

# ReGenBot: Design of An Autonomous Robot to Revitalize Burnt Soil in Southern American Forests

Alina Santander Vinokurova, Mechatronics Engineering<sup>1</sup>, Tatiana Jaimes, Mechatronics Engineering<sup>1</sup>, Cristian Sorto, Mechatronics Engineering<sup>1</sup>

<sup>1</sup>Vaughn College of Aeronautics and Technology, NY, USA

alina.santandervinokurova@vaughn.edu, tatiana.jaimes@vaughn.edu, cristian.sorto@vaughn.edu

**Abstract** – *This paper displays the design process and features of the ReGenBot. ReGenBot is a low-cost autonomous robot intended to collect burnt soil data to analyze its characteristics, especially deficiencies, and distribute fertilizer or nitrogen. This balances the soil's composition and revitalizes the terrain, according to the level program set by the user, and subsequently prepares the environment for reforestation. It is targeted at South American forests that record a high number of wildfires every year. The robot is programmed with Arduino and includes an ultrasonic sensor, a servomotor, NPK, moisture, and temperature sensor, 4416 Adafruit encoder motors, and a solar panel to power the system.*

**Keywords** – *Arduino, forest, fire, revitalization, path avoidance*

## I. INTRODUCTION

Forests' survival rate has become a high-priority issue as, over the last 50 years, Earth has lost almost one-third of its forests due to growing environmental problems such as deforestation and wildfires. This is especially critical in South America as forests cover approximately 22 percent of the continent which is equivalent to about 27 percent of the world's global forest coverage [1]. As of 2021, South America recorded approximately 325 thousand wildfires. Following these wildfires, secondary succession, known as the natural recovery and reestablishment of the environment, could take decades to come into effect, while manual revitalization requires high manual labor and funds.

Nevertheless, according to research conducted at the National University of Colombia, published through the Science Advances Journal in 2021, 48 percent of Latin American ecosystems that burned in 2003, were wiped out in the following years [2]. This is particularly worrying as wildfires in the Amazonian Forest in Brazil have been the worst in a decade, and currently, approximately 40 percent of the Brazilian Amazon is nearing the tipping point of turning into a savannah, urging a major scale intervention to slow down the dying pace of these ecosystems [3]. Wildfires pose a threat to the future of forests worldwide; although, they bring positive and negative effects. On one hand, it causes a tremendous affectation on the living organisms and decaying vegetation in the forest's environment as it simultaneously releases all the soil's nitrogen, sulfur, phosphorus, and carbon, leaving post-fire soil enriched with nutrients. However, due to extreme temperatures, most of the nitrogen, fundamental

for plants' cellular structure is vaporized, leaving the soil nitrogen-deficient [4]. This affects forests as it directly depends on the prior nitrogen composition of the soil, as well as other soil properties. As a result, the current state of the ecosystem must be analyzed to measure the nutrients in the post-fire soil to begin the revitalization process.

## II. NEED STATEMENT

There is a lack of agricultural robots targeted to aid in forest revitalization after an ecosystem has been devastated by fire. Automated processes that collect soil samples to inform the farmers of its deficiencies and its current state along with the distribution of fertilizer, can cut down costs and natural regeneration by approximately half, which could, subsequently, decrease the probability of the ecosystem being wiped out years after the wildfire. As a result, the main purpose of ReGenBot, is to act immediately after a forest fire to collect data on the soil's characteristics (i.e., nitrogen, phosphorous, potassium composition, moisture, and temperature) to determine how feasible replantation is, based on the ideal NPK ratio of 4:2:1 for plant growth [5]. If the soil presents a high deficiency, the robot distributes what is needed to prepare the soil for replantation.

Excessive trampling of burned areas should be avoided to prevent erosion. Therefore, a robot that can do the work of several humans and apply less pressure to fewer places on the burned area will help avoid more damage to the terrain.

### A. Societal Impact

The objective of the project is to increase the survival rate of forests after being affected by wildfires by preparing the soil or land for the regrowth process. Ensuring the continuity of the environment is essential to re-establish the habitat of displaced organisms, decelerate climate change in the countries affected, and restore local communities' source of income and shelter.

### B. Environmental Impact

The goal of ReGenBot is to help the environment rapidly recover after wildfires by preparing the land for its regrowth and recolonization process. Considering that the robot is purposely designed to have a positive impact on the environment, its main source of energy is solar panels, located at the top of the structure.

The robot's parts can be assembled and disassembled with basic tools, so any mechanical or electronic component can be easily replaced, extending ReGenBot's operation life.

**Digital Object Identifier (DOI):**

<http://dx.doi.org/10.18687/LACCEI2022.1.1.475>

ISBN: 978-628-95207-0-5 ISSN: 2414-6390

## II. MARKET RESEARCH

The cost of similar agricultural bots in the market ranges from \$700 to \$6000. These robots do not have all the sensors and actuators needed to make an analysis of burnt forest soil. RegenBot offers a specific solution for the post-fire regeneration risks of the forest, giving the user two options: soil sample study and soil revitalization aid. Considering the target area and the characteristics of healthy terrain, RegenBot is created to satisfy the initial stage of soil recovery having an approximate total cost of USD 650.

## III. AUDIENCE

ReGenBot is targeted at South American forests that suffered from fires, such as the Brazilian Amazon and the Bolivian Chiquitania. Governments, communities, and organizations from these regions can benefit from the functions of this bot.

## IV. ENGINEERING REQUIREMENTS

ReGenBot must meet the following requirements to meet its purpose as a revitalization agricultural robot:

**Compact:** The bot is intended to fit in a 50x50x50cm cube, and the center of gravity must be calculated so that the bot maintains stability during the ride.

**Avoid obstacles:** The bot will use an ultrasonic sensor to avoid obstacles that it will not be able to go through.

**All-terrain:** The bot will have tank wheels that will allow it to go through all other terrain obstacles.

**Solar-powered:** The bot will have a solar panel that will power the electrical system.

## V. BUDGET

The components of ReGenBot are listed in Table 1.

TABLE I  
COMPONENT PRICE LIST

Part	Unit cost	Subtotal USD
2 x Large Breadboard	5.00	10.00
1 x Arduino Mega	37.95	37.95
1 x Soil Moisture Sensor	5.88	5.88
1 x Waterproof 1-Wire DS18B20	9.95	9.95
1 x Soil NPK PH Sensor	175.78	175.78
1 x HC-SR04 ultrasonic sensor	7.09	7.09
4 x 4416 Adafruit Industries LLC	13.50	54.00
Motor w/ Encoder Outputs – 7VDC		
2 x TB67H420FTG Dual / Single	9.95	19.90
Motor Driver Carrier		
1 x MAX485 TTL to RS-485 Interface	1.99	1.99
Module		
Multipurpose 6061 Aluminum Sheet	60.96	60.96
1/8" Thick, 12"x24"		
Multipurpose 6061 Aluminum Sheet	62.75	62.75
1/8" Thick, 6"x48"		
4x 1kg of PLA	22.99	91.96
<b>Total</b>		<b>538.21</b>

The robot has a net worth of USD 650, considering the components price listed below, the drivetrain assembly (including gears, chains, and sprockets), and manufacturing prices.

## VI. DESIGN CONCEPT

ReGenBot is a two-level robot that functions as a miniature laboratory and a distributor throughout the land. In level one, the robot is a mobile laboratory that collects data to run soil tests. This informs the user about the soil's nutrient percentage and composition, moisture, and temperature. The robot's collected data is displayed on graphs that relate the location of the sample to the soil's collected characteristics. The user then interprets and analyzes the data to consider what the soil needs beforehand to be ready for replantation. Based on the findings, the user adds fertilizer or another treatment at the top of the robot, its storage compartment, and starts level two. During this period, the desired treatment will be distributed according to the previous places from which the soil samples were taken.



Fig. 1 Isometric view of ReGenBot.

### A. Mechanical

#### 1) Wheels

An essential aspect of the mechanical design is the locomotion of the ReGenBot since the terrain is expected to be rough and uneven. To address this, a tank drivetrain is the selected design to have more flexibility with the varying topography of a forest after a fire. The drivetrain includes tank tracks powered by a motor with sprockets and a shock absorption system of roller wheels.

The tank treads consist of ANSI 40 roller chain links with two flanges to secure the pads of the tank track on each connector link. The tank chain has both connecting and adding links which are stainless steel, single-strand, and have a pitch of ½ an inch. The qualities of stainless steel ensure the links will be corrosion resistant, which is necessary as the ReGenBot is expected to encounter humidity. The pads attached to the connecting links have a ribbed pattern for better grip across the terrain as the bot encounters minor obstacles not registered by the path avoidance system.

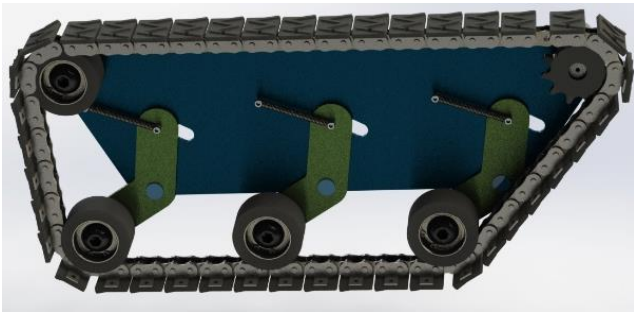


Fig. 2 Side view of tank track system

The chain-based track is being driven by a sprocket along with a motor equipped with an encoder to measure and update the distance traveled by each track. It is a brushed DC motor with a 1:20 ratio and a 3mm shaft diameter. Because of the size difference between the 3mm diameter of the encoder motor shaft and the 12.7mm diameter socket of the steel roller chain sprocket, a coupler is used to transmit the rotational power to the sprocket. The coupler is a 3mm-to-3mm connector; however, the outside radius is the correct size to fit in the sprocket and be secured using two set screws. The outer diameter of the sprocket is 5.08cm.

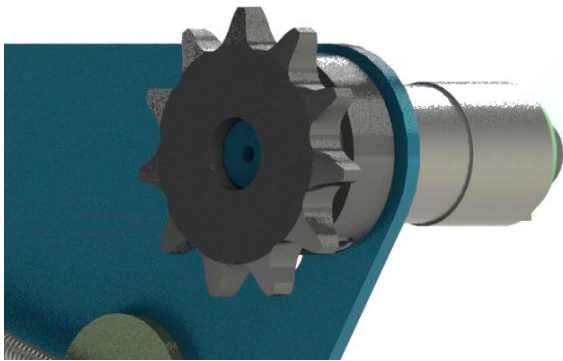


Fig. 3 Encoded DC geared motor with sprocket

Aside from the sprocket, the tension is kept in the tank treads using four threaded neoprene idler roller wheels. The neoprene rubber properties are also important for their resistance to flames and weather, which are some expected conditions during the ReGenBot's operation.



Fig. 4 Neoprene idler rollers of tank tracks

Each of the bottom three threaded rollers on each side is secured to a shock absorption system. This system consists of the roller wheels being connected to a lever arm, which pivots at the middle fulcrum on the side panel of the main body, and the other end is connected to a spring anchored on

the side panel. This system is included to minimize vibrations on the main body of the system and increase stability when rolling over smaller terrain inconsistencies.

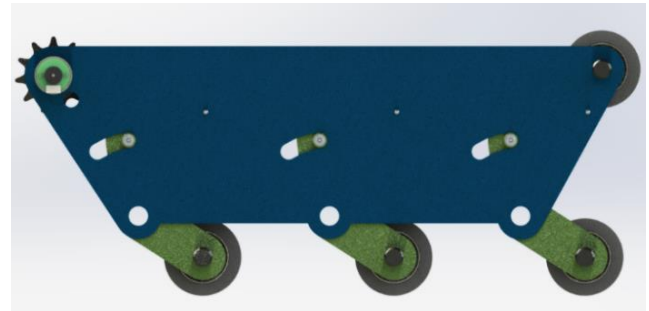


Fig. 5 Inner view of side panels with shock absorption guide slots

The sketch for the middle base design is illustrated in Fig. 6 including different subsystems needed for operation. The sketch provided visualization and faster drafting prior to beginning the CAD design, analyzing the placement of each portion to maximize space efficiency.

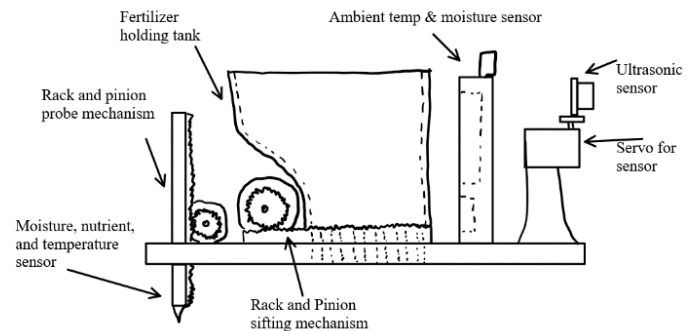


Fig. 6 Initial sketch of ReGenBot main body to estimate placement

## 2) Sensor probe

The linear motion of the sensor probe is accomplished by a rack and pinion mechanism, and the depth of the measurement is determined by the rotation of the motor. The rack and pinion are both made of brass, have a 20-degree pressure angle, and have a 3mm face width. The gear has a hole diameter of 3mm and a pitch diameter of 8.5mm which is secured to the DC motor using a set screw. The probe extends up to 20 cm from the fully lifted position. The sensors on this probe include a temperature sensor, NPK sensor, and moisture sensor, with a case meant to protect from possible breaking when inserted into the soil. (Fig. 7)

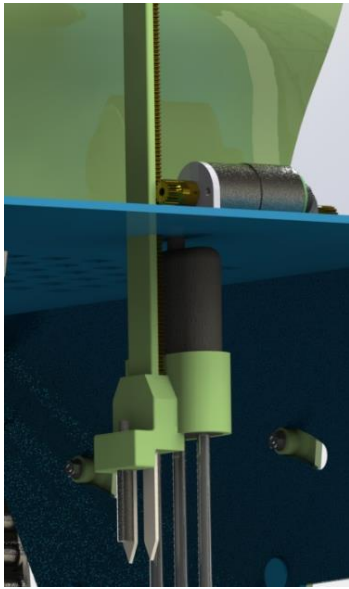


Fig. 7 Sensor probe mechanism

### 3) Electronics casing

A casing is also mounted on the base plate to protect the main controlling components from the weather or any debris from the forest fire which may cause interference or damage to the sensitive components. These components include an Arduino Mega 2560, motor drivers, and other modules for sensors. The ambient temperature and humidity sensors are mounted on the casing, so they are secured and still allow appropriate readings.

The ultrasonic sensor needed for the obstacle avoidance system required the ability to rotate the sensor about the axis normal to the floor. This is achieved by a casing design for the ultrasonic sensor attaching to an SG90 servo to create a controlled rotating motion about the axis mentioned. The ultrasound sensor-servo system can be seen in Fig 8.



Fig. 8 Ultrasonic sensor mount on SG90 servo

### 4) Fertilizer Tank

The subsystem referred to as the fertilizer tank is meant as the storage tank for any of the nutrient types which will be spread across the surveyed land. This tank system serves as storage as well as the distribution system for the nutrients, using a sieve-style interface with linear motion.

The sieve being referred to is a panel with a 6 by 11 grid of  $1\text{cm}^2$  square openings, with a matching set of openings on the base plate. When both sets of openings match up, that is when the most fertilizer is being dispensed. There is no fertilizer dispensed when the panel slides 1cm since this motion closes all the gaps in the interface.

The linear motion of the panel is achieved through the rack and pinion mechanism. The brass pinion gear is attached to a brushed DC motor fixed to the base plate. The sliding panel slides on top of the base plate and is controlled by the brass gear rack fixed to it as seen in Fig 9. Both the rack and pinion have a 20-degree pressure angle and a 3mm face width to mesh and create linear motion.

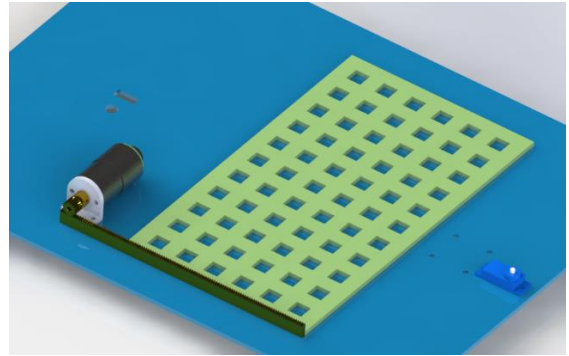


Fig. 9 Rack and pinion sieve mechanism

The material selected for the base plate and side panels is aluminum for the corrosion resistance and chemical resistance to nitrogen, phosphorus, and potassium expected for distribution. These properties are useful for the moist environment it is expected to be around.

The placement of the tank was also considered and can be seen in Fig 10. The storage tank is set in this location to set the center of gravity as near to the center as possible, also considering some tilting from uneven terrain. Keeping this center of gravity as close to the middle and the ground prevents the ReGenBot from possible tipping.

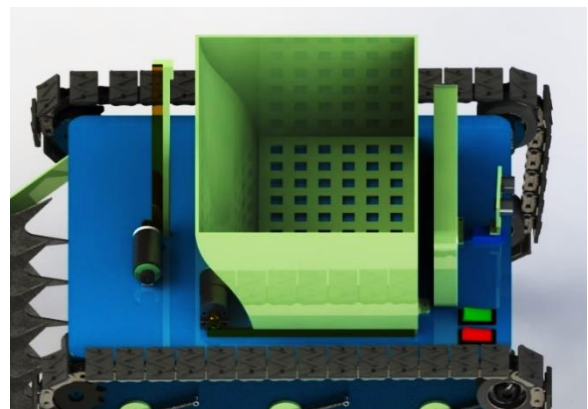


Fig 10 View of tank on the base plate with view of the sieve mechanism

### 5) Solar Panel Placement

The solar panel is located on top of the lid for the fertilizer tank to have the most exposure to the sun during operation. Since the deforestation caused by wildfires clears



the trees creating shade, the solar panel receives more sunlight than if it were to travel in a forest with healthy foliage.

#### 6) Spike lawn aerator

The spike lawn aerator is a rolling mechanism with spikes pulled behind the drivetrain of the ReGenBot. Its function is to create holes and fluff up the soil to improve the circulation of water and air through the substrate and facilitate the development of new roots. It also enhances rainwater infiltration into the seedling environment. The system is seen in Fig 11.



Fig. 11: Aerator model on ReGenBot

### B. Electrical

ReGenBot consists of a moisture, temperature, and NPK (Nitrogen, Phosphorous, Potassium) sensor, along with an ultrasonic sensor, a servo motor, and 4 encoded motors. The circuit is controlled by an Arduino Mega 2560 due to its capacity and number of digital and analog pins. (Fig. 12)

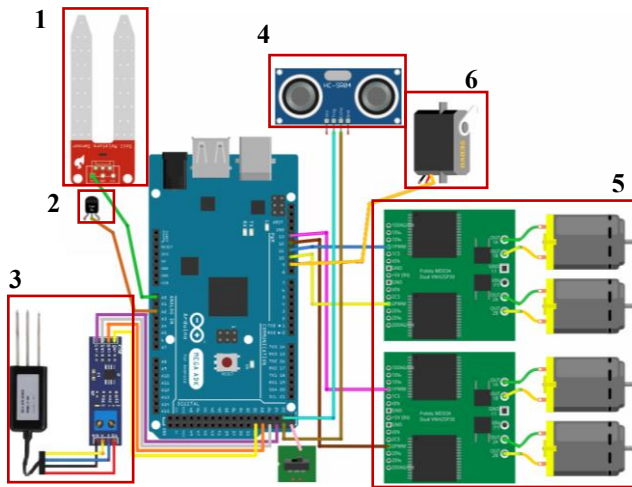


Fig. 12: ReGenBot Electrical Overview

#### 1) Moisture

The Soil Moisture Sensor is connected as seen in Fig 12.1, using analog pin 1 and digital pin 7. It measures the volumetric content of water when inserted in the targeted terrain. The sensor connects through an electronic module to

the Arduino. This module produces an output voltage corresponding to the resistance of the sensor. The moisture is measured in percentages, so when the sensor is connected in analog mode giving a value from 0 to 1023. It is needed to map these values from 0 to 100 with the formulas shown below.

$$\text{AnalogOutput} = \frac{ACvalue}{1023} \quad (1)$$

$$\text{Moisture \%} = 100 - (\text{AnalogOutput} * 100) \quad (2)$$

#### 2) Temperature

The waterproof version of the DS18B20 is connected as seen in Fig. 12.2, using analog pin 0. The sensor measures the temperature from -55°C to 125°C with an accuracy of  $\pm 5$ , and its data is saved into the temperature text file to export to Excel and develop a graph to analyze the data.

#### 3) NPK

The NPK sensor is connected as seen in Fig. 12.3, using digital pins. It detects the content of nitrogen, phosphorus, and potassium in the soil, through different inquiry frames, directly associating the type of fertilizer needed to balance the nutrient-deficient compositions. A Modbus Module like RS485/MAX485 is needed to connect the sensor to the Arduino MEGA and interface its module. The sensor operates on 9-24V, and its accuracy is up to within 2 percent.

#### 4) Ultrasonic Sensor

HC-SR04 ultrasonic sensor is connected as seen in Fig. 12.4. It allows the robot to avoid obstacles. The 5V and ground pins are respectively connected. The Trig and Echo pins are connected to digital pins 2 and 3, respectively.

#### 5) Encoded motors

ReGenBot has four DC encoded motors for the wheels, 4416 Adafruit Industries LLC DC Motor Gearmotor, with a 1:20 gear ratio and 12 pulses, or ticks, per revolution. It drives a maximum input of seven DC volts throughout the system, while its pulse width modulation (PWM) varies depending on the desired speed for each motor.

These are rotary encoders mounted to a motor that read the pulses per unit that the motor takes in a determined time. This provides closed-loop feedback signals of the motor for speed verification and position tracking. ReGenBot's motor encoders are used to track and assign positioning. For this, the conversion of ticks per meter is calculated, representing the rate of change, to then input the desired distance for the motors to travel, resulting in the number of ticks (pulses) it takes to reach the distance.

The Arduino Mega communicates and controls the motors through the TB67H420FTG Motor Driver Carrier, capable of controlling up to two motors at a time. (Fig. 12.5) It actively limits the current through the motors by using a fixed frequency PWM current regulation (also known as current chopping) [6].

### 6) Servo motor

One micro servo motor is used to slightly turn the ultrasonic sensor from right to left when the robot detects an obstacle in front of itself. This component acts as the mounting point for the ultrasonic sensor that continuously runs the path avoidance code throughout both levels of the robot. Its connection to the Arduino is seen in Fig. 12.6.

### 7) Solar power

ReGenBot uses solar panels to provide the necessary power for all subsystems. A lithium-ion rechargeable battery that provides a 5V source from a single cell is used. (Fig. 13)

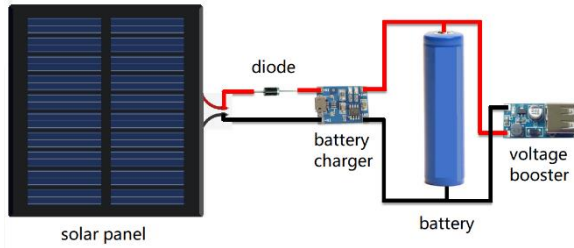


Fig. 13 Circuit for solar panel

### C. Coding

ReGenBot's code is developed on Arduino's IDE software. The development and execution of this program are divided into six parts: terrain and program level input, sensor data collection, obstacle avoidance, path correction, and navigation. The integration of these subsystems is displayed in the flowchart in Fig. 14 and Fig. 15 and discussed in further detail in the following subsections.

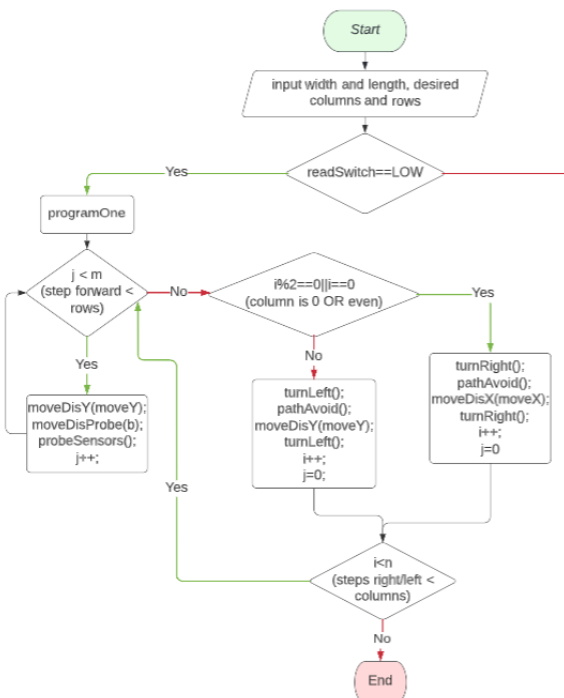


Fig. 14 Overview of the system input and programOne flowchart

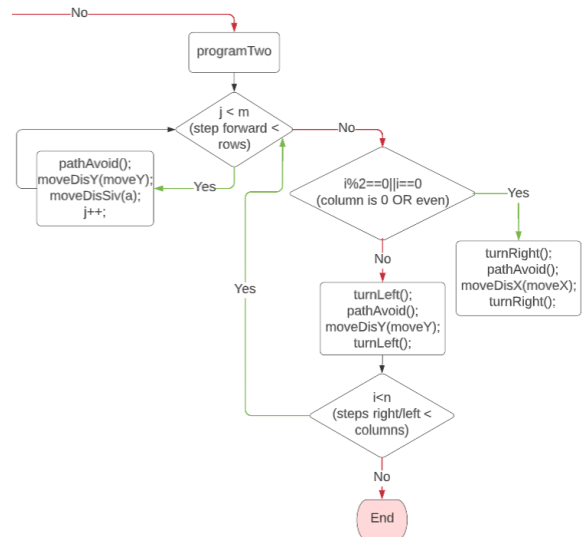


Fig. 15: Overview of programTwo flowchart

### 1) Terrain input

As soon as ReGenBot's program is initialized, the robot asks the user to input the width and the length of their terrain in meters. Once this information is saved in the robot's memory, it asks the user to input the number of rows and columns they wish to map out the terrain and mark collection points at the intersections of the rows and columns. This user input is utilized in level one and level two of the program as it sets where the robot stops to collect soil samples and distribute fertilizer, respectively. The serial function parseInt (Fig. 16) is implemented to read the integer the user inputs at the beginning and store it in the respective variables. Since this is executed independently of the level selected for the robot, it is coded outside of the program switch cases.

```
Serial.println("Width of the terrain in meters: "); //prompt for input
while (Serial.available() == 0) {} //wait for input
x = Serial.parseInt(); //read user input and hold in x
Serial.println("Length of the terrain in meters: "); //prompt for input
while (Serial.available() == 0) {} //wait for input
y = Serial.parseInt();
Serial.println("Desired columns: "); //prompt for input
while (Serial.available() == 0) {} //wait for input
n = Serial.parseInt();
Serial.println("Desired rows: "); //prompt for input
while (Serial.available() == 0) {} //wait for input
m = Serial.parseInt();
```

Fig. 16 Terrain input code display

### 2) Program level input

ReGenBot is composed of two levels. This is initially recorded according to the input on the slide switch. If the switch is at a low state, to the left, it executes program one, the testing phase. Else if the switch is to the right, it executes program two, the distribution phase. (Fig 17)

```

int readSwitch = 0;
readSwitch = digitalRead(switchL);
if(readSwitch == LOW)
{
    programOne();
}
else
{
    programTwo();
}

```

Fig. 17 Read switch input

The first level is targeted to collect soil samples throughout the intersections of the rows and columns desired by the user. For this, void function programOne is called and executed. (Fig. 18) In this function, the robot moves in the positive Y direction until the number of rows is met. After this, the robot turns right or left depending on the column number. If the column is zero or even, it will turn right while if the column is odd, it will turn left. After the robot reaches each row point throughout the column, it executes the probeSensors function that lowers all the sensors into the soil, records it into their respective text files, and saves it to the SD card.

```

void programOne() //testing
{
    int i = 0;
    int j = 0;
    /*----MOVING THROUGH THE TERRAIN-----*/
    while (j<m)
    {
        pathAvoid();
        moveDisY(moveY);
        j++;
        moveDisProbe(b);
        probeSensors();
    }
    if((i%2==0||i==0))
    {
        turnRight();
        pathAvoid();
        moveDisX(moveX);
        turnRight();
        i++; j=0;
    }
    else if((i%2!=0))
    {
        turnLeft();
        pathAvoid();
        moveDisX(moveX);
        turnLeft();
        i++; j=0;
    }
    if (i<n)
    {
        Stop();
    }
    delay(500);
    /*-----MOVING THROUGH TERRAIN END -----*/
}

```

Fig. 18 programOne function

Level two (Fig. 19) is targeted to distribute fertilizer or treatments that balance out the composition deficiencies in the soil. Similar to the code syntax followed for level one, the robot moves along the positive Y direction and trans left or right depending on the number of the column it is at. Once it reaches a row intersection, the robot stops and drops the fertilizer, or treatment, by executing the moveDisSiv function that has a nested while loop to control the sieve motor at the top base of the robot. This causes the robot to move the sieve from a locked position to an unlocked one, traveling about 1cm. Once it is unlocked, it travels the same distance in reverse to lock.

```

void programTwo() //distribution
{
    int i = 0;
    int j = 0;
    /*----MOVING THROUGH THE TERRAIN-----*/
    while (j<m)
    {
        pathAvoid();
        moveDisY(moveY);
        j++;
        moveDisSiv(a);
    }
    if((i%2==0||i==0))
    {
        pathAvoid();
        turnRight();
        moveDisX(moveX);
        turnRight();
        i++; j=0;
    }
    else if((i%2!=0))
    {
        pathAvoid();
        turnLeft();
        moveDisX(moveX);
        turnLeft();
        i++; j=0;
    }
    if (i<n)
    {
        Stop();
    }
    delay(500);
    /*-----MOVING THROUGH TERRAIN END -----*/
}

```

Fig. 19 programTwo function

### 3) Sensor data collection

Sensors are activated for level one of ReGenBot's system. the information for the temperature (Fig. 20), NPK, moisture, and pH (Fig. 21) data collection it's grouped up into the probeSensors function. Within this function, each criterion is read according to the sensor and saved in an individual file in the SD card. This allows the user to display the data in a graph for soil analysis. Similar to the movement the sieve performs, the moveDisProbe powers an encoded motor that inserts the sensors about six cm into the soil. (Fig. 22)

```

void probeSensors()
{
    //temperature
    regenbot = SD.open("temp.txt", FILE_WRITE);
    now = rtc.now();
    regenbot.print(now.hour());
    regenbot.print(":");
    regenbot.print(now.minute());
    sensors.requestTemperatures();
    Celcius=sensors.getTempCByIndex(0);
    Fahrenheit=sensors.toFahrenheit(Celcius);
    regenbot.print(" C ");
    regenbot.print(Celcius);
    regenbot.print(" F ");
    regenbot.println(Fahrenheit);
    delay(1000);
    regenbot.close();
}

```

Fig. 20 probeSensors function

```

//moisture
regenbot = SD.open("moist.txt", FILE_WRITE);
now = rtc.now();
regenbot.print(now.hour());
regenbot.print(":");
regenbot.print(now.minute());
output_value= analogRead(sensor_pin);
output_value = map(output_value,550,0,0,100);
regenbot.print("Moisture : ");
regenbot.print(output_value);
regenbot.println("");
delay(1000);
regenbot.close();

//npk sensor
byte nitrogen_val,phosphorus_val,potassium_val;
regenbot = SD.open("npk.txt", FILE_WRITE);
now = rtc.now();
regenbot.print(now.hour());
regenbot.print(":");
regenbot.print(now.minute());
nitrogen_val = nitrogen();
delay(250);
phosphorus_val = phosphorus();
delay(250);
potassium_val = potassium();
delay(250);
//print
regenbot.print("Nitrogen_Val: ");
regenbot.print(nitrogen_val);
regenbot.println(" mg/kg");
regenbot.print("Phosphorous_Val: ");
regenbot.print(phosphorus_val);
regenbot.println(" mg/kg");
regenbot.print("Potassium_Val: ");
regenbot.print(potassium_val);
regenbot.println(" mg/kg");
delay(2000);
regenbot.close();
}

```

Fig. 21 NPK and moisture sensor configuration for data collection

```

void moveDisProbe (int b) //testing motor
{
    //ticks per m = 763.94
    int ticks ;
    ticks = b*763.94;
    while (encA2[1]<ticks)
    {
        moveProbe();
    }
}

```

Fig. 22 moveDisProbe function

#### 4) Obstacle avoidance

The ultrasonic sensor sends out a pulse and if an object is detected within 19cm (Fig. 23) of the robot, the motors decrease their speed and change their path. Once the robot stops, the servo motor rotates the ultrasonic sensor from right to left to send signals once again and detect which way is best to go. This is done through the compare Distance function, which chooses to turn right or left depending on which side detects the furthest distance from an obstacle.

```

servo.write(80); //servo position at 80 deg
delay(100);
distance = search();

if (distance < 19) //object is close
{
    analogWrite(PWM[0],100); //decrease the speed
    analogWrite(PWM[1],100);
    changePath();
}

else if ((distance >= 19) && (distance < 60)){ //object is far
    analogWrite(PWM[0],200); //increase the speed
    analogWrite(PWM[1],200);
    forward(); //move forward
}

```

Fig. 23 Path avoidance code

#### 5) Path correction

Once the ReGenBot changes its path, it travels at an angle until another obstacle is detected in its way. For the robot to continue a straight path, a path correction function is implemented after the robot turns. (Fig. 24) For example, the function backToPathR is implemented to direct the robot that changed path and turned right, back to the original path (straight). For this, the ultrasonic sensor turns to the opposite direction its wheel turns and runs a similar syntax code to the path avoidance function, to check if the object is still in sight. The speed implemented for the path correction is low to decrease the probability of the robot overshooting. The number of steps that the robot takes is incremented while the distance to the obstacle is less than 19cm. Once the obstacle is far away, the robot turns in the opposite direction and takes the same number of steps. Once the number of steps is equivalent, it turns in the opposite direction one more time to continue along the column.



```

void backToPathR() {
    int numberSteps = 0;
    int i = 0;
    while (distance < 19) //object is close
    {
        analogWrite(PWM[0],100); //decrease the speed
        analogWrite(PWM[1],100);
        numberSteps++;
    }
    turnLeft();
    for (i=0; i<numberSteps; i++)
    {
        analogWrite(PWM[0],100); //decrease the speed
        analogWrite(PWM[1],100);
    }
    turnRight();
}

void backToPathL() {
    int numberSteps = 0;
    int i = 0;
    while (distance < 19) //object is close
    {
        analogWrite(PWM[0],100); //decrease the speed
        analogWrite(PWM[1],100);
        numberSteps++;
    }
    turnRight();
    for (i=0; i<numberSteps; i++)
    {
        analogWrite(PWM[0],100); //decrease the speed
        analogWrite(PWM[1],100);
    }
    turnLeft();
}

```

Fig. 24 Path correction code

## VII. PRODUCT EVALUATION

The current state of development for the ReGenBot is to receive funding and begin the production of a physical prototype. After the final assembly of the prototype, tests will be conducted in New York forests of different terrains to assess the readings and mobility of the robot to make necessary changes or adjustments. Then the ReGenBot will be sent with the team to Bolivia and Colombia to test the performance of the soil nutrient data collection with possible collaboration with Armonia, a Bolivian conservation non-governmental organization.

## VIII. CONCLUSION

ReGenBot is a robot design that focuses on soil revitalization and data collection of burned forests. It decreases labor cost and time by more than half, and its low cost compared to the market makes it affordable for organizations and communities in South America. The user-robot interaction is simple and straightforward, making the robot accessible to many individuals in the audience.

ReGenBot reduces human interaction with the delicate and dangerous state of forests while incrementing its survival rate.

For future work, new features can be implemented to prepare ReGenBot to be a multipurpose agricultural robot. Artificial Intelligence could be developed to make decisions based on the collected data, where the device could scan the terrain and display a map of the surroundings, identifying the amount of burned wood, the color of the soil, and signs of wildlife. With time, ReGenBot can be repurposed to conduct the replantation and regrowth process by spreading seeds and water while traveling through the area.

## ACKNOWLEDGMENT

The authors would like to thank the President of Vaughn College of Aeronautics and Technology, Dr. Sharon DeVivo, along with the Engineering and Technology Department Chair, Dr. Hossein Rahemi, the Department of Education federal grant (Title III, Part F, HSI-STEM, and Articulation grant) which provided necessary funding support to engage Vaughn's students in STEM-related scholarly and professional activities. and faculty professor Dr. Miguel Bustamante for their continued support throughout the whole project. Also, a special thank you to Professor Donald Jimmo for his continuous support in putting this project together.

## REFERENCES

- [1] Butler, R. A. "Rainforest Information." *Mongabay*. August 2020. [Online]. Available: [https://rainforests.mongabay.com/deforestation/archive/Total\\_South\\_America.htm](https://rainforests.mongabay.com/deforestation/archive/Total_South_America.htm). [Accessed: January 21, 2022].
- [2] Arenteras, D., Dávalos, L. M., Barreto J. S., Miranda A., Retana J., et al. "Fire-induced loss of the world's most biodiverse forests in Latin America." *ScienceAdvances*. DOI: 10.1126/sciadv.abd3357. August 13, 2021. [Online.] Available: <https://www.science.org/doi/10.1126/sciadv.abd3357>. [Accessed: February 7, 2022].
- [3] Harvey, F. "Amazon near tipping point of switching from rainforest to savannah – study." *The Guardian*. October 5, 2020. [Online]. Available: <https://www.theguardian.com/environment/2020/oct/05/amazon-near-tipping-point-of-switching-from-rainforest-to-savannah-study>. [Accessed: February 9, 2022].
- [4] Minas. S., "How Do Wildfires Affect Soil?" *AES, C.M. APPLIED EARTH SCIENCES*. November 12, 2019. [Online]. Available: <https://www.aessoil.com/how-do-wildfires-affect-so>. [Accessed: February 9, 2022].
- [5] Naresh. R., "What Are the Ideal Levels of NPK in Soil in Ppm?" *ResearchGate*. March 10, 2022. [Online]. Available: <https://www.researchgate.net/post/What-are-the-ideal-levels-of-NPK-in-soil-in-ppm>. [Accessed: May 5, 2022].
- [6] "TB67H420FTG Dual/Single Motor Driver Carrier." *Pololu*. [Online]. Available: <https://www.pololu.com/product/2999>