Prototype I and Customer Feedback

Vidvath Tanjore

Wesley Savage

Andrew Bettin

Connor Rennie

Andrew Walisser

University of Ottawa

GNG 1103 - Introduction to Engineering Design

Mr. David A. Knox

Sunday, March 2nd, 2025

Introduction

The purpose of this report is to develop the first 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 several conceptual design concepts were developed. The final design of the prototype was created, and a comprehensive list of materials required for its construction was provided, including the cost of each item. Additionally, a schedule for both the construction and testing phases of the prototype was outlined.

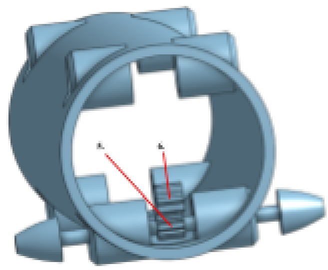
The goal of this report is to incorporate the feedback provided by the TA and PM, making the necessary changes to the design based on their input. Furthermore, this report will outline the test plan for Prototype I, detailing the tests to be conducted, and documenting the results of these tests.

TA / PM Feedback

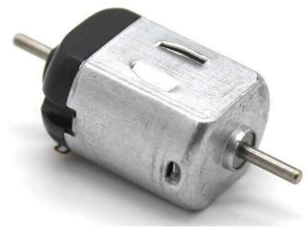
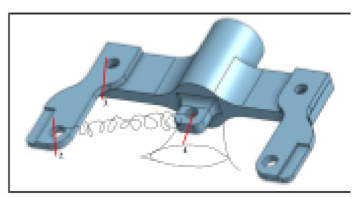
The TA and PM identified several issues with the design that was presented in the meeting. The table below outlines the detailed feedback provided regarding the design:

Table 1: Feedback Provided by the TA/PM.

Part of the Design.	**Description of the Part.**	**Feedback provided regarding the part.**

<table>
<tr>
<td>Drivetrain.</td>
<td>
Figure 1: Assembly of the drivetrain.

The drivetrain (Figure 1) allows the device to be driven through the tube by a DC motor whose shaft is connected to gear 6, which transfers power to gear 5, which in turn drives the rear wheels. It is important to note here that when the motor is not spinning, the device will not move as the gears are interlocked. Additionally, the shaft of the rotary encoder will be connected to gear 6 and this will allow it to measure the distance travelled.</td>
<td>The TA and PM have approved this part of the design for use in a horizontally oriented tube, where the device will be able to move easily and stop at the correct position, with the distance traveled being accurately measured. However, there is concern about the feasibility of implementing this design in a vertically oriented tube. In this orientation, the wheels may not be able to grip the surface of the tube effectively, preventing the device from moving upwards. Additionally, the device would not be able to stop at a specific point in the tube, as the motor would not be able to overcome the force of gravity acting on the device.</td>
</tr>
</table>

	 Figure 2: DC motor.  Figure 3: Rotary encoder.	
Sample collection and storage.	 Figure 4: Assembly of the scraping blade holders and the part that is connected to the DC motor. The scraping blade holders shown in Figure 4 are connected to a part that is connected to a DC motor. When the DC motor moves,	The TA and PM raised concerns that this system may not function as intended. There were doubts regarding the spring's ability to ensure that the blade holders remain in constant contact with the surface of the pipe. Additionally, concerns were raised about the vacuum pump's capacity to collect the sample through the funnel, as it is uncertain whether the pump can generate enough

	the blades scrape the sides of the cylinder to collect material. It is important to note that the blades are movable in order to prevent contact with the wall of the tube while the device is moving. This movement is achieved by a system that involves a cable connecting part 2 with a servo motor that is mounted inside the main body of the device. As the servo motor rotates, the tension in the cable builds and this pulls the blade holder inwards and keeps it away from the surface of the pipe. When the tension is released, the spring that connects part 2 and part 4 ensures that the blade returns to its original position. The blades used to scrape the sample are mounted on the depression in the blade holder. The storage of the collected sample involves the use of a	power to transport the sample to the container without it becoming stuck in the tube.

	vacuum pump that is mounted inside the main body of the pump.  Figure 5: Vacuum pump. The funnel shown in Figure 4 is connected to a tube that is connected to the input port of the vacuum pump. The pump sucks the sample through this funnel and tube and sends it to a detachable container (located inside the body of the device) via a tube connected to its output port.	

Design Concept

Tool Structure Overview

Movement System (Inchworm Mechanism)

- Entry into the Tube

    - The tool is inserted into the fuel channel tube through an access point.

- Flexible silicone cones on both ends ensure a sealed environment, preventing contamination and containing swarf (scraped material).

- Inchworm movement

- The tool moves using an inchworm mechanism, which consists of:
  - Two clamping legs (front and rear) that expand radially to grip the tube.
  - A linear extension system (cascade lifting mechanism or servo-driven linkage) that pushes the tool forward.
- How it Moves:

1. The rear clamp engages (grips the tube).
2. The front clamp releases, allowing the front section to extend forward.
3. The front clamp engages, securing the tool in its new position.
4. The rear clamp releases, allowing the rear section to be pulled forward.
5. The process repeats until the tool reaches the target depth (~15–16 feet inside the tube).

- Positioning and Feedback

- A rotary encoder tracks the distance traveled.
- The operator receives real-time depth feedback via the ESP32 wireless link.

Scraping System (Rotating Collar with Deployable Blades)

- Deployment of the Scraper

- Once the tool reaches the correct depth:
  - The scraper blades extend radially outward from the tool.
  - Servo-driven deployment system ensures precise control of the blade extension.

    - A mechanical stop or adjustable screw controls the depth of scraping.

- Circumferential Scraping Process

- The rotating collar begins spinning around the tool's central axis.
- The scraping blades remove a thin, uniform layer of the tube's inner surface.
- Load sensors on the blade arms monitor cutting resistance to prevent excessive force.

- Swarf Collection

- Scraped material is contained between the silicone cones at both ends.
- A vacuum system or brush system directs the material into a sealed collection chamber.

Sample Weighing and Containment

- Sample Transfer

- The scrapped material is funneled into a precision weighing chamber using a small servo-driven auger or funnel.
- The precision load cell (HX711-based system) measures the sample mass in real time.
- If the sample weight is below 30 mg, another scraping pass is triggered.
- If the weight exceeds 80 mg, the scraping stops.

- Containment System

- Once the correct sample mass is achieved:
    - The sample is transferred into a sealed storage capsule within the tool.
    - A servo-driven sliding door or latch secures the sample container.
- The storage container is removable so the operator can extract the sample safely after tool retrieval.

Retraction and Exit

- Blade Retraction
  - The scraper blades are retracted into their housing using a spring-loaded or servo-driven mechanism.
- Inchworm Retraction
  - The inchworm mechanism reverses direction to pull the tool back toward the entry point.
- Final Containment Check
  - The tool verifies sample containment before exiting the tube.
  - A sensor confirms the storage chamber is sealed to prevent contamination.

5.4 Operator Notification

- Once the tool reaches the entry point:
  - The operator receives a final status update via the ESP32 remote interface.
  - The sample container is safely removed for laboratory analysis.

Control and Monitoring System

- Dual ESP32 Microcontroller System
  - Tool ESP32:
    - Controls all actuators (servos, stepper motor).
    - Monitors sensors (distance, blade position, load cell, containment status).
    - Sends real-time data to the deployment station.
  - Deployment Station ESP32:
    - Receives data from the tool.
    - Displays system status on a UI (LCD screen, smartphone app, or PC dashboard).
    - Allows the operator to send commands (e.g., start/retract the tool).
- Sensor Feedback

- Rotary encoder → Tracks tool depth.
- Hall effect sensors → Verify clamp engagement.
- Load cell → Monitors sample weight.
- Blade position sensor - Confirms scraping readiness.
- Vacuum sensor - Ensures debris collection.

Power System

- Power Source

- Single 6S LiPo Battery (22.2V, 5000 mAh)
  - Compact and lightweight.
  - High discharge rate supports servos and the stepper motor.

- Voltage Regulation

- 24V Direct Line - Stepper motor.
- 6V Buck Converter - Servos.
- 5V Buck Converter - ESP32s, sensors, and logic circuits.

- Safety Features

- BMS (Battery Management System) → Prevents overcharging/discharging.
- Fuses:
  - 10A fuse for servos.
  - 5A fuse for stepper motor.
- Capacitors (1000 µF, 100 µF) → Reduce voltage spikes.

Objectives

The objectives of this prototype are to test the code that will interface with the microprocessor to control the motors responsible for moving the device, operating the scraping tool, and controlling the lid of the collection container. Additionally, the code must be tested to ensure that it enables the microprocessor to accurately receive data regarding the distance traveled by the device.

In addition to testing the code, several other components need to be tested. These include the clamps that grip the surface of the tube, the motors that control the various parts of the device, the rack and pinion system used to move the device, the cables that operate the scraping mechanism and the lid of the collection container, the blades that scrape the surface of the tube, the lid of the collection container, and the rotary encoder that measures the distance traveled.

Analysis of Critical Components and Systems

Here is a list of critical components that were identified in this project:

1. The clamping mechanism is a critical component as without the clamp gripping the tube on one side, the rest of the device will not be able to move.
2. The rack and pinion system must work smoothly in order to move the device forward and backward.
3. The scraping system must work efficiently to collect 30 - 80 grams of the sample. In order for this to occur, the scraping tool and motor must be able to come in contact with the surface of the tube, and must also be able to be moved away from the surface of the tube when collection is not going on.
4. The motors and sensors must work efficiently to perform their respective functions.
5. Lastly, the failsafe winch is a critical component as it is the only way through which the device can be retrieved from the tube in the event of a failure of the motors or power supply.

Prototype Testing Plan

Part being tested	**How will it be tested ?**	**Type of test**	**What is being tested ?**

Movement System (Rack and Pinion)	To evaluate the movement system, a pipe or tube with a 4-inch diameter will be used. The machine will be operated within the pipe, and the distance traveled per inchworm movement will be measured. Through multiple trials, the range of movement per cycle will be determined, allowing for the calculation of the number of movements required to travel a distance of 15 feet.	This 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.	This 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.
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	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,	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

	repeated multiple times, and the results will be averaged to determine the precise operating duration for the scraper.	performance against a metal tube will be assessed to ensure efficacy.	material removal rate in milligrams per second.
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 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 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 material removed during the scraping process.connected in function.	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 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.
Failsafe	Testing the failsafe requires ensuring a secure attachment at both the winch and the machine to prevent	This test will be a comprehensive, high-fidelity evaluation. Since the failsafe serves as the final safeguard against	This test will evaluate whether the winch can successfully retract the device without causing additional damage to the

	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.	errors in the device, testing should be conducted only after a complete prototype is available to ensure it meets all required specifications.	surrounding tube. The assessment will be conducted on a pass/fail basis rather than relying on measurable metrics to determine success.

Testing of The Code for the Microcontrollers

The following figures represent the code that will be used to operate the various parts of the device:

```
#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;
```

```css
  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";
```

Updated Bill of Materials

Table 1: Bill of Materials

Item	**Quantity**	**Total Cost**
DC motor	1	$ 0.00
Stepper motor	1	$ 15.00
ESP-32	2	$ 19.99 (for a set of 2)

Jumper cables	25	$ 2.95 (for a set of 40 cables)
6s 22.2V LiPo battery	1	$ 28.80
Battery holder (3d printed)	1	$ 0
Servo motor	3	$ 2.50
Carbide burr drill bit	1	$ 11.80
Rack and pinion (3d printed)	1	$ 0
Many 3D printed parts	Numerous	$ 0
Total Cost (exclusive of tax and shipping)		$ 86.04

Note - The items that are priced at $ 0.00 are those that the team already possesses or those that are available for free at stores such as IKEA or Home Depot.

Round Two of Testing Plans

As the first round of testing has 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 initial testing has been collected and analyzed, a re-evaluation will be conducted to ensure that subsequent tests are appropriately targeted based on the initial results.

Conclusion

After completing the final design and ensuring that it works logically and mathematically, a sufficient round of initial testing to assess all critical functionality of the device has been planned. Once the results for these tests are obtained, further tests will be planned, and this will increase in comprehensiveness and fidelity to ensure that the device is in working order and ready for design day.