Self-Sufficient Farm Project Final Report

Jack Aylesworth, David Strom, and Brianna Levasseur

Introduction

The device is a self-sufficient farming system for a plant. The main feature of this device is an actuated valve which will release water into a pot containing a plant at timed intervals based on the conditions indicated by the moisture sensor in the soil. The inspiration for this project is the prevalence of farming in the Jaghori and Bamyan City and the high poverty rates in Bamyan City. This project will hopefully inspire children in these regions to come up with their engineering solutions to food poverty including creating self-sufficient farms allowing for greater crop yield with less labor intensity.

The main objective of our project is to remove the need for periodic watering for a plant by automating when water is dispensed. This timing will be determined by a moisture sensor that periodically outputs moisture and temperature values. This signal would be intercepted by the Arduino Uno where it would interpret the sensor's output for if the soil is wet or dry. If the soil moisture is low enough, the Arduino will activate a stepper motor which will actuate the valve to let gravity bring water through the tubing until enough water sufficiently saturates the soil. If the temperature sensor within the moisture sensor inputs values indicating a day-night cycle has passed the valve will also release. Otherwise, no water will flow. This system will allow the farming of a potted plant while only having to fill the reservoir much more infrequently than watering it daily.

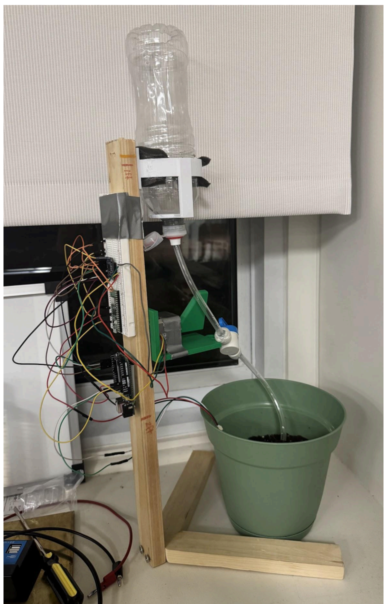
Materials and Methods

Bill of Materials

Part Name	Vendor/ Link	Description	Qty	Cost (w/o tax)
3D Printed Sprinkler Head Tube Attachment	Innovation Studios 3D Printers, WPI	This was added to the tube end in the plant soil, it's meant to ensure that the water is equally divided within the soil. Using GrabCAD and other online models to develop our own on SolidWorks.	1	N/A
3D Printed Stepper Motor-to-Water Valve attachments	Innovation Studios 3D Printers, WPI	This was the piece which allowed the stepper motor to actuate the water valve	1	N/A

9V Battery with wire connector	Higgens Lab 031, WPI	This was the external power source used to fully power the stepper motor	1	N/A
Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor JST PH 2mm	https://www.adafruit.com/product/4026	Moisture sensor used for the soil of the plant and used to record temperature to interpret data for a day-night cycle	1	$17.50
Adafruit Stepper Motor - NEMA-17 size	Higgens Lab 031, WPI	Used to open/close the water valve, the main actuator that was wired to the microcontroller	1	N/A
Arduino Uno Rev 3	https://store.arduino.cc/products/arduino-uno-rev3	Microcontroller used for processing input and output	1	N/A
Eastrans 10ft x ⅝" ID Clear Vinyl Tubing Flexible Hybrid PVC Tubing Hose	Higgens Lab 031, WPI	Tubing used for getting the water from the reservoir to the plant	1	N/A
Experimental Stand	Innovation Studios Maker Space, WPI	Using a few spare pieces of wood and two 3D printed parts, one to hold the motor and one to hold the water bottle, we made a stand that functioned as the main terminal for all elements of the experiment to meet	1	N/A
Half-Breadboard	Higgens Lab 031, WPI	Used for connection between all wired components	1	N/A
Jumper Wires (female to male)	Higgens Lab 031, WPI	Used for wiring of all components	1	N/A
Pot and Soil	Home Depot	This was the "plant" used in the test for our design	1	$9.00
PPSV041212WHD Shut Off Valve	Home Depot	Valve used as the main actuator for letting water flow/stop flow through the tubing	1	$9.99

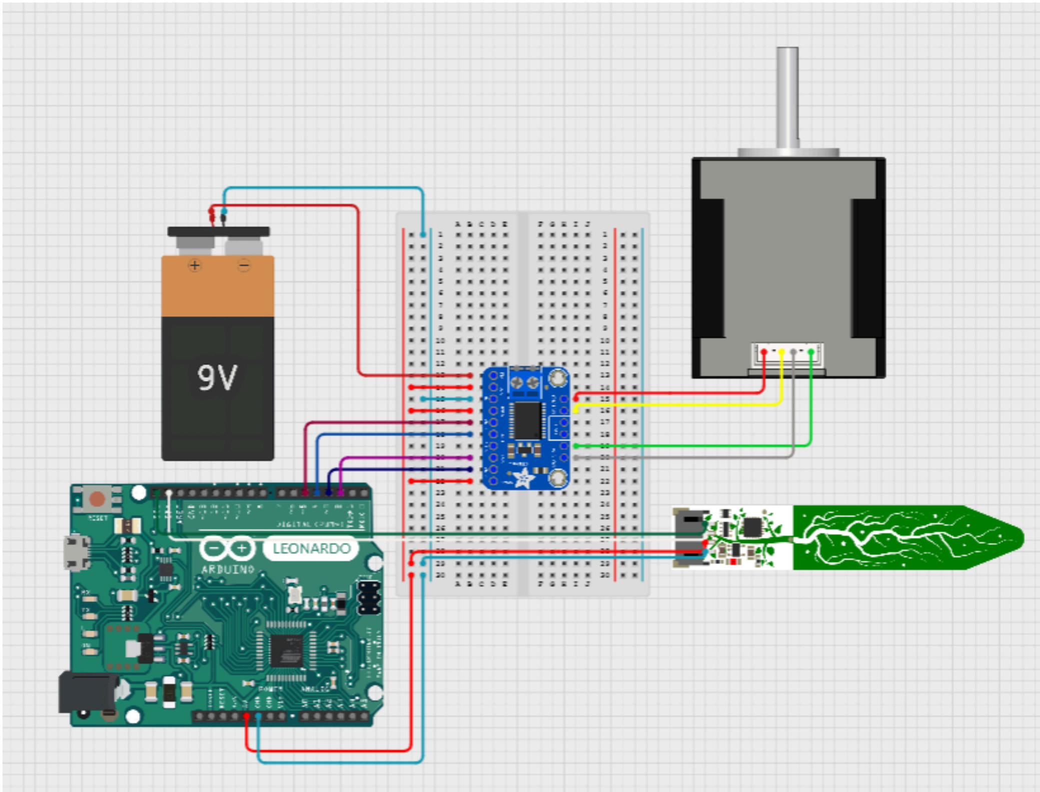
TB6612fng Motor Driver	Higgens Lab 031, WPI	Used to actuate the stepper motor	1	N/A
Water bottle with squeezable top	Any convenience store	This acted as the water reservoir to store the water used	1	$2.00



Experimental Setup:

The potted plant was positioned beneath the stand holding the water reservoir with the rest of the components above the plant. The filled water reservoir should be connected to the tubing with the water valve leading to the sprinkler head tube end, which should be embedded in the soil. The Arduino Uno and half-breadboard should be attached to the stand, and the stepper motor should be positioned on its designated stand in front of the valve with its valve attachment on. The moisture sensor should also be embedded in the soil, just deep enough so that its small bit of circuitry is still sticking out.

When it comes to the wiring, the Arduino was wired for power into the half-breadboard (5V and GND). The moisture sensor was connected to power through the breadboard and connected directly to the Arduino for SCL and SDA (each of the wires of the sensor are labeled in the manual provided in the appendix). The motor driver was directly attached to the breadboard, and the stepper motor was wired directly to the motor a and b inputs (red and yellow to MOT A and green and gray to MOT B). The motor driver also received power from the 9V battery, which was necessary for the stepper motor to actuate with maximum torque. Each of the four pins of the Arduino used for the sequence of the stepper motor were wired appropriately to the motor driver, allowing for a connection from the microcontroller to the motor. In the image on the left, there is a Raspberry Pi Pico attached to the half-breadboard which can be ignored.



Above is a wiring diagram detailing the specificities of what was summarized above. Although in this diagram an Arduino Leonardo is used instead of an Arduino Uno Rev 3 only because the software used to create the diagram did not have that specific microcontroller. The connections are still the same as the Uno, and the code provided will work specifically for the Uno.

As for the tubing and the valve-to-motor connection, the tubing was hot glued to the two sides of the valve for a water-tight seal, and the motor-to-valve connector piece was meant to be glued directly to the blue valve handle. All in all, the readings from the moisture/temperature sensor go to the Arduino, the Arduino processes the data and actuates the motor using the motor driver, and the motor actuates the valve, which lets water flow to the plant at the appropriate time.

Results and Discussion

The majority of this experimental system worked quite well. The moisture sensor accurately recorded data detailing whether it was placed in soil that was properly watered or soil that was too dry. The moisture sensor also recorded accurate temperature data, albeit at a lower accuracy compared to the moisture data it produced. Using the code provided in the annex, the Arduino successfully took the moisture and temperature data and interpreted the optimal time for the stepper motor to actuate the water valve. Although the hot glue was strong enough to create a water-tight seal on the two halves of the tubing, and the valve blocked the water effectively without leaking, the valve required too much force for the stepper motor to consistently open and close it. The details for each of the tests conducted on each component of the device are summarized below.

Design Testing

Design Objective	Experimental Testing for the Objective	Results
Moisture sensor records accurate moisture and temperature values	By placing the moisture sensor in wet soil, it should send a higher value than whatever value it should send when it is in dry soil. Additionally, it should send a temperature value accurate to that of a thermometer in similar conditions	The moisture sensor **successfully** sent accurate moisture and temperature values to the Arduino. Moist soil is recorded with a value of about 800 or above, dry soil with around 600-700, and no soil (air) with around 350.
Motor-to-valve connecting piece lets the motor actuate the water valve	By having the motor spin while connected to the valve via the 3D-printed attachment, the valve should be able to be opened/closed. If the valve can be opened solely by spinning the motor the test is successful.	The stepper motor **failed** to provide enough torque to open/close the water valve even with the added lever supporting force with the connector
Water valve can control the flow of water through the tubing	By filling the reservoir with water and attaching the valve with tubing, we can see if the valve provides enough pressure to stop the water from flowing as well as if the opening is big enough for it to flow through. There should also be no leakage. The test is successful if the water flows through the full length of the tubing.	The water valve **successfully** controlled the flow of water through the tubing
Dispersion piece spreads water out evenly while being embedded in the soil	Once the water gets through the tubing consistently, we can move the end piece around until we find the optimal spot for the water to spread evenly within the soil. The test is successful if the piece waters the soil proportionally, creating a radius of wet soil measuring at least 2 inches.	The dispersion piece **successfully** spread the water from the reservior in a radius of about 2.1 inches while being embedded in the soil

Considering how tight the valve purchased for this experiment was, it is easy to see that if time alluded us to find or create a loose valve, the device would have worked perfectly, as the motor would have likely been able to turn the valve with the assistance of the connector piece. There was no formal test conducted on the connector piece itself, but the print did come out with an efficiently sliced hole for the motor, as it was able to be fit on firmly and the motor could spin it without slipping.

Conclusions

The essential lessons of this project were the overall process of experimentation and the fundamental aspects of design management. Throughout this process, there were many instances of designing tests and experimenting in ways that we had never been exposed to prior to this class. If we were to do this project again we would have tested the moisture sensor earlier, since the fact that we were only able to do it during the final week made things a bit difficult. This was because all of the previous code for the device (mainly for actuating the stepper motor) was written for a Raspberry Pi Pico, as it was convenient at the time. However, once we looked through the manual for the moisture sensor and tried to run code for it, it was far easier to use it with an Arduino Uno, which meant we had to rewrite the code for it. We would also obviously spend more time on designing our own valve or researching a specific valve to use. Because the valve was only introduced into the project in the last 2 weeks or so we did not have that option.

Appendix (A zipped folder including the following will be submitted with this report)

1. Code used with comments on how it works
2. STL Object files of each 3D printed part
3. Manual for the moisture sensor