Phase 1 Specification

Nathan Lee, David Luo, Archie Deng, Ryan Nguyen, Henry Do

We have 3 main deliverables corresponding to 3 aspects of our smart plant caretaker. For phase 1, we aim to have the 3 aspects functioning independently.

AI Image Detection:

To detect images, we used a Raspberry Pi 4 Model B board with an attached V2 camera module. The camera module allows the Raspberry Pi to observe its surroundings and recognize objects that are within its field of view, so that the system can interpret those recognized objects accordingly. We decided to use OpenCV as our program to detect and recognize objects using the Raspberry Pi's camera. Using the OpenCV library, a labeled library of plant images needs to be found and used to train a model. The detection follows the instructions here: https://core-electronics.com.au/guides/object-identify-raspberry-pi/.

For Phase 1, we plan to have a basic model that detects and recognizes basic objects. Once we have the basic model created, we plan to narrow its scope down to focus more on plants and have a greater accuracy detecting them. Adjustments to settings / optimizations of the camera and model will be needed. The computer vision code must eventually communicate with the watering system (and likely the STM32) to water plants when detected.

Autonomous Movement:

To program the autonomous movement capabilities for the robot, we first started off by wiring the motors to the H bridge and the distance sensors to the raspberry pi. The next step involves programming the raspberry pi in python to respond to move the motors in response to various commands. In total, we will program 6 separate commands including movement forward, moving backwards, turning left, turning right, pivoting left, and pivoting right. Without the H bridge, we would have only been able to move the robot forward. Although backwards movement is not necessary for the robot, programming backwards capabilities added allowed extra mobility and easier obstacle avoidance. For initial testing of commands, we will program different keys to correspond to the various robot commands to ensure the direction modules functions correctly. We will also attach the Edimax adapter to allow for wifi communication between the computer and robot. After ensuring the direction commands functioned as expected, we then wire the distance sensor utilizing a breadboard to the raspberry pi. Using the distance sensor, we are able to program the robot to change direction if it sensed an object in front of it. We then finally program the robot to respond to random direction commands simulating random autonomous movement.

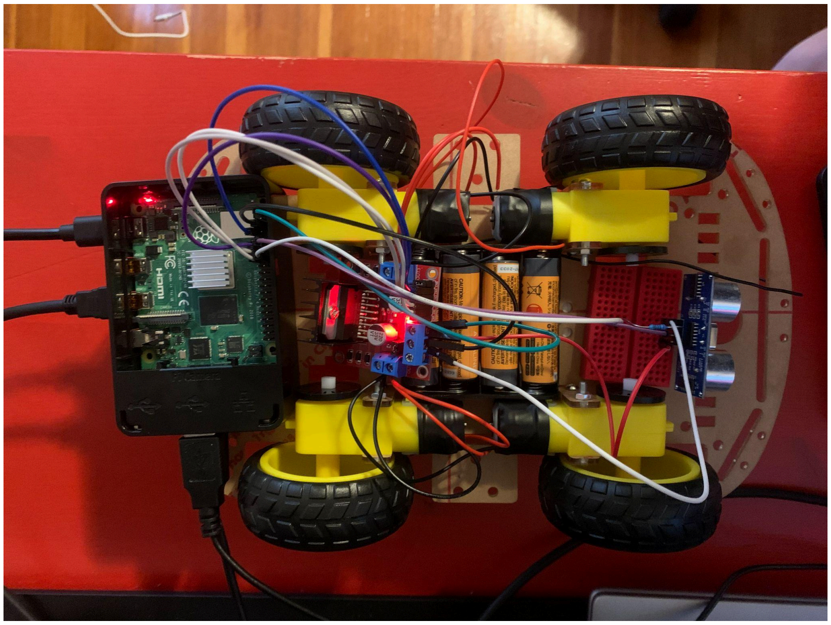

Figure 1. Layout of the robot

Plant Watering Capabilities:

We first bridged the water pump to a tube connected to a water bottle filled with water. Next, we plan to use the STM32 board to turn on and off the water pump. Lastly, we will ensure the water pump functions correctly and can turn the flow of water on and off in response to Raspberry Pi signals.

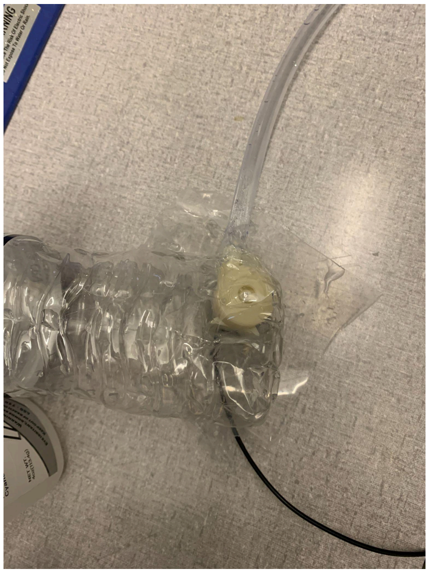

Figure 2. Water tank and pump connection

A general block diagram is shown in Fig. 3.

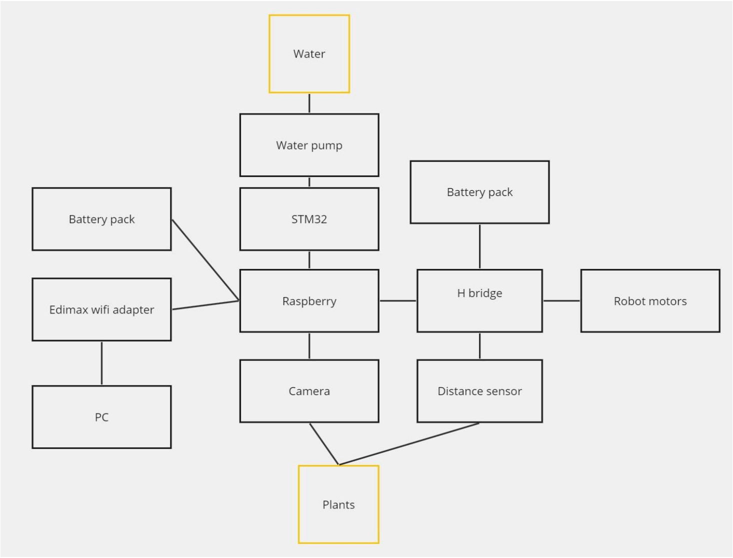

Figure 3. General block diagram of components / peripherals

Phase 2:

For phase 2, we plan on connecting each of the separate components we have developed allowing them each to communicate with each other to form the finished product. The first step will be to connect the camera and AI image detection programs to the robot autonomous movement capabilities. We will need to program the robot to stop long enough every time it senses an object for the camera to determine if the object in front of it is a plant. Next, we will need to send a signal to the water pump to release water if the object is a plant. After watering the plant for 5 seconds or if the object is not plant, the robot will then continue its previous autonomous movement until it senses another object. In addition, we could also start to implement additional capabilities during phase 2 such as a second distance sensor, plant fertilizer, and further optimization of the AI image detection algorithm.