Using IoT to Mimic the Benefits of In-Person Plant Care

Paper ID: 11
Computer Science and Software Engineering
University of Washington, Bothell
Bothell, Washington, United States

***Abstract*— Since the industrial revolution there has been an explosion of urbanization which shapes every aspect of our lives from what we do to what we see. Though the majority of these changes have been positive and brought great convenience for us, we have lost much of the natural imagery that used to surround us on a daily basis. Research has shown that exposure to nature and plants has a positive effect on our cognitive wellbeing in more ways than one, and that the lack of any natural imagery has a contrary effect. Through Internet of Things technologies linked through the cloud to a mobile application we can create a system that gives users a portal to the plants they already own and encourages frequent viewings and interactions with said plants. The system provides value to the user as a plant care tool so that it seamlessly synergizes with the typical modern lifestyle to avoid artificially creating tasks the user would not already want to do. Early results show that users benefit from interactions with the system in similar ways as described in the plant interaction studies.**

***Index Terms*— Internet of Things, plant, watering, monitoring, smart home, mental health**

I. Introduction

Our lives have been exponentially improved by the technology we've built and the advancements we've made since pre-industrial times. We enjoy many comforts and conveniences that are designed to take the hassle out of our daily routines. However, one may also notice that the lives we lead today are becoming increasingly distanced from the environment our biology evolved around. While we enjoy the benefits of things like urbanization, we've begun to lose the greenery and nature that our minds and bodies have been tailored for.

The BBC wrote an article in 2019 describing a phenomenon known as 'plant blindness' where we fail to notice, or appreciate, the plants and greenery around us [1]. The article explains that people are paying less attention to the plantlife around them which has cascading effects from simply being less aware of the plants around them to having greater apathy towards plants. Without getting into the ecological dangers of placing less value in plants and nature, there are many more direct reasons why we should be spending more of our time and attention observing and interacting with nature. There has been a growing body of work investigating the effects of including more natural imagery and plant care in our daily lives.

It's not uncommon for people to choose to take a step in this direction on their own accord. Whether or not most plant owners recognize that plants to not contribute to significant air quality improvements [2], a survey in 2020 showed that over 30 million households participated in indoor houseplant gardening every year over the past decade [3]. Although the benefits may not be as direct as air cleaning, there have been countless psychological and anthropological studies that show how increased plant and nature involvement in our lives benefit us in less apparent ways.

Additionally, such a large population of plant owners incentivizes further development and research into methods of improving plant care. Research on this has been split largely between small, indoor plant care and large agricultural systems. The primary goal of nearly all the systems proposed thus far have been to fully automate the care of plants, essentially working to minimize the needed interactions of the plant owner. This approach is useful in large-scale agriculture and irrigation systems, but on an individual scale it only works to build another wall between us and nature.

The system proposed in this paper works to bring the advancements of Internet of Things (IoT), Amazon Web Services (AWS) cloud, and mobile technology to lessen the distance that technology puts between people and nature. The Emerald Thumb system is designed to be a tool for remote plant care that still promotes frequently viewing plants and the types of thoughts and considerations that come with caring for an indoor plant. The crux of this is the idea that people will be more willing to view and care for an indoor plant of their own rather than attempting to show plants or nature imagery that a person feels no connection to. The goal is to increase people's exposure to plant imagery and care in a convenient way that synergizes with their current lives while still providing value as a remote care service for their plants.

II. Related Work

Many attempts have been made at using technology to ease the demands of plant care; however, much of the effort has been directed at total automation. This emphasis on fully automated systems goes directly against the body of research that finds benefits in regularly seeing and interacting with living plants, essentially attempting to remove the care aspect of owning a plant. Additionally, these works tend to focus on agricultural and ecological benefits of smart plant watering rather than providing toolsets that better mimic in-person plant care, both in terms of care options and benefits to the caregiver.

In 2019, J. Kumar et al [4] describe a system in which IoT technology is used to read measurements remotely and control water supply. The efforts of this system are primarily to reduce water wastage. Their device uses NodeMCU, a microcontroller likened to Arduino except with built-in Wi-Fi, and a DHT-11 sensor which provides humidity and temperature digital readings. This provides a reliable core device for their automated system; however, it lacks a camera module and is limited in its scope of user inputs and processing power. Similarly, Muhammad et al [5] created a plant irrigation system utilizing IoT with a Raspberry Pi 4. Their stated purpose was to aid Malaysian farmers in raising crops with climate change affecting their yields. Although their sensing device is more like the one described in this paper, it is unnecessarily powerful for its simple task and all the data is managed by a web based third-party IoT service, Ubidot.

On the side of anthropology there has been a strong rise in research examining the benefits of the presence of, and regular interactions with, live plants. In a *Plants for People* conference in 2002, Ulrich highlighted the many benefits of having plants, gardens, and nature in a healthcare setting [6]. The paper brings up several studies that show that viewing plants and nature has stress reducing effects, increase staff satisfaction, and even improve healthcare outcomes.

There have already been many attempts at exploring the effects of interacting with nature and plants on a person's cognitive wellbeing. A study in 2011 found that gardening had a stronger stress reducing effect than even reading [7]. In fact, there are many studies demonstrating the benefits of gardening [8], however these are slightly distanced from the focus of the system outlined in this paper due to our particular interest on indoor plant care, but it's important to demonstrate that the benefits of plant care are seemingly universal. Lee at al published a study in 2015 further describing the stress-reducing effects of interacting with indoor plants [9]. Specifically, the study compared the feelings of young adults when performing a "computer task" versus a "plant care task". In this study the computer task was working on a word processor document and the plant task was transplanting. The groups carried out their tasks on the first day and then switched tasks on the second day. The findings show that the adults performing the computer task had significantly higher sympathetic nervous system activity and felt less comfortable and soothed relative to those performing the plant care activity.

More recently, a study from 2019 examined these benefits in elementary school children [10]. By observing the students' brain activities with EEG measurements, the team was able to demonstrate a notable decrease in the frontal lobe's theta waves when shown plant imagery. The team associates this response with improvements in attention and concentration, in addition to the separately observed positive mood states, again showing increased comfort. Findings like these are becoming increasingly common as we take a critical eye to our industrialization and urbanized lifestyles which move us further from nature and natural imagery.

These, and many more, studies show that the need for increased nature imagery in modern life is growing as natural scenery is replaced with modern efficiency. The advancements in Internet of Things technology have made it possible to connect and interact with plants remotely and reap the benefits of owning a plant even when not in its proximity.

III. System Model, Problems Statement, and Analysis

A. *System Model*

This system is comprised of a physical sensing and water dispensing device, cloud resources, and the implemented software on the device and in the mobile app. The device and app are connected through Amazon Web Services cloud services, specifically their IoT Core, Lambda, and S3 services. The app is designed for Android devices and uses the AWS Amplify library for its cloud functions instead of the Android Mobile SDK as support for the SDK was stopped on October 30, 2021.

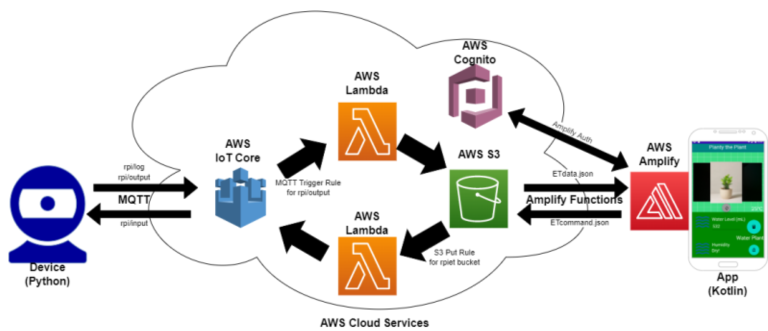

Fig 1. System overview diagram

The device is comprised of a Raspberry Pi Zero W, a camera module, a soil moisture sensor, and a motor and relay. The mobile app is designed to give a few statistics on the plant's conditions, but more importantly it will display front-and-center a large image of the plant for the user to see while interacting with the app. This implementation is designed specifically for Android devices and written in Kotlin. The AWS Cloud is responsible for the management of data and the communications between the device and app.

B. *Problem Statement and Analysis*

The goal of this project is to increase and incentivize viewing and caring for plants. Though the system has been designed for indoor plant care, it could theoretically be used outdoors as well given a stable network connection and some cover from the elements. To incentivize this, the system has been built as a remote care service for individuals to monitor their plants. With the slow timescale that plants live under, it can be very difficult for a person to notice signs of poor health, and even harder to bring the plant back to its former vigor. Plant care is primarily proactive and requires the owner to be informed on the plant's health before any problems arise. For this, the system must include metrics associated with plant health which are actionable by the owner. Arguably the most important aspect of plant care is its ideal watering schedule, which can wildly vary based on several factors. Light, temperature, season, size, and many more can affect how much water a plant needs and how frequently it's needed for each species.

Most plant owners are capable of finding information on the plant's needs online, however it can be difficult, if not impossible, to understand the plant's current state below the soil's surface and decide what type and level of care the plant needs. Therefore, the Emerald Thumb system is built around filling this gap in information, allowing the user to make better informed decisions and provide more tailored care to their plant. Armed with this, the system also provides the user the tools to remotely water their plant and set the amount of water to dispense. This encourages the owner to actively interact with the system and regularly be shown images of their plant.

IV. Design and Implementation

A. Physical Components

The device itself is a Raspberry Pi Zero W, chosen for its low price point, versatility, and integrated wireless network capability. It runs on Raspbian with NOOBS for the minimal needs of this system.

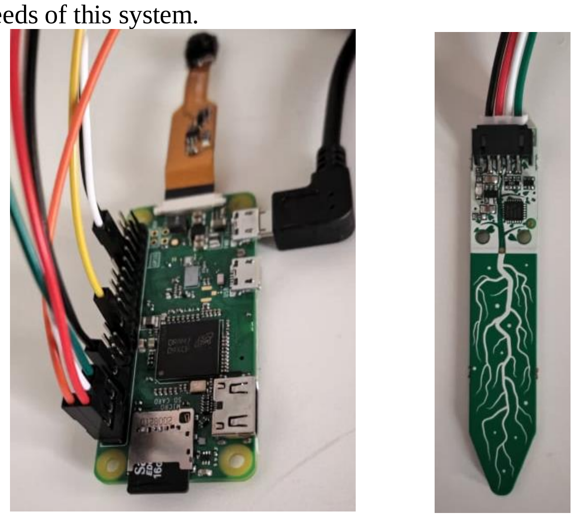

Fig 2. Raspberry Pi Zero W (left), Soil sensor (right)

The soil moisture sensor is an Adafruit STEMMA sensor designed specifically for this purpose. This sensor uses capacitive measurement, in contrast to the systems described in the Related Works which use the more common resistive measurement. The benefit of capacitive measurement is that it does not use any exposed metal, which are liable to oxidize over time and slowly increase in resistivity, requiring recalibration over time. The capacitive method does not suffer from this and don't have the drawback of introducing a DC current into the plant's soil. Of small note is also the capacitive sensor's improved performance in loose soil. This sensor also provides an ambient temperature reading, which is a useful metric for plant care and will be included in the mobile app's readout. Using the sensor is as simple as inserting it into the soil up to the top of the root image where the colors invert. Additional wiring was used to extend the reach of the soil sensor from the system casing. The sensor uses the default 0x36 I2C address for data transmission. It is connected to the Pi with a STEMMA cable, pictured in Figure 2. The color-coding connection guide for the STEMMA cable is black (GND) to GND, red (VIN) to 3V3, white (SDA) to SDA, green (SCL) to SCL.

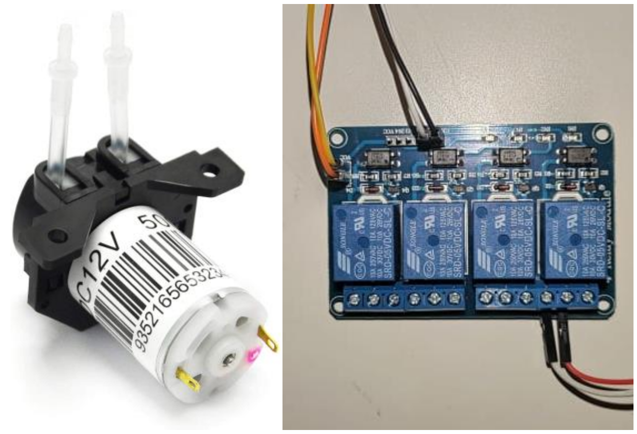

Fig 3. Peristaltic pump (left), 4-channel motor relay (right)

The pump used is a 12V 5000RPM peristaltic liquid pump, meaning that the liquid never comes in contact with the pump itself. This means that water can safely be pumped from a reservoir to the plant without introducing any harmful substances. Additionally, the pump's small size allows the system to remain fairly compact and movable. Since this pump uses a 12V motor a separate power supply must be provided, as the Raspberry Pi is only capable of outputting 3.3V of power at most. Additionally, it is highly advised to not connect a motor directly to the Pi, as fluctuations in its power draw are all but guaranteed to destroy the Pi. The system uses a 4-channel motor relay module with support for multiple plants in mind, however only one channel is used for this implementation. The relay is connected directly to the Raspberry Pi and the motor to the relay, with an additional 12V power supply interfacing between the relay and motor. This protects the Pi from any surges or fluctuations in power draw from the motor. The tubing used for the pump is cut asymmetrically, with the shorter end intended for use with the reservoir and the longer end for the plant. The main limitation of this is that the device now requires two open power outlets where the device is to be set up, however once this requirement is met the device can operate without interruption indefinitely.

The relay is connected to the Pi through two pairs of jumper cables. The first pair is an orange wire (JD_VCC) that connected to the Pi's 5V pin and a yellow wire (VCC) that connects to a second 3V3 on the Pi. The next pair is a black wire (GND) with connects to the Pi's GND, and a white wire (IN1) which was connected to Pi Pin 12. The relay is then connected to the pump and power through red and white wires screwed into one channel. The 12V power supply is connected to a wire-screw adapter which connects to the relay and pump. The relay's red wire connects from the central output for the channel to the + end of the power supply adapter. A white alligator clip wire (male) connects the leftmost output of the channel to the pump (it doesn't matter which input on the pump, it will only affect flow direction). A similarly black alligator clip connects from the adapter's – end to the other pump input.

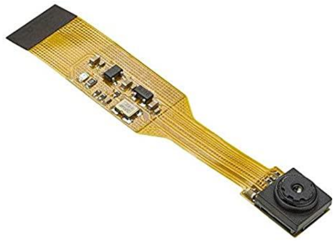

Fig 4. Raspberry Pi Zero Spy Camera

The camera used is a minimalist so-called "spy camera" for the Raspberry Pi Zero. This camera has a 5MP sensor and is capable of up to 1080p video with a FOV of 70 degrees, however it will only be used for taking pictures. This camera was selected for its relatively low cost and ease of use. The picture quality is more than sufficient in good lighting and provides a clear, vibrant image of the plant.

Fig 5. Emerald Thumb final assembly

Overall, the system is designed to have a small footprint and comfortably fit in any space an indoor plant would fit. Every component except the pump and tubing is held in a small container with a small slit to allow necessary wiring and the camera head to be exposed, but otherwise protects the Raspberry Pi itself and motor relay. The pump and tubing are kept outside this container to safeguard the internal components from water exposure in the case of a leak. Additionally, the adapter plugs for both power supplies are located outside of the container for easy disconnection and transportability. The device can be disassembled easily into the core device + soil sensor, two power supplies, and the pump with tubing. Of note is the tubing's tendency to retain water for long periods after operation has ceased, so care must be taken to avoid stray drops landing on electronics during transportation.

B. Cloud Resources

The cloud resources used are the backbone of the system. The AWS Free Tier subscription is sufficient for the functions required by this system. Its greatest limitation is the cap of 2000 S3 put requests, or in the context of this system, command and data uploads. Assuming the device remains operational for a 30 day cycle it will consume 1440 of the 2000 by uploading hourly plant metrics as both raw and processed data, which will still allow the user over 500 interactions per month. The other Free Tier limits are not low enough to be of concern for this system.

The Raspberry Pi device communicates with IoT Core to publish and subscribe to MQTT messages. It's set up as a Thing in IoT Core and has a policy attached to its certificate that allows MQTT publishing to "rpi/output" and subscribing to "rpi/input" (additionally, logging messages are published to "rpi/log"). Certificates for the device generated from IoT Core are saved locally on the device itself. Additionally, the policy must allow for the device to put objects into the S3 bucket "rpiet". The device publishes its metrics in JSON format to the topic "rpi/output", comprised of a moisture reading (200 – 2000), a temperature reading (in Celsius), and the current water level (> 0 mL). A message published to this topic triggers a Lambda function "emerald-thumb" which then processes and saves this data. This lambda converts the numeric moisture data into easy-to-understand text indicating the dryness of the soil (from 'Very Dry!' to 'Drenched!'). The same is done with temperature and water level, however for this implementation the text conversions are not used for these two metrics. The processed and raw data are saved separately to the "rpiet" S3 bucket under the "raw/" and "data/" folders respectively. This requires that the lambda function be granted read/write permissions for S3.

The Emerald Thumb device is configured to listen for MQTT messages published to "rpi/input" for commands. Commands are in the JSON format as indicated below. The current iteration of the device is capable of receiving multiple commands for ease of use to the user, outlined in the table below:

MQTT command input format:

```
{
  "command": "string",
  "value": int
}
```

MQTT output format:

```
{
 "moist": "string int",
 "temp": "string float",
 "wtr": "string int"
}
```

Fig 6. MQTT communication formats

Command	'value' parameter	Effect
reboot	Not used	Restarts the device
water	Not used	Dispenses water
setamount	Used	Set amount of water to dispense
updatewater	Used	Set a new water level
updateimage	Not used	Take a new picture
updatestats	Not used	Update temp/moisture metrics
updateall	Used	Update metrics, take a picture, and set a new water level

Table 1. Supported Emerald Thumb MQTT commands

Originally, these commands were intended to be sent directly from the mobile app to the device. However, during development AWS ceased support for Mobile Hub, which interacted with Android Mobile SDK, and began migrating its services to Amplify. Currently, there is no MQTT support for Amplify in Kotlin. As a workaround, the app creates a JSON formatted file containing the command input and uploads the file to the S3 bucket "rpiet/app/ETcommand.json". The upload is performed by a simplified Amplify command that uses TransferUtility. A successful upload triggers another cloud lambda function with a policy allowing access to IoT Core MQTT publishing and republishes the contents of the file as an MQTT message to the topic "rpi/input". This completes the data flow cycle managed by AWS cloud services. Access to the app is authorized by AWS Cognito federated identities which requires that the user create an account with a username, password, and email address. Cognito then sends an email to the user's address with a one-time PIN to confirm the account and then allows them to login unhindered with their chosen user/pass combination in future sessions.

C. *Software Components*

The Emerald Thumb device is configured with crontab to run its script after 30 seconds on reboot. The script is written in Python and uses the AWS Python SDK to connect to the cloud. The script subscribes to the "rpi/input" topic for commands and otherwise updates its measurements and picture once per hour. The image is uploaded to the "rpiet/photo/" bucket as a jpg file using a one-line console command run from within the script. The script runs in the background and logs to a file "ETlog.txt" in the same directory as the script (which is in a Desktop folder). Since this is all performed automatically on startup the device assumes that there is a network it can connect to. If not preconfigured, the device must be connected to a display and peripherals to establish a connection prior to use. On error, the device must be manually restarted. Typical causes of errors are unrecognized commands or input formats.

The mobile application was designed to be a tool that encourages more interactions with plants and a method of introducing more plant imagery into the user's daily life. This means that the image of the plant and its presentation must be front-and-center anytime the app is used. The color scheme of the app is a combination of blues and greens to prime the user into thinking about natural imagery. The icon is a small green leaf against a lighter green background, the same as the starting screen of the app. When opened, the app prompts the user to create an account and sign in with their self-appointed username and password through the backend component. This is called by Amplify through AWS Cognito and redirects to a browser-based login or signup form which then returns to the app on completion.

Once signed in, the main content of the app is displayed. The top includes a modifiable text field for the plant's name, which remains consistent between sessions. This is to encourage naming the plant and nudge the user to forming a more personal connection with their plant friend. Under the plant image a refresh button is included which will send an "updatestats" command via S3 and then download the new data to update on the app. Buttons are initialized with listeners that are mapped to a function which sends the corresponding command on button click. Once the command is sent the app waits 10 seconds before downloading the new data and applying it; this gives a buffer for slower network connections. Next to that button the most recent temperature reading is displayed. Beneath this section there is a "Water Plant" button which simply sends a "water" command to the device, and next to it there is a "Water Level" editable text field. This field loads the most recent water level received from the device, however modifying this field sends the new value with an "updatewater" command to the device. If the user attempts to dispense water when it would bring the water level to, or below, 0, the device will output an error message to "rpi/log" and take no action. Similarly, the device checks if the set water level value is greater than zero and updates its value if so. Under this the processed moisture reading is displayed.

Unlike the conversion for water level, the soil moisture level is intentionally converted to arbitrary "dryness" values and not rated as "good" or "bad" since the needs of the plant may require different levels of dryness than more common plants (think cacti and tropical plants). This current implementation of the app does not have an interface for the "reboot", "setamount", or "updateimage" commands. The default water dispensing amount is 100mL. For now, these commands can be manually executed by sending it as an MQTT message via the AWS IoT Core MQTT Test Client.

V. Evaluation

One of the major goals of this system is to make the user feel as though they are taking care of a plant, and not as though they are performing another typical "computer task", otherwise the system would have the opposite of the intended effect. Additionally, the system is meant to provide a valuable service to the user beyond benefitting their mental health, so the displayed image should be sufficient for the user to analyze the health and condition of their plant alongside the displayed metrics, as visual inspection is a natural and necessary part of plant care. To this end, the system was given to 10 individuals for their plant monitoring needs. The individuals were given the device to set up with their favorite indoor plant and requested to check on the plant's condition throughout the day. Of these ten, eight reported feeling more positively after taking a moment to view their plant in the application. This is certainly an incredibly small sample size and only provides a preliminary view into the potential for good the app can have. The subjective nature of such an experiment makes it hard to quantify the value the system brings, but further explorations may provide more concrete evidence of the benefits of a system such as this.

On the technical side, the device's responsiveness was evaluated by time between a command sent and the execution and upload of the command. For this evaluation the mobile app was discounted with the reasoning that in the real world mobile devices are constantly under fluctuating and unknown network conditions, whereas AWS and the Emerald Thumb device are under stable conditions. To control for this the responsiveness test begins with a manual upload of the command JSON file to the cloud S3 bucket, effectively starting the timer as soon as the mobile app is able to successfully upload to the cloud. 30 "water" commands were uploaded and timed until the device began dispensing. The execution on average took an average of 2283 milliseconds from upload time with notable variance, 430 milliseconds in standard deviation.

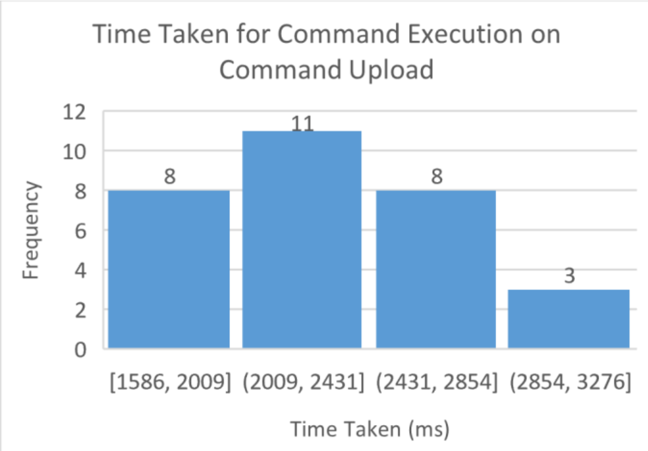

Fig 7. Histogram of the responsiveness of the system beginning at command upload

Additionally, the device's service start time was measured given a preconfigured, stable network connection. Note that the device is configured to wait 60 seconds before starting the service, as the Raspberry Pi Zero is a weak computer and takes time to start up. With this in mind, measurements of time between sending a "reboot" command until the time of the first saved processed data was 147 seconds. This value was consistent for all 10 measured trials measured to the whole second with a standard deviation of 0 seconds, where 2 seconds are attributed to the Lambda processing time (time to first MQTT output is 145 seconds).

Given this Lambda processing time in combination with the command execution time we find that a round-trip data request will take approximately 5 seconds, not accounting for the variable time taken by the app to upload the command depending on its current network environment.

Finally, the sensors' precision was evaluated over 30 measurements. The device was allowed to boot up and rest for 5 minutes, the capacitive sensor was left in open air on a tabletop. Within a window of 30 seconds, 30 "updatestats" commands were sent to the device, once per second, and each new measurement recorded. As this was performed in such a small timeframe the assumption is that the surrounding conditions were consistent and any variance in measured data is a result of the device's imprecision, and not an accurate reflection of its environment.

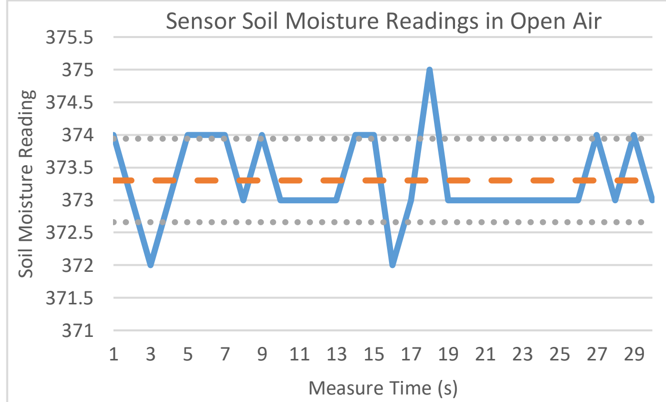

Fig 8. Soil sensor moisture readings in open air over 30 seconds

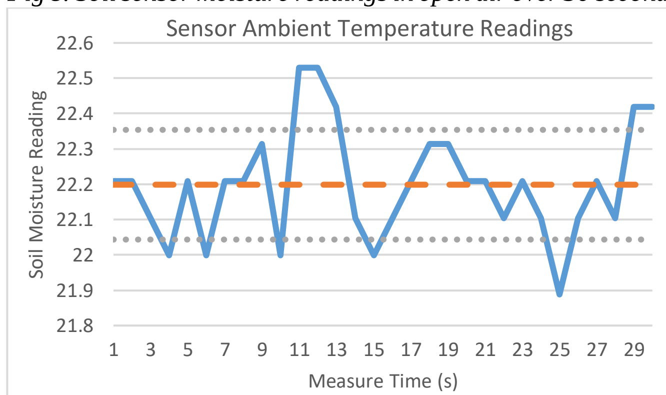

Fig 9. Soil sensor temperature readings over 30 seconds

The sensor's readings are reliably consistent with neither temperature nor moisture readings exceeding far beyond one standard deviation from the average value. The moisture readings resulted in an average of 373.3 +/- 0.64 units, and the temperature 22.20 °C +/- 0.155 °C. Both standard deviations demonstrate the device's ability to reliably give accurate readings that the user can utilize.

VI. Conclusion

Though the results are preliminary, they align well with the existing body of work examining the effects of plant-related visual stimuli on the mental state of observers. Given the small sample size it is difficult to make conclusive statements, however the existing data supports existing research. The Emerald Thumb system provides an external incentive for users to engage with plants and keep their interest in the subject without artificially probing them to look at nature pictures they have no connection to. As our modern lifestyles advance and move further from our evolutionary environment it is important that we take care to consider the effects these changes can have on our subconscious states.

The Emerald Thumb system provides a convenient way to stay in touch with the small pieces of nature that we choose to surround ourselves with at home and provides a convenient toolset to care for plants. With a total response time of 5 seconds on sent commands and a 1-hour refresh interval on metrics the system encourages active engagement and frequent interactions with the mobile application, thereby ensuring the user sees plant life more frequently in their daily routine. By keeping the device component costs at a minimum the Emerald Thumb is an affordable device that can be adopted by most plant owners.

VII. Future Works

For the purposes of expanding this simple app, the natural next step is to implement notification features at user-defined threshold values for the various metrics the app supports. These can serve to remind the user to check the status of their plant when the soil gets too dry, or water level gets too low, etc. Additionally, there is currently a preliminary "evaporation" feature which attempts to account for the natural loss of water from an open-top reservoir. This was calculated for a container roughly the size of a 2L soda bottle cut across the middle, however deeper domain knowledge can help bring this from a broad estimation to a more precise calculated value. That being said, the largest potential for future expansion lies in expanding the tool arsenal of both the device and application. The simplest version of this would be to store a history of plant measurements to organize and visualize for the plant owner in a digestible way, however this ventures further from plant care and delves deeper into "computer task" territory.

Clearly, there is potential for variable automation given the many prior related works that focus on this effort. Reghukumar and Vijayakumar attempted this in their own system by applying machine learning to the collected sensor data in an effort to optimize plant watering and minimize water waste [11]. However, to the purpose of encouraging more plant viewing and interacting, it would be preferable to minimize such features and promote regular engagement with the app. One altered approach is to use machine learning for insights on the plant that the owner would not otherwise be able to attain. Mohanty et al in 2016 developed a deep learning model that attempts to diagnose plant disease based on images [12]. Such a model would perfectly align with the goals of the Emerald Thumb system and would encourage the plant owner to take up a more proactive role in their plant's care. Similar efforts were carried out by Ziamtsov and Navlakha in 2019 where the team used machine learning of 3D point clouds to identify the different parts and genetic expressions of plants [13]. A combination of these two groups' efforts would provide a second-to-none analysis of plant health that could meet or surpass in-person visual inspection of the plant, and theoretically could be extended to recreate a virtual 3D model of the plant for deeper interaction and investment by the user.

It is important when deciding on how to modify this system to be cautious of turning plant care into too much of a "computer task" that works against the intended goal of calming and comforting the user as real interactions with a plant would. These proposed additions would all further the goal of increasing user interaction with their plant and therefore improve the rate at which plant imagery is presented to them without automating away the care aspects and inadvertently encouraging "min-maxing" behavior, or the compulsion to optimize interactions for a cognitive reward.

VIII. References

[1] C. Ro, "Why 'Plant Blindness' Matters — and What You Can Do about It." BBC Future, April 28, 2019.

[2] M. Apte, J. Apte, "A Study of the Effectiveness of Indoor Plants for Removal of Volatile Organic Compounds in Indoor Air in a 7-story Office Building", Epidemiology: January 2011 - Volume 22 - Issue 1 - p S46

[3] S. Lock, "Indoor houseplant gardening participation in the United States from 2010 to 2019" Statista, May 2020.

[4] J. Kumar, N. Gupta, A. Kumari and S. Kumari, "Automatic Plant Watering and Monitoring System using NodeMCU," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 545-550.

[5] Z. Muhammad, M. A. A. M. Hafez, N. A. M. Leh, Z. M. Yusoff and S. A. Hamid, "Smart Agriculture Using Internet of Things with Raspberry Pi," 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2020, pp. 85-90

[6] R. Ulrich, "Health Benefits of Gardens in Hospitals," presented at Plants for People at the International Exhibition Floriade, Haarlemmermeer, Netherlands, 2002.

[7] A.E. Van Den Berg, M.H. Custers, "Gardening promotes neuroendocrine and affective restoration from stress". J Health Psychol. 2011 Jan;16(1):3-11.

[8] J. Webber, J. Hinds, and P. Camic, "The Well-Being of Allotment Gardeners: A Mixed Methodological Study" Ecopsychology 2015 7:1, 20-28. 24 Mar. 2015.

[9] M.-S. Lee et al. "Interaction with indoor plants may reduce psychological and physiological stress by suppressing autonomic nervous system activity in young adults: a randomized crossover study." Journal of physiological anthropology vol. 34,1 21. 28 Apr. 2015.

[10] Y.-A. Oh, S.-O. Kim, S.-A. Park, "Real Foliage Plants as Visual Stimuli to Improve Concentration and Attention in Elementary Students". International Journal of Environmental Research and Public Health. 2019; 16(5):796.

[11] A. Reghukumar, V. Vijayakumar, "Smart plant watering system with cloud analysis and plant health prediction" Procedia Comput. Sci., 165 (2019), pp. 126-135

[12] S. Mohanty, D. Hughes, M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection". Frontiers in Plant Science, 22 Sept. 2016.

[13] I. Ziamtsov, S. Navlakha, "Machine Learning Approaches to Improve Three Basic Plant Phenotyping Tasks Using Three-Dimensional Point Clouds", Plant Physiology, Volume 181, Issue 4, Dec 2019, Pages 1425–1440