UNIVERSITY OF Cincinnati

Plant Watering Robot

Joel Ryan, Sudarshan Pillai, Justin Boamah

EECE5128C – Intelligent Machine Design

Class section: 001

Semester: Fall Semester 2020

Course Instructor: Zachariah Fuchs

Table of Contents

Introduction

The goal of the plant watering robot was to follow a line to a potted plant, actuate a sensor into the dirt to check the moisture level of the dirt, water the plant if needed, and continue on to the next plant when the moisture threshold is reached. The lines the robot follows is laid out by the user to lead the robot to their plants. Once the robot reaches the plants, a bump sensor would be triggered to cause actuation of a servo motor with a moisture sensor on it. The LCD displays the current moisture level, and the goal moisture level. At this point a pump would water the plant until the goal moisture level is reached. In our case, the pump wasn't present due to concerns about water being near our only prototype, so it was instead represented by a green LED turning on and off. In the case that the user thinks the plant has enough water, but the robot keeps watering, there is a manual override button that will force the robot to move onto the next plant. After the goal moisture is reached (or the sensor is overridden by the user), the robot will turn to the right and find the path to the next plant to repeat the watering process.

If at any time the robot senses anything too close to its proximity sensors, the robot will stop the wheels from turning and hold its position until the obstacle is removed. Since the robot needs to be able to run into the potted plants to begin the soil measuring routine, the proximity sensors are above the height of the pots. The robot is still able to detect obstacles using the bump sensor on the front which will cause the wheels to stop spinning, and a buzzer to go off and alert the user.

This paper will go on to outline the organization of the system, the actuators this system is using to complete its task, unique behaviors the robot will be using, and the sensors the robot uses to acquire input from the real world.

Integrated System

The system will primarily consist of an Arduino Uno to do all the processing, and 2 separate batteries to power the sensors and processor. One battery will be a 9 volt connected directly to the Arduino, and the second battery will be a 5 volt mobile USB charger. All our sensors will communicate with the arduino and be powered by the external battery. The power and sensor layout is shown in figure 1 below.

Pin Number	Purpose	IO Configuration
A0	Left IR Sensor	Analog Input
A1	Water Filling LED	Digital Out
A2	Right IR Sensor	Analog Input
A3	Water Moisture Sensor	Analog Input
0	H Bridge A1	Digital Output
1	H Bridge A2	Digital Output
2	H Bridge B1	Digital Output
3	Bump Sensor	Digital Inputs
4	Beeper	Digital Output
5	PWM A	PWM
6	PWM B	PWM
7	H Bridge B2	Digital Output
8	Line Follow Leftmost	Digital Input
9	Line Follow Left Center	Digital Input
10	Line Follow Right Center	Digital Input
11	Servo Control	PWM
12	Line Follow Rightmost	Digital Input
13	Pushbutton	Digital Input
15	I2C	Serial Communication
16	I2C	Serial Communication

Figure 1: Pin Assignments

There are 3 PWM signals, 3 analog inputs, 6 digital outputs, and 6 digital inputs. The PWM signals are for the left and right motors, and the servo arm that actuates the sensors. There are 4 digital inputs for the line following sensors, one for the user override button, and one more for the bump sensor. There are 2 digital outputs for each motor to control direction, one for the LED that signifies the pump, and 1 for the obstacle hit buzzer. All of the pin assignments are shown in figure 2.

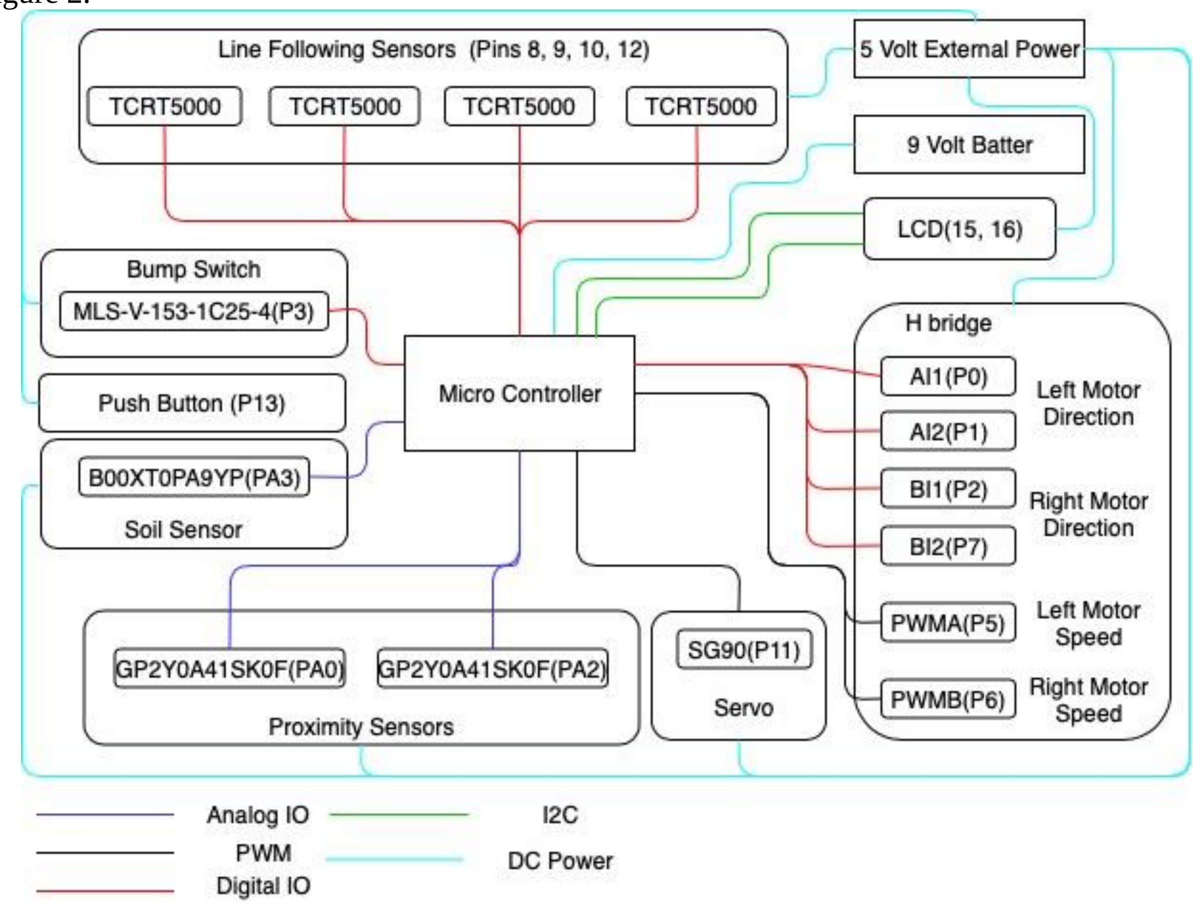

Figure 2: Functional Diagram

The general layout of the code will use modes to run different functions inside a timer interrupt every 20ms. Switching between these modes will also determine what is being displayed on the LCD. There are 4 modes. Line follow, obstacle detect, soil measure, and find next line. By default, the robot is in the line follow mode. It will stay in this mode until a bump switch is triggered, or something is too close to the proximity switch. If something is too close or the bump switch wasn't triggered by a plant, the robot goes into obstacle detect mode. If the bump switch was triggered by a plant, the robot goes into the soil measure routine. When this is done, the plant goes into the find next line. Once the next line is found the robot will line follow and repeat the above process again.

Actuation

Wheels

There will be 2 wheels for our robot to be able to move. They will use an H-Bridge to control them. An H-bridge is an integrated circuit that controls the polarity of a PWM signal. This allows us to control the direction of the motor and the speed of the motor easily. The H-Bridge uses 3 inputs for each wheel. There will be 2 for direction control and a 3rd PWM signal for speed control. Speed and direction of the motor are very important in multiple portions of the robots operation. More specifically the motor speed is an integral part of line following. The direction can also be set to a stopped state which is important for detecting obstacles and the plants. The reverse direction and speed also allow us to make the sharp turn that is required for turning and finding the next line once the the soil measuring procedure is completed. The H-Bridge pinout is shown in figure 3.

+5V	VM	PWMA	Pin 5
+5V	VCC	A12	Pin 1
GND	GND	AI1	Pin 0
Motor 1 +	A01	STBY	Not Used
Motor 1 -	A02	BI1	Pin 2
Motor 2 +	B02	BI2	Pin 7
motor 2 -	B01	PWMB	Pin 6
GND	GND	GND	GND

Figure 3: H-Bridge Pinout

The H-Bridge used for this robot was the TB6612FNG. Each motor would have a normal current draw of 1.2 mA and up to 3.2 mA at peak. The peak draw would be when turning on or switching directions.

<u>Soil Actuator</u>

To measure the soil, the sensor needed to be out of the way as the robot approached the plant. To do this, the sensor was placed on an extended servo arm that was 3D printed. The servo also needed to be placed above the bump sensor so it didn't obstruct the method of finding the plant. This servo and bump sensor holder was also 3D printed. The model and physical layout is shown in figure 4.

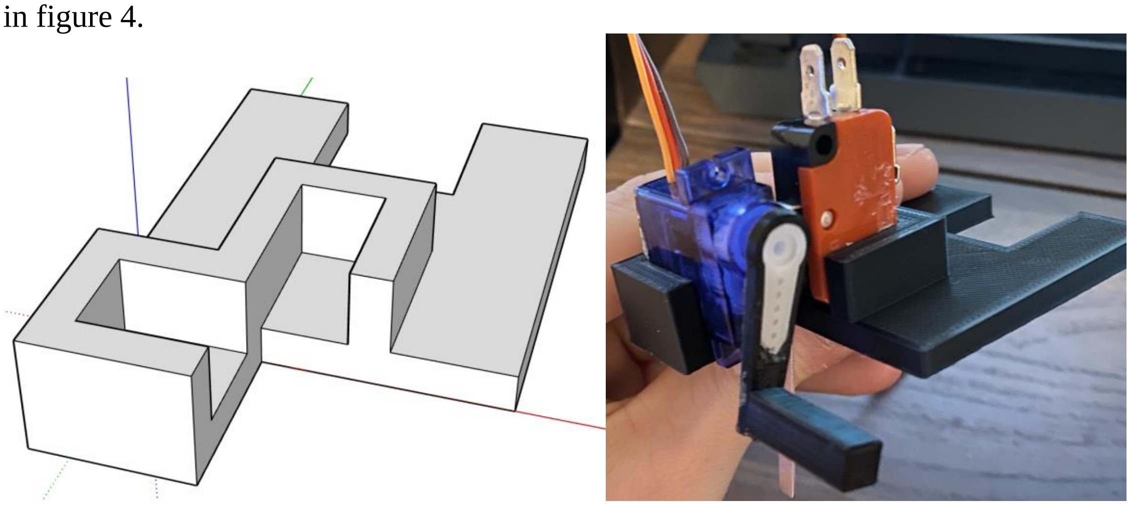

Figure 4: Servo and Bump Sensor Holder

This arm would be in a lifted state until the robot detected the plant in front of it. At this point it would actuate into the lowered position until the soil reaches the desired moisture level or is manually overridden. The arm gets raised back up and the robot continues to the next plant. The servo only requires. The servo in this project was an SG90. It required 5V, ground, and a PWM signal. The duty cycle of the PWM signal determined the position of the servo. At 0 ms the position was 0 degrees, -90 degrees at 1ms, and 90 degrees at 2 ms. With a torque rating of 2.5 kilograms per centimeter, there was no problem actuating into the dirt.

Sensors

Proximity Sensor

The system will use a single proximity sensor on each side of the chassis for obstacle avoidance. These will be GP2Y0A41SK0F proximity sensors that use IR to detect distance. This will be relayed back through as an analog signal with a voltage between 0 and 5. One limit of these proximity sensors is their range is limited between 4 and 30 centimeters. Within 4 centimeters, the reading will start to increase again. The selected proximity switch is pictured below.

Figure 5: Proximity Sensor (GP2Y0A41SK0F)

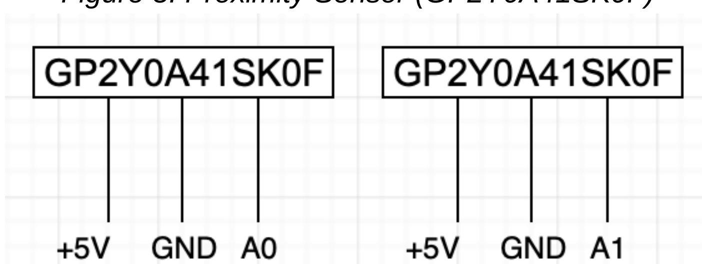

Figure 6: Proximity Sensor Wiring

Bump Switch

The chassis will use 2 bump switches on the font to detect when we have hit the plant that we are trying to sample the soil moisture for. This bump sensors use a metal plate that will hit an object in front of our robot and return a digital input to the microcontroller. The particular bump sensor we chose is MLS-V-153-1C25-4. These were chosen for their relatively low cost. These bump sensors will tell us if we have hit an obstacle or if we have reached the plant that needs watered. We will use 2 on the front of our robot both wired into the same interrupt routine.

Figure 7: Bump Switch (MLS-V-153-1C25-4)

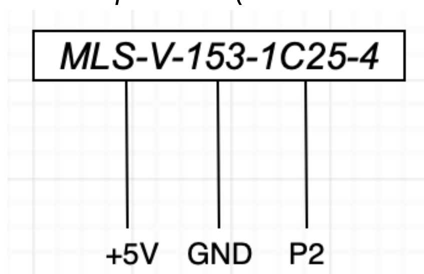

Figure 8: Bump Switch Wiring

<u>*Moisture Sensor*</u>

Our first special sensor will be a device that can measure the moisture of soil. These come with a comparator that is attached that will either return the signal as an analog voltage value, or a digital input with an adjustable threshold. We are choosing to use the analog input method because this will give us higher resolution and make our robot more helpful. With an analog value, we can tell how much water the soil needs. If we were to use the digital value, we would know that it needs water, but no idea how much. We will show the moisture level to the robot owner on an LCD display and tell if the soil needs more water or not.

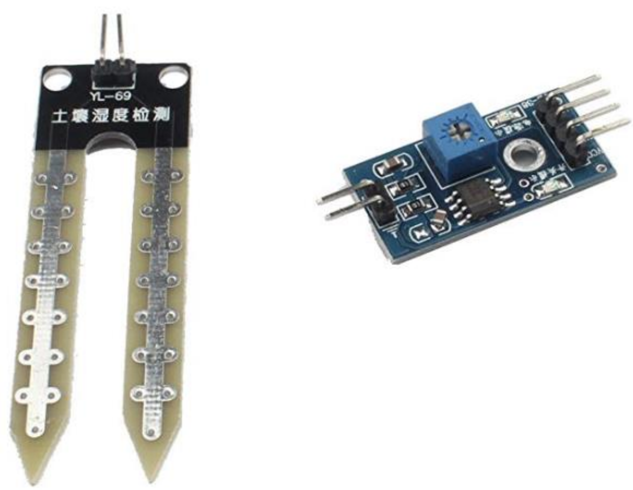

Figure 9: Soil Moisture Sensor (B00XT0PA9Y)

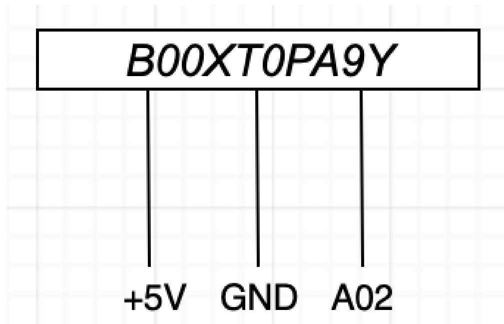

Figure 10: Soil Moisture Sensor Wiring

<u>*Line Following Sensor*</u>

Our second special sensor will be an array of IR line tracker sensors. This sensor will send out IR and return a digital input based on how much of the IR is being sent back to the sensor. These devices came with the option of receiving an analog signal or a digital signal. We chose the version that send a digital signal back, because unlike the soil moisture sensor, no one will be interfacing with this sensor. Instead, we can set the threshold on the device itself and not have to worry about setting this programmatically. The specific model we went with was the TCRT5000. We will line up 5 of these along the front of our robot in an array and use these inputs to control the speed of motors to stay on a line.

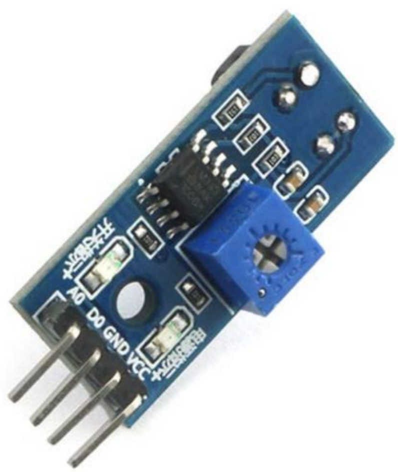

Figure 11: Line Following Sensor (TCRT5000)

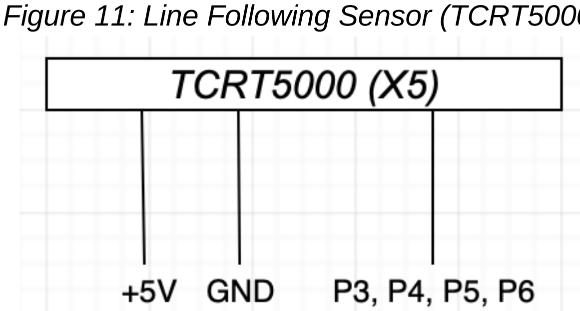

Figure 12: Line Following Sensor Wiring

Behavior

Line Following

The robot was capable of following a line and taking turns to be able to find the plant. This was done by using 4 sensors placed alongside each other in a 3D printed sensor holder that was placed on a 3D printed rail that allowed for the height to be adjusted for best performance. The sensors would detect if there was a line below them and based on which sensors were active, the speed of the 2 wheels were set. These sensors were also used for the process of knowing if the robot had reached a plant, or if the robot had hit an obstacle. A plant would have a solid rectangle of tape in front of it causing all 4 sensors to be high at the same time. If the bump sensor was triggered and all sensors were reading high, the robot would move into the moisture checking routine. If less than all 4 sensors were high then the robot would move into the obstacle detected behavior. The line following behavior is explained if figure ? below. Also worth noting is that the bump sensor was wired to an interrupt pin. The figure shows the bump sensor as a periodic thing that is being checked, but in the actual system it was constantly monitored.

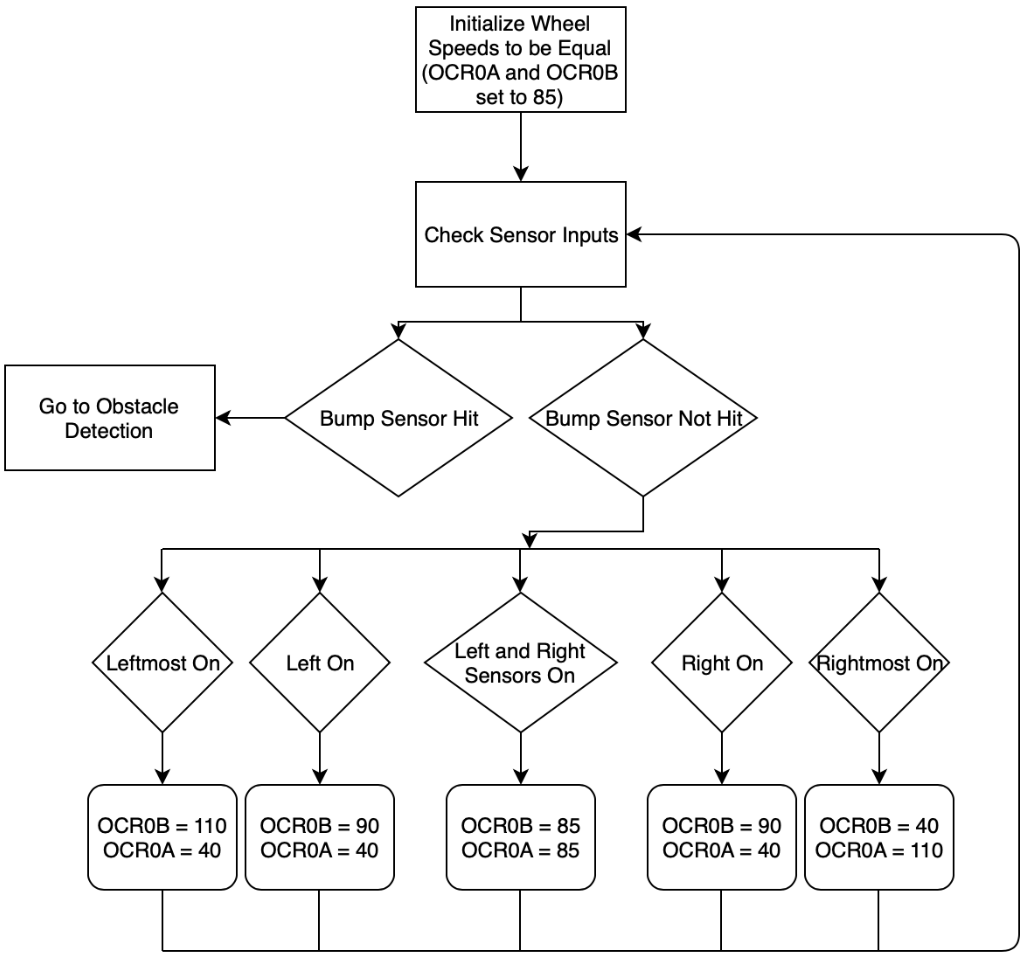

Figure 13: Line Follow Flow Chart

The line following array was planned to initially use 5 sensors. However, using 4 was enough to control our robot. It wasn't able to make sharp right turns, but adding more sensors probably wouldn't have allowed for this to happen, only decreasing the time between our timer 1 interrupt which wasn't an option for us.

Moisture Checking
Once arrived at the plant which is checked by if all the sensors were reading high, the robot will go into moisture checking routine. During this routine, the robot will lower an arm (held by servo motor) holding the moisture sensor into the soil of the plant. Once the moisture sensor is in the plant it will start to read data from the sensor. After receiving readings from the sensor, the LCD on the robot will display information as to whether or not you need to water the plant. If the soil is good on moisture then the robot moves to the find next plant behavior. If the soil needs water then the robot goes to the waiting for water behavior.

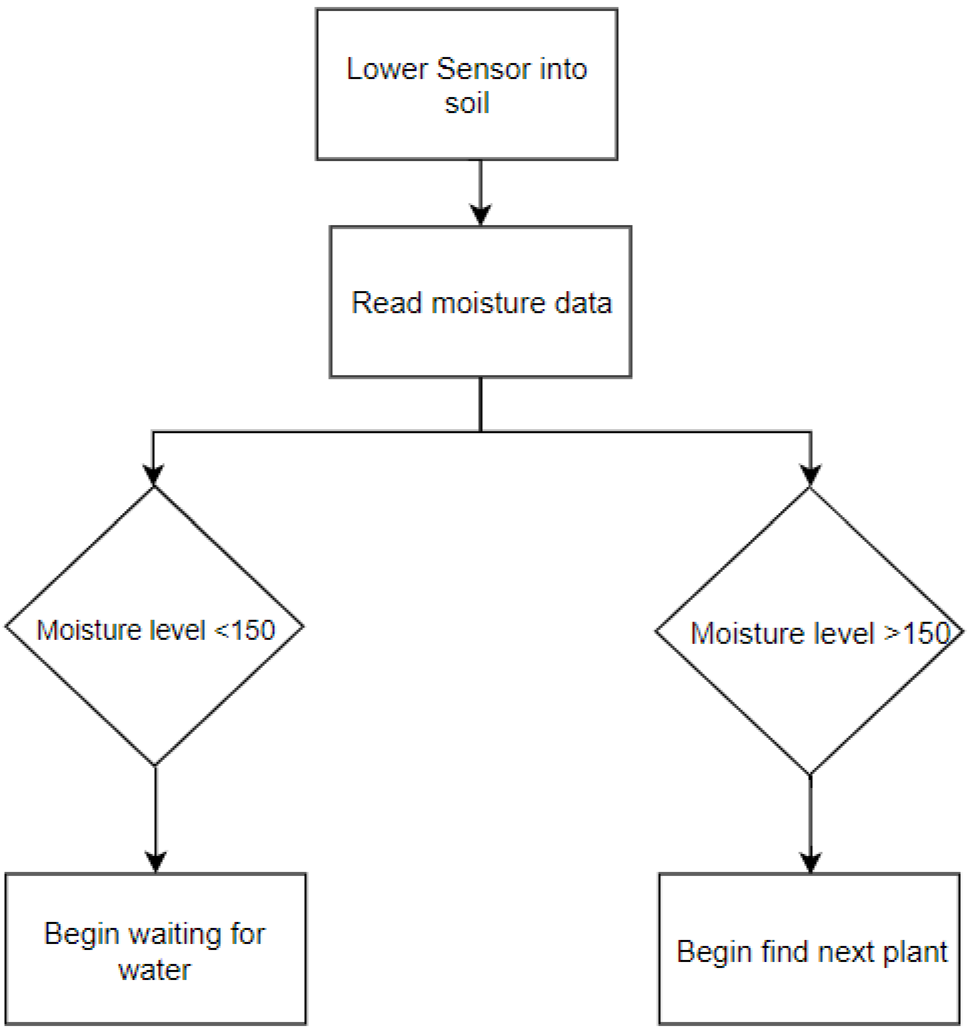

Figure 14: Soil Moisture Check Flow Chart

Waiting For Water
When the plant needs water, the LCD display will tell the user to water the plant and the robot will wait until the user has watered the plant. Additionally, a green LED will turn on which represents that water is being sprayed or poured onto the plant like a water pump. The robot will then wait for either the override button to be pushed or for the moisture level to be greater than

or equal to 150. When either of these are achieved the robot will move into the find next plant behavior. If none of these conditions are met the robot will stay in the wait for water mode.

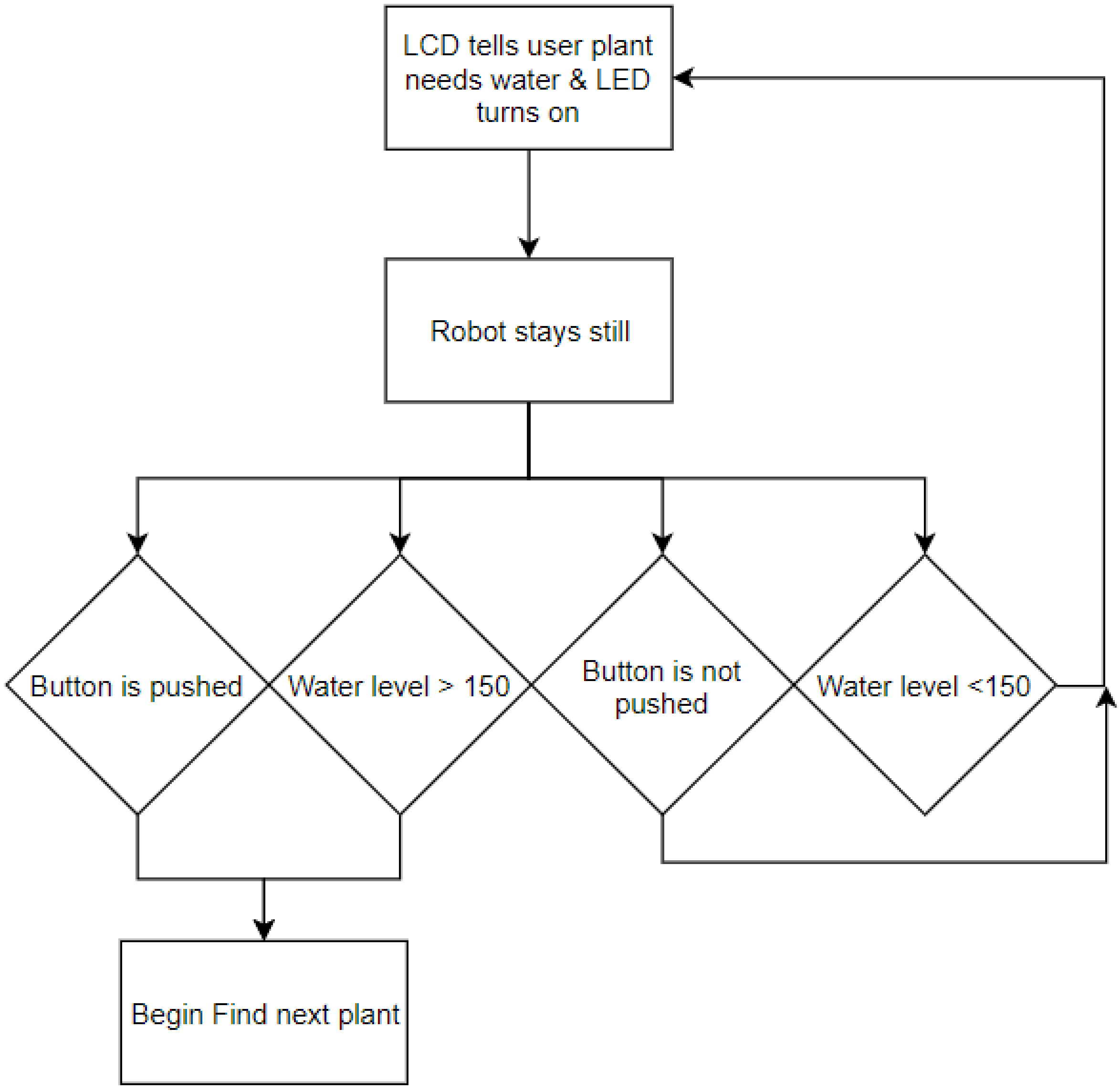

Figure 15: Human Interaction Behavior

Find Next Plant

After checking the status of a plant and verifying that the moisture level is adequate the robot will reverse for a set distance and will spin around 180 degrees. After this is done the robot will move back to behavior 1.

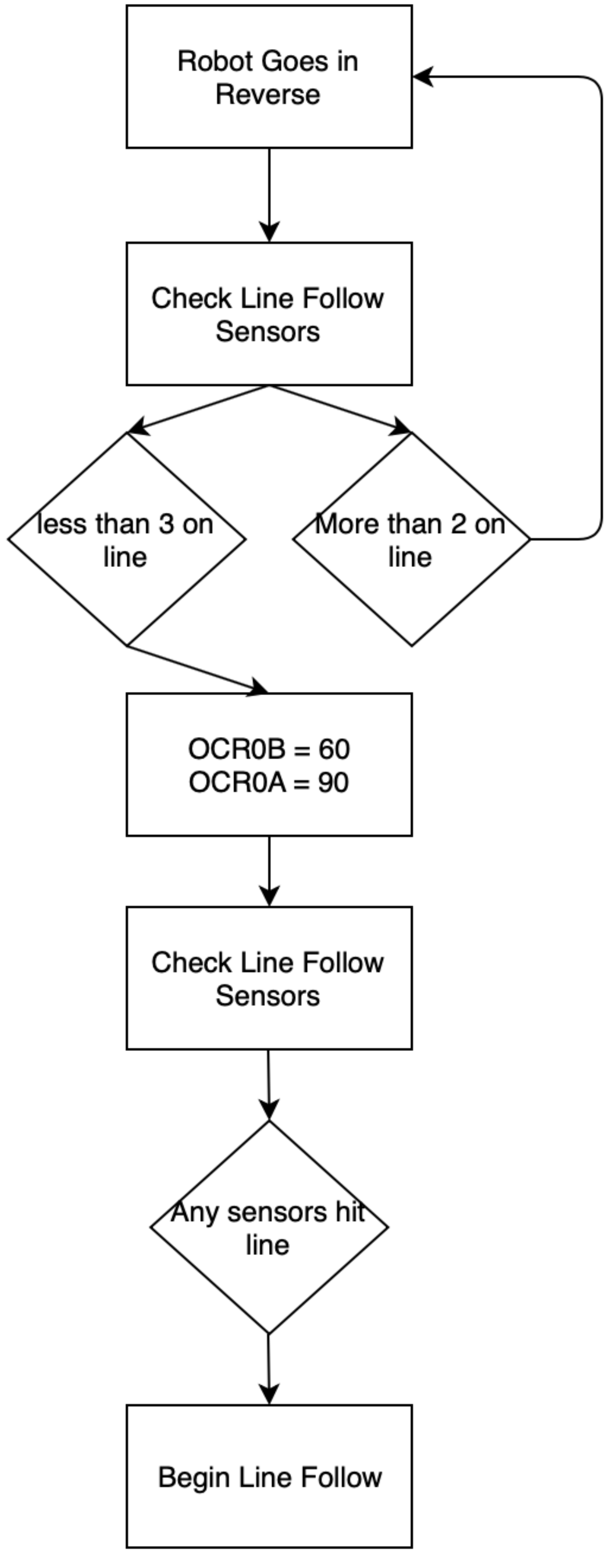

Figure 16: Find Next Plant Behavior

Obstacle Detection

After the bump sensor has been triggered the robot will check all the line following sensors. This is to detect if the robot has found a plant, or hit an obstacle while driving on the way to the plant. If the robot checks the line following sensors and all sensors show there is a line then the robot will jump to behavior 2 and begin the moisture check. Alternatively, when the robot has hit something and detected that it isn't a plant, the wheels will stop and a horn will beep to notify the user that an obstacle was hit. When the obstacle is removed, the robot will continue driving.

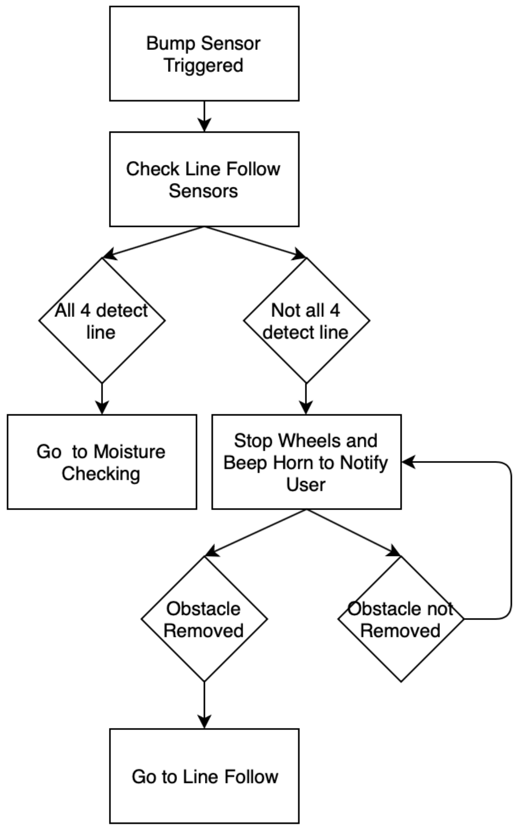

Figure 17: Obstacle Detection Feature

Conclusion

Our robot did a great job of completing the task. It went through all the behaviors during the first demonstration without needing to interfere at all. However, there were many things that our robot did struggle with. The biggest one that plagued us throughout the project was lack of friction on our tires. This would result in our robot getting stuck when trying to turn. Also, the line following sensors we used had a very narrow window of operation. This meant when turning over the tape the sensors would hit some times resulting in the robot getting stuck or when trying to find the next line the robot wouldn't be able to make the initial turn over the line it was already on. The solution to this problem was to make the robot back heavy and 3D print an array holder so the sensors could be held at a very specific level. In the future, a larger chassis with stronger motors would fix a lot of the issues that this robot had. Stronger motors means there could be more weight on the chassis, causing more friction between the wheels and the ground. The larger chassis would help with how messy our final robot looked. The final product looked more like a jumble of wires than a robot, as shown in figure ?. The larger chassis would allow more room for routing wires. Also, if the final robot were to include an actual pump instead of an LED representation of a pump, there would have been nowhere to put it.