientations.

5. **Motors and Sensors** – These components must function efficiently to perform their respective roles, ensuring the overall operation of the device.

6. **Failsafe Winch** – This is a critical safety component, as it provides the only means of retrieving the device from the tube in case of motor failure or power loss.

Prototyping Test Plan

Table 2: Table representing the test plan for prototype II.

Part being tested	**How will it be tested ?**	**Type of test**	**What is being tested ?**
Movement System (Rack and Pinion)	The system will be tested in two ways. The first method will be a physical test in which the device will be operated within a 15 ft long pipe, and the distance traveled per inchworm movement will be measured. Through multiple trials, the range of movement per cycle will be determined, allowing	The physical test will be a focused, high-fidelity test. The outer shell of the device will be constructed, containing only the necessary components required for this evaluation. The test on the simulation will be one of medium-fidelity as it does not travel 15 ft and also does not account for conditions such as the	The physical test will measure the total distance traveled by the device. It is essential to ensure that the device moves a predetermined length consistently and reliably across multiple executions. The distance traveled will be recorded in meters. The goal of the simulation

	for the calculation of the number of movements required to travel a distance of 15 feet. The second method involves a test of the movement system using a Solidworks system.	friction among moving parts, which adds to the load applied on the system, which cannot be tested.	test is to simply demonstrate that the movement system moves as it should, with the rack and pinion system driving the middle and free end of the device in a straight line.
Clamping system.	The system will be tested in two ways. The first method involves physically testing the system inside the tube to ensure that the clamps are able to extend up to its diameter (101.6 mm) and grip its surface. The second method of testing involves using a Solidworks simulation to demonstrate the extension of the clamps up to an imaginary circle, which is of the same diameter as the tube.	The physical test will be a high-fidelity test which will involve a test on all the components of the ends of the device. This includes the body part, the four clamping legs, and the servo motor that operates the system. The test on the simulation will involve the same parts. However, this will be a test of medium-fidelity as it does not does not account for conditions such as the friction among moving parts, which adds to the load applied on the system, which cannot be tested.	The goal of the physical test is to test the extension and the gripping of the clamps, which will then allow the rest of the device to move. The goal of the simulation test is to demonstrate the extension of the clamping legs. Unfortunately, the gripping of the clamps onto the surface of the tube cannot be tested as the simulation does not demonstrate whether there is enough friction between the clamps and the surface of the tube.

Scraping System	To evaluate the scraping system, a material that produces a dust-like substance when scraped, such as chalk, will be used. The test will involve timing the duration required to remove 30-80 mg of material. This trial will be repeated multiple times, and the results will be averaged to determine the precise operating duration for the scraper. This test will be conducted in horizontal and vertical orientations of the tube.	This test will be a focused, medium-fidelity evaluation. The scraping arm will be fully constructed; however, initial tests may not be conducted against a metal tube. Instead, materials such as chalk or PVC tubes may be used to verify proper operation. In later comprehensive tests, performance against a metal tube will be assessed to ensure efficacy.	This evaluation will measure the amount of material removed from the test surface. Additionally, the rate of material removal will be recorded to determine an average. Data will be collected in milligrams of material scraped, total scraping time in seconds, and material removal rate in milligrams per second. The idea of conducting the test in horizontal and vertical orientations of the device is to ensure that the blade is able to
Container	The primary requirement for testing the container is to ensure it captures the material and successfully closes its lid. To conduct this test, the container will be	This test will be a focused, low-fidelity evaluation. It is expected to coincide with the testing of the scraper tool, as both components are functionally connected. The objective is to achieve a high collection rate of the	This test will measure the amount of material that falls into the collection container. The objective is to ensure that the majority, if not all, of the scraped material is captured within the container rather than dispersing into the

	positioned beneath the scraper during operation to verify that the majority of the sample is collected. Following this, the functionality of the lid will be confirmed to ensure proper closure. This test will be conducted in horizontal and vertical orientations of the tube.	material removed during the scraping process.	surrounding tube. The evaluation will involve quantifying the material collected in the container versus the amount that falls into the tube and calculating the ratio between these two values in milligrams. The idea of conducting the test in horizontal and vertical orientations of the device is to ensure complete collection of the sample and to prevent it from being deposited to other parts of the device. Special emphasis will be given to this part of the test as the client has highlighted a discrepancy in this area, and it is expected that the team does a good job of correcting it.
Code.	The code used to operate the microprocessor, motors, sensors, etc will be	The physical test will be a high-fidelity one involving the code and other electronic components such	The goal of the physical test is to determine how the various electronic components respond to

	tested in two ways. The first method involves physically testing the code in the presence of all the other components such as the microprocessor, motors, and sensors. The second method involves uploading the code to an AI application such as ChatGPT to detect any bugs.	as the microprocessor, motors, and sensors. The test on ChatGPT will be a low-fidelity one as the electronic components on which the code acts are not present and as a result, the response of these devices cannot be assessed. Additionally, there is a concern regarding the reliability of the inputs provided by the AI application.	the commands provided by the code. The goal of the test on ChatGPT is to simply test for any errors in the code and make the necessary changes.
Failsafe	Testing the failsafe requires ensuring a secure attachment at both the winch and the machine to prevent disconnection. To verify this, the machine will be positioned 15 feet from the winch, with resistance applied to the machine once the winch is engaged to confirm that the attachment remains secure.	This test will be a comprehensive, high-fidelity evaluation. Since the failsafe serves as the final safeguard against errors in the device, testing should be conducted only after a complete prototype is available to ensure it meets all required specifications.	This test will evaluate whether the winch can successfully retract the device without causing additional damage to the surrounding tube. The assessment will be conducted on a pass/fail basis rather than relying on measurable metrics to determine success.

Results of the Test on the Code

The following code was tested using ChatGPT:

```
#include <Servo.h>
#include <AccelStepper.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <ArduinoJson.h>

// Servo Configuration
#define SERVO1_PIN 12
#define SERVO2_PIN 13
Servo clampServo1;
Servo clampServo2;

// stepper Motor config
#define STEPPER_IN1 14
#define STEPPER_IN2 27
#define STEPPER_IN3 26
#define STEPPER_IN4 25
#define STEPS_PER_REV 2048
#define GEAR_REDUCTION 64
#define PINION_DIAMETER 0.5   // Inches

// movement Parameters
const float INCHES_PER_REV = PI * PINION_DIAMETER;
const float INCHES_PER_STEP = INCHES_PER_REV / (STEPS_PER_REV *
GEAR_REDUCTION);
```

```
AccelStepper stepper(AccelStepper::HALF4WIRE, STEPPER_IN1, STEPPER_IN3,
STEPPER_IN2, STEPPER_IN4);

// web config
const char* ssid = "RobotControl";
const char* password = "robot1234";
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");

// battery monitoring
#define BATTERY_PIN 34        // Voltage divider input
#define ADC_REF_VOLTAGE 3.3   // ESP32 reference voltage
#define ADC_RESOLUTION 4095.0 // 12-bit ADC

// system state
float current_position = 0;
float battery_voltage = 0.0;
unsigned long last_update = 0;

void setup() {
  Serial.begin(115200);

  // mechanics initialization
  clampServo1.attach(SERVO1_PIN);
  clampServo2.attach(SERVO2_PIN);
  closeClamps();

  stepper.setMaxSpeed(1000);
  stepper.setAcceleration(500);

  // battery monitoring initilization
```

```
  analogReadResolution(12);

  // WiFi AP
  WiFi.softAP(ssid, password);
  Serial.print("AP IP: ");
  Serial.println(WiFi.softAPIP());

  // web server
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html);
  });

  ws.onEvent(onWebSocketEvent);
  server.addHandler(&ws);

  server.begin();
}

void loop() {
  static unsigned long last_send = 0;

  // main movement sequence
  openClamps();
  moveRobot(6.0);  // Move forward 0.5 feet
  closeClamps();
  moveRobot(-6.0); // Return

  // update battery and send data
  if(millis() - last_send >= 200) {
    updateBattery();
    sendWebData();
```

```
    last_send = millis();
  }
}

//mechanical functions
void openClamps() {
  clampServo1.write(90);
  clampServo2.write(90);
  delay(1000);
}

void closeClamps() {
  clampServo1.write(180);
  clampServo2.write(180);
  delay(1000);
}

void moveRobot(float inches) {
  long steps = inches / INCHES_PER_STEP;
  stepper.move(steps);

  while(stepper.distanceToGo() != 0) {
    stepper.run();
    current_position = stepper.currentPosition() * INCHES_PER_STEP;
  }
}

//web functions
void updateBattery() {
  float adc_value = analogRead(BATTERY_PIN);
  float voltage = adc_value * (ADC_REF_VOLTAGE / ADC_RESOLUTION);
```

```
  battery_voltage = voltage * 3.0; // Adjust multiplier based on voltage divider
}

void sendWebData() {
  DynamicJsonDocument doc(128);
  doc["distance"] = abs(current_position);
  doc["voltage"] = battery_voltage;
  doc["moving"] = (stepper.speed() != 0);

  String json;
  serializeJson(doc, json);
  ws.textAll(json);
}

void onWebSocketEvent(AsyncWebSocket *server, AsyncWebSocketClient *client,
                     AwsEventType type, void *arg, uint8_t *data, size_t len) {
  if(type == WS_EVT_CONNECT) {
    Serial.println("Client connected");
  }
}

// ======== HTML CONTENT ========
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <title>Robot Monitor</title>
  <style>
    .container {
      display: flex;
      flex-direction: column;
```

```
      align-items: center;
      font-family: Arial, sans-serif;
    }
    .battery {
      width: 100px;
      height: 200px;
      border: 3px solid #333;
      border-radius: 5px;
      position: relative;
      margin: 20px;
    }
    #battery-level {
      position: absolute;
      bottom: 0;
      width: 100%;
      background: #4CAF50;
      transition: height 0.3s ease;
    }
    .distance {
      font-size: 2em;
      margin: 20px;
    }
    .status {
      font-size: 1.2em;
      color: #666;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Robot Monitor</h1>
```

```
  <div class="battery">
    <div id="battery-level"></div>
  </div>
  <div id="voltage" class="status">Voltage: 0.00V</div>

  <div class="distance">
    Distance: <span id="distance">0.00</span> in
  </div>

  <div id="status" class="status">Status: Connected</div>
</div>

<script>
  const ws = new WebSocket('ws://' + window.location.hostname + '/ws');

  ws.onmessage = function(event) {
    const data = JSON.parse(event.data);

    // Update battery display
    const batteryLevel = document.getElementById('battery-level');
    const voltageDisplay = document.getElementById('voltage');
    const minVoltage = 10.0;
    const maxVoltage = 12.6;

    const percent = Math.min(Math.max(
      ((data.voltage - minVoltage) / (maxVoltage - minVoltage)) * 100,
      0
    ), 100);

    batteryLevel.style.height = percent + '%';
```

```
      voltageDisplay.textContent = `Voltage: ${data.voltage.toFixed(2)}V`;

      // Update distance
      document.getElementById('distance').textContent = data.distance.toFixed(2);

      // Update status
      document.getElementById('status').textContent =
        `Status: ${data.moving ? 'Moving' : 'Stopped'}`;
    };

    ws.onclose = function() {
      document.getElementById('status').textContent = 'Status: Disconnected';
    };
  </script>
</body>
</html>
)rawliteral";
```

The prompt used is as follows: “Test this code and debug it”

The result provided was a debugged code:

```
#include <Servo.h>
#include <AccelStepper.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <ArduinoJson.h>

// Servo Configuration
#define SERVO1_PIN 12
#define SERVO2_PIN 13
```

```
Servo clampServo1;
Servo clampServo2;

// Stepper Motor Configuration
#define STEPPER_IN1 14
#define STEPPER_IN2 27
#define STEPPER_IN3 26
#define STEPPER_IN4 25
#define STEPS_PER_REV 2048
#define GEAR_REDUCTION 64
#define PINION_DIAMETER 0.5   // Inches

// Movement Parameters
const float INCHES_PER_REV = PI * PINION_DIAMETER;
const float INCHES_PER_STEP = INCHES_PER_REV / (STEPS_PER_REV *
GEAR_REDUCTION);
AccelStepper stepper(AccelStepper::HALF4WIRE, STEPPER_IN1, STEPPER_IN3,
STEPPER_IN2, STEPPER_IN4);

// Web Config
const char* ssid = "RobotControl";
const char* password = "robot1234";
AsyncWebServer server(80);
AsyncWebSocket ws("/ws");

// Battery Monitoring
#define BATTERY_PIN 34        // Voltage divider input
#define ADC_REF_VOLTAGE 3.3   // ESP32 reference voltage
#define ADC_RESOLUTION 4095.0 // 12-bit ADC

// System State
```

```
float current_position = 0;
float battery_voltage = 0.0;
bool isMoving = false;
unsigned long last_update = 0;

void setup() {
  Serial.begin(115200);

  // Mechanics Initialization
  clampServo1.attach(SERVO1_PIN);
  clampServo2.attach(SERVO2_PIN);
  closeClamps();

  stepper.setMaxSpeed(1000);
  stepper.setAcceleration(500);

  // Battery Monitoring Initialization
  analogReadResolution(12);

  // WiFi Access Point
  WiFi.softAP(ssid, password);
  Serial.print("AP IP: ");
  Serial.println(WiFi.softAPIP());

  // Web Server
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html);
  });

  ws.onEvent(onWebSocketEvent);
  server.addHandler(&ws);
```

```
  server.begin();
}

void loop() {
  static unsigned long last_send = 0;
  stepper.run();

  if (millis() - last_send >= 200) {
    updateBattery();
    sendWebData();
    last_send = millis();
  }
}

// Mechanical Functions
void openClamps() {
  clampServo1.write(90);
  clampServo2.write(90);
  delay(1000);
}

void closeClamps() {
  clampServo1.write(180);
  clampServo2.write(180);
  delay(1000);
}

void moveRobot(float inches) {
  long steps = inches / INCHES_PER_STEP;
  stepper.move(steps);
  isMoving = true;
```

```
}

// Web Functions
void updateBattery() {
  float adc_value = analogRead(BATTERY_PIN);
  float voltage = adc_value * (ADC_REF_VOLTAGE / ADC_RESOLUTION);
  battery_voltage = voltage * 3.3; // Adjust multiplier based on voltage divider
}

void sendWebData() {
  DynamicJsonDocument doc(128);
  doc["distance"] = abs(current_position);
  doc["voltage"] = battery_voltage;
  doc["moving"] = isMoving;

  String json;
  serializeJson(doc, json);
  ws.textAll(json);
}

void onWebSocketEvent(AsyncWebSocket *server, AsyncWebSocketClient *client,
AwsEventType type, void *arg, uint8_t *data, size_t len) {
  if (type == WS_EVT_CONNECT) {
    Serial.println("Client connected");
  } else if (type == WS_EVT_DATA) {
    String msg = String((char*)data).substring(0, len);
    if (msg == "MOVE_FORWARD") {
      openClamps();
      moveRobot(6.0);
      closeClamps();
    } else if (msg == "MOVE_BACKWARD") {
```

```
            openClamps();
            moveRobot(-6.0);
            closeClamps();
        }
    }
}
```

Results of the Simulation Tests of the Clamping and Drive Mechanisms.

Please click on the links attached to this submission to access videos of the simulation tests. These simulation tests have successfully demonstrated what was specified in the test plan.

Round Two of Testing Plans

As the physical tests on the components have not been completed due to the pending approval of the BOM, proceeding with the development of follow-up prototypes and testing is not advisable at this stage. Once data from the physical tests has been collected and analyzed, a re-evaluation will be conducted to ensure that subsequent physical tests are appropriately targeted based on the initial results.

Conclusion

After finalizing the design and verifying its logical and mathematical accuracy, a comprehensive round of physical and simulation-based testing was planned to assess all critical components. While the simulation tests were successfully conducted, the physical tests could not be performed due to a lack of materials. Once these tests are completed, additional testing will be planned, increasing in scope and accuracy to ensure the device functions as intended.

Objectives

The objective of this prototype is to test and refine the code that interfaces with the microprocessor. This code controls the motors responsible for moving the device, operating the scraping tool, and controlling the lid of the collection container. Any bugs that arise during testing will be identified and fixed.

In addition to testing the code, several other components will be evaluated. These include the clamps that grip the tube's surface, the scraping tool, and the collection process in both vertical and horizontal orientations. The performance of the motors, the rack-and-pinion system for movement, the cables operating the scraping mechanism and lid, and the microprocessors measuring the distance traveled will also be evaluated.

Analysis of Critical Components and Systems

The critical components and systems of this prototype are similar to those in Prototype I. However, certain new components have been added as critical. Below is an updated list of the critical components identified in this project:

7. **Clamping Mechanism** – The clamping mechanism is essential because, without a secure grip on the tube, the rest of the device will not be able to function properly.

8. **Rack and Pinion System** – This system must operate smoothly to enable forward and backward movement. In addition to ensuring smooth operation, the speed of the system is also critical, as it directly affects the efficiency of the device.

9. **Scraping System** – The scraping mechanism must efficiently collect between 30 and 80 grams of the sample. To achieve this, the scraping tool and motor must be able to make contact with the tube’s surface when collecting samples and retract when collection is not in progress.

10. **Collection System** – The collection system must gather all scraped samples while preventing contamination by ensuring that no material falls into other parts of the device. Additionally, the system must be effective in both horizontal and vertical orientations.

11. **Motors and Sensors** – These components must function efficiently to perform their respective roles, ensuring the overall operation of the device.

12. **Failsafe Winch** – This is a critical safety component, as it provides the only means of retrieving the device from the tube in case of motor failure or power loss.