IoT-Based Smart Watering System Towards Improving the Efficiency of Agricultural Irrigation

Thilina N. Balasooriya
Department of Physics
Arizona State University/
Hamilton High School
Tempe, USA
tbalasoo@asu.edu

Pranav Mantri
Hamilton High School
Chandler, USA
pmantri@asu.edu

Piyumika Suriyampola
School of Life Sciences
Arizona State University
Tempe, USA
psuriyam@asu.edu

2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT) | 978-1-7281-8420-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/GCAIOT51063.2020.9345902

***Abstract*—A large amount of water is wasted in agriculture today due to inefficient irrigation techniques. By monitoring the soil moisture of crops and the pH level of the irrigation water, not only can water be conserved, but healthier plants can also be cultivated. Even though several smart watering systems have been proposed, currently none of the proposed systems consider both the pH of irrigation water and soil moisture together. This research proposes an IoT-Based Smart Watering System (IBSWS) which addresses both of these concerns by using pH and soil moisture sensors to take real-time data and process it through a cloud environment using microcontrollers. This enables continuous monitoring of soil moisture and pH levels. In addition, IBSWS implements a mobile app for farmers who use the system to monitor and control the irrigation system as well as the crop environment. The IBSWS prototype demonstrates that the use of sensors and WiFi-enabled microcontrollers over a cloud environment can be used to implement such a system and properly manage crop irrigation.**

***Keywords*-Internet of Things; Smart Watering; Irrigation; Cloud Communication; Mobile App; Water Conservation.**

I. INTRODUCTION

The Internet of Things (IoT) is a system of interconnected devices, sensors, and computer systems working together over a network to accomplish assigned tasks. IoT sensor-based systems often help gather data about the physical world and interpret them with computing algorithms to find the optimal solutions to problems. Currently, IoT is beginning its integration into agriculture with various methods to implement smart irrigation including humidity sensors, soil moisture sensors (SMS), and evapotranspiration sensors (ETS) [1]. This leads to reduced chance for crop damage/loss while also accounting for cost and waste management [2]. Water is a valuable resource, with fresh water only accounting for a small percentage of the world's water supply. Large amounts of freshwater are used for agriculture, which claims no less than 80% of the United States ground and surface water and over 90% in many western states [3]. Problematically, 60% of the water diverted or pumped for irrigation is wasted partially due to inefficient watering methods [4]. With agricultural yield feeling pressure to increase as the world population grows, the need for irrigation systems to become more efficient is imminent[5]. Soil pH is another important factor in plant growth, as it affects the ability of nutrients to dissolve in the water present in the soil [6]. Unsafe soil pH is problematic as it can cause an excess or deficiency of dissolved minerals in the soil water content, which can hinder plant growth [7].

The objective of the IoT-Based Smart Watering System (IBSWS) is to maximize the efficiency of watering crops and minimize water wastage that occurs due to conventional agricultural methods. The IBSWS approaches both problems of soil pH and water conservation autonomously. The system monitors soil moisture on a consistent basis and waters crops if the soil moisture is low. This is similar to some current prototypes made in water conservation attempts which take sensor data and then water crops accordingly. However, IBSWS also keeps track of the pH of irrigation water and balances the pH according to the requirements of the specific crop. Therefore, soil pH is modified via the pH of the irrigation water. Since watering is necessary for soil pH modification, this system is most effective in environments without extensive precipitation. Additionally, the IBSWS is programmed with an app using the Blynk platform, which is an IoT-based platform that allows app programming for microcontroller-based systems. Blynk allows users to create their own apps to control hardware remotely and display sensor data. It also allows users to access the Blynk Cloud Server, through which information can be stored and communicated [8]. The IBSWS app is a critical component of the system because it allows the user to select the type of plant being watered, since different plants have different soil pH requirements. IBSWS also uses the app by communicating pH and moisture values to the app which are displayed to the user.

The remainder of this paper is organized as follows: Section II references related works, Section III presents the IBSWS Architecture, Section IV elaborates on the IBSWS data processing algorithm and the operational structure, Section V discusses the design and features of the IBSWS App created using Blynk, Section VI presents the Experimental Evaluation and Results, and finally Section VII presents conclusions and future directions.

II. RELATED WORKS

Various related works have tested the efficiency and feasibility of smart irrigation systems, giving insight into how IBSWS may be beneficial and can improve upon past research. In [9], the authors present a wireless sensor-based system that controls irrigation of crop fields by using data from soil moisture sensors (SMS). Sub-level nodes would work independently to gather SMS data from crop fields, and this data would be processed and transmitted to a master node using a NRF24L01 radio wave communication module.

978-1-7281-8420-3/20/$31.00 ©2020 IEEE

The authors conclude that the use of the NRF24L01 module leads to a large amount of data lost in communication between the nodes. The IBSWS hopes to advance on this with the use of an ESP-01 module, which allows for nodes/microcontrollers to connect to a Wi-Fi network. If the node has a good Wi-Fi connectivity, using the ESP-01 module (a module of ESP8266) will allow for communication at virtually any distance. In [5], one main difference from the IBSWS in the implementation of an irrigation prototype is that their system waited for a duration of every two hours to check the status of the plant. The implementation of this timer allows them to save energy by only checking at given intervals. However, the IBSWS was designed with priority given to real-time data collection so that the user can receive up-to-date information through the app. For this reason, the IBSWS model takes SMS readings of the plant much more frequently (every 5 seconds).

In [10], the authors compared the efficiency of using different types of data sensors (SMS, evapotranspiration, rain) to lead smart irrigation decisions. IBSWS uses only SMS to detect whether crops need watering, due to the fact that **a)** Several commercially available SMS controllers tested in [11] ranged from 69% to 92% water savings compared to a control irrigation system that does not use sensors, showing that there is great potential to save water using SMS-based systems and **b)** In [10], authors concluded that medium threshold SMS gave almost optimal results compared to evapotranspiration and rain sensor-based smart irrigation systems. This allowed for water savings and quality turfgrass [10], leading to a conclusion that SMS allows for efficient use of water while also sufficiently watering the plant. In [12], the authors proposed a system that uses humidity and temperature sensors, but not soil moisture sensors. By doing so, an accurate reading of the overall environment of the plant/crop is attained but not of the actual soil moisture content. The IBSWS differs from most other watering systems since it implements an accurate water pH manipulation system to ensure that crops have a suitable soil pH for nutrient absorption in addition to maintaining proper soil moisture.

III. IOT- BASED SMART WATERING SYSTEM ARCHITECTURE

IBSWS is designed to be able to control multiple crop fields to ensure use for farms and large agricultural lands.

As shown in Figure 1, IBSWS is composed of multiple subsystems, each controlling a single crop field to best irrigate each crop field independently and in a customized manner. Each subsystem is composed of two sub-level microcontrollers (AM and BM) and one Main Microcontroller (MM). In Figure 1, an example of two subsystems working together to irrigate two crop fields is shown, which can be expanded to multiple more fields.

Similar to the system proposed by authors in [9], multiple soil moisture sensors are utilized by the AM and BM microcontrollers (as shown in Figure 1) to measure the soil moisture data, which is used to efficiently water crop fields. Each subsystem also has its own main water reservoir, as well as pH solution reservoirs to mediate the pH of the main water reservoir. Using the IBSWS mobile app, users can

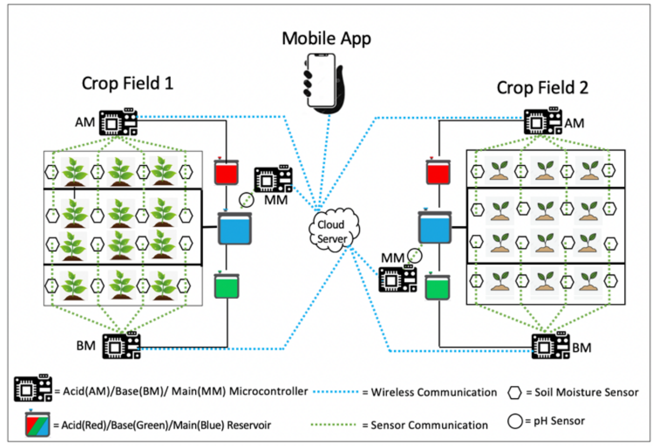

Figure 1. Physical Layout and Communication of the Large-Scale IBSWS Concept

identify the crop that they are growing and pH sensors and pumps are used to customize and modify the pH level of the water to be safe for the specific crop. Soil moisture data acquisition is done by AM and BM microcontrollers, while pH sensor data are gathered by the MM. All sensor data are eventually sent to the main microcontroller, where the data analysis is conducted. Based on the analysis of the MM, the correct actions are taken, and pumps/valves will be activated to irrigate the crops. All of the sensor data are also sent to the cloud, where it can be stored. Data stored in the cloud will be pulled and displayed by the IBSWS mobile app. The app allows for users to manage, control, and monitor the system and crop field by displaying sensor data.

Managerial Impact: IBSWS improves upon conventional agriculture irrigation systems by implementing several smart features which will involve the addition of new infrastructure. Specifically, to transition to a pH regulating irrigation system, each farm will require installing two tanks (referred to as reservoirs in the paper) to hold plant-friendly acidic and basic pH solutions. These tanks should implement an automatic system to replenish its supply of pH solution on a timely basis (daily, weekly) based on the tank size. For each crop field, a water tank is necessary to intake and store water from the regular water supply and utilize the pH mediation tanks to modify the pH of the water within using motors controlled by microcontrollers in order to adequately suit the plant that is being watered in that given crop field. To transition from a conventional irrigation system to IBSWS, the regular water supply must be directed into each water tank which is attributed to each crop field. In order to track the regulation of pH in these water tanks, a pH sensor must be installed in each tank. Another key feature in the IBSWS is the ability to water based on a measured soil moisture value obtained by using the average measurement of several soil moisture sensors, which must be installed in each crop field. Microcontroller devices should be installed and connected to pH and soil moisture sensors to communicate and take actions upon measured data. Motors to facilitate the flow of water from tanks into crop fields when necessary. The operation of the system can be controlled over IoT using the IBSWS mobile app installed in the farmer's mobile phone, as demonstrated by our prototype. The app allows for many features that help in managing and operating the technology described in section V. The diagnostic features of the IBSWS app decrease the effort spent by farmers to pinpoint problems with the IBSWS, allowing for faster and smarter maintenance. IBSWS maintenance is similar to maintaining a conventional irrigation system, the only differences being that the functionality of motors pumping water to and from the tanks must be checked, and the functionality and accuracy of the electronics (pH sensor, SMS, microcontrollers) must be assessed frequently.

A. IBSWS Prototype Physical Layout and Components

As shown in Figure 2, the prototype IBSWS built in this research has several main physical components. The prototype is effectively one subsystem of the complete system shown in Figure 1. Firstly, it has 3 microcontrollers: Base Microcontroller (BM), Acid Microcontroller (AM), and Main Microcontroller (MM). For the AM and BM microcontrollers Wemos R1D1 boards were used, which have several input/output pins as well as a built in ESP8266 Wi-Fi module for wireless communication. Capacitive soil moisture sensors are connected to AM and BM via an analog input pin. The BP and AP peristaltic pumps (motors used to pump fluids) are connected to the AM and BM via digital output pins.

Using the BP and AP, the AM and BM control the flow of plant friendly pH solutions from acid and base reservoirs respectively, to the main water reservoir from which the crop field is watered. This accurately balances the pH to be suitable for the crops. These microcontrollers are designated as either the BM or AM based on which pH solution they control. For the MM, an Arduino Mega was used, which contains many input/output pins and a separate ESP-01 Wi-Fi module was connected to it. Using these Wi-Fi modules, the microcontroller components of the system will be able to connect via the Blynk Cloud Server to the IBSWS app and with each other. The app can then control and monitor the operation of the system. The pH sensor is connected to the MM and is placed in the main reservoir. This helps the MM to keep track of the pH of the irrigation water. Finally, the water pump (WP), which is also a peristaltic pump, is connected to the MM to facilitate water flow from the main reservoir into the crop field. Each of the physical components are useful when they communicate data with each other. This communication allows the system to distribute sensor data and use this data to make decisions on the actions of the system. The following sections explain the mechanism by which the different components of the system communicate with each other while sharing data simultaneously.

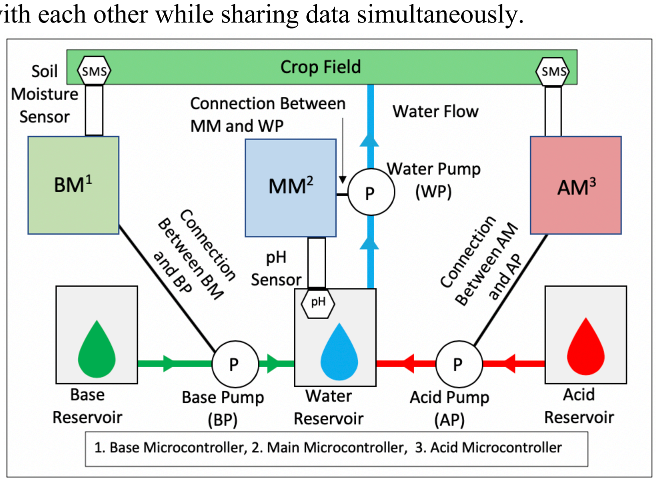

Figure 2. A physical layout of the components of the prototype IBSWS system

B. System Prototype Communication Architecture

This section explains the architecture of the IBSWS and how communication and data flows occur among the different components of the system. The communication done within the IBSWS takes place at 3 different layers: sensor communication, IBSWS app to microcontroller communication, and communication among microcontrollers. This section explains each of these communication layers and how they enable the IBSWS functionality.

1) Sensor Communication: The two types of sensors that are used in the IBSWS are a pH sensor (which has two components: a pH probe and a BNC connector) and two capacitive soil moisture sensors. The pH sensors and soil moisture sensors all work on a 5v power supply, which is applied and regulated by the microcontrollers (AM, BM, and MM). The pH sensor works by using the principle that having a high hydrogen ion concentration increases the potential to conduct electric current. The voltage difference is measured by the use of a measurement electrode and a reference electrode and this voltage difference is outputted by the pH sensor. The pH sensor outputs the voltage of the signal mapped out as integers from 0 to 1023. This value is then correlated to the pH of the water being measured. The soil moisture sensors are connected to the AM and BM modules. The soil moisture sensors are capacitive sensors, which means that they consist of two capacitor plates and function as described in [13]. The soil acts as a dielectric between the plates, which changes the capacitance. The addition of water is measured by calculating the change in water-dissolved ions within the soil water content. A voltage proportional to the capacitance is sent to the analog communication pin of the Wemos boards, which is then calibrated to indicate the relative amount of water with an integer output. The data output is calibrated such that an output of 1023 indicates dry air and an output of 0 indicates that the sensor is completely submerged water.

2) App to Microcontroller Communication: The communication between the IBSWS App and a microcontroller (AM, BM, or MM) happens through the Blynk Cloud Server [14]. To communicate via the internet, an ESP8266 ESP-01 WiFi module is connected to the MM using a connector and jumper wires. The MM is provided with the WiFi name and password, which is then passed onto the ESP-01 module and

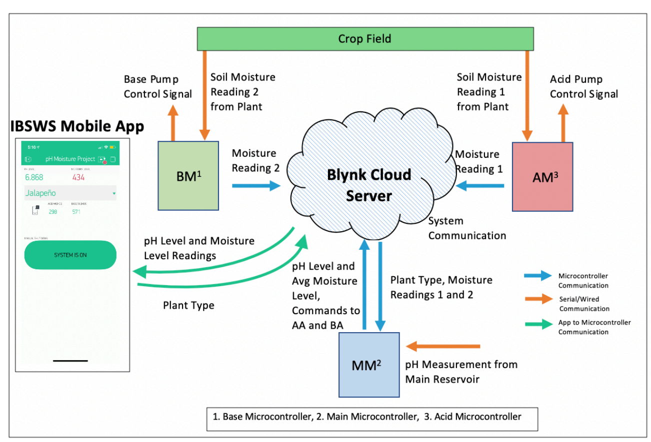

Figure 3. IBSWS Communication Architecture Diagram showing which components communicate with each other to share data

used to connect to the local WiFi. The AM and BM controller boards have ESP8266 modules integrated within them and operate the same way. The framework for communication from devices to the Blynk App is through an authentication token. When a hardware device (such as a MM or AM/BM) is added to a Blynk project, it is assigned an authentication token. This token is used as a tag to verify the location and exchange of data. In order to communicate data, it must first be stored in a virtual pin. The app or the microcontroller can send data to be stored in any virtual pin, and it can be read by the other. In order for the data to be transferred correctly, it is tagged with the authentication token of the device it is being sent from/to. In this way, virtual pins cannot be accessed by other microcontrollers and the microcontroller for each virtual pin must be specified on the app in order for data transfer to occur.

The pH value of the water reservoir and the computed average soil moisture are sent in this fashion to the app and are stored in virtual pins. These virtual pins are then linked to UI elements called “widgets” (such as displays and buttons), and these values can be displayed on the app. Values are continuously updated every few seconds so that information displayed represent current and not past conditions of soil and pH. The app also contains a dropdown menu from which the user can select which plant they are watering (for testing purposes, only three plants were added to the dropdown menu however any amount can be added for commercial use). Each plant type has an index associated with it (1, 2, 3, etc.) and the index is outputted to a virtual pin which can be read by the MM. Based on the plant type index sent by the app, the MM will establish the pH range that is deemed suitable for the plant. For example, when the user selects “Blueberry” as their plant type, the MM will identify the suitable pH range as 4.5-5.5 because blueberry plants require an acidic soil pH to grow. Each of these pH ranges for each plant is specified in the programming for the MM. In this way, two-way communication is directed between the app and the MM to update its calculations and actions to the needs of the user while also notifying the user about the status of the plant in real-time.

3) Communication among Microcontrollers: The IBSWS system prototype has two types of microcontrollers as shown in Figure 3: one MM and two AM/BM controllers. The objective of microcontroller communication is to be able to communicate sensor data and communicate commands for actions that must be taken based on the sensor data between the three microcontrollers. The method for communication between microcontrollers is similar to the method for communication between the app and a microcontroller. The Blynk app is given the authentication token for each device within the project. Using the authentication token, data is stored in virtual pins as detailed in 2.2. When the authentication token of one microcontroller device is shared with another device, a bridge between devices can be created. Each bridge has a virtual pin designated for storing data that travels between the two microcontrollers, however information is only sent one way. In this application, by adding two one-way bridges between each microcontroller and manually sharing the authentication tokens of each device, a two-way communication bridge via the Blynk Cloud Server is created between the MM and the AM/BM microcontrollers. When microcontrollers communicate, the data that is being sent is distributed to a specific virtual pin, from which the receiving device can take the data. Using this communication mechanism, the AM/BM microcontrollers gather SMS data and send it to the MM. The MM also sends commands based on pH sensor data to the AM and BM to alter the pH by pumping pH solutions from the acid or base reservoir into the main water reservoir using the BP and AP shown in Figure 2.

The sensor data and commands that are received from communication between different components must be utilized and processed. Using the data and commands that are communicated through the system, an algorithm is used to compute and process the data and commands that are received. The following section explains the data processing algorithm used by the IBSWS and how it makes decisions related to plant watering and pH balancing.

IV. Data Processing and Operational Algorithm

This section discusses the data processing algorithm for the IBSWS along with how the system operates by using the data processing algorithm and how it uses this data to take the appropriate actions. The algorithm runs in a continuous loop for a set amount of time that can be modified.

In step 1 of Figure 4, the system checks the type of plant that has been inputted by the user on the dropdown menu of the IBSWS app. This plant type is communicated to the MM. In step 2, data from the pH sensor is read and stored by the MM. The code snippet below shows how pH sensor data are processed in the system: As shown in Figure 5, Using the analogRead() function in the Arduino Library, a voltage from 0-5v, which is mapped out to an integer number from 0-1023, is read through the analog pin which is connected to the pH sensor. Ten readings of the analog pin are stored in an integer array, and these readings are then sorted using a bubble sort algorithm. Once they are sorted, the data points in the first two and last two positions of the array (the two least and greatest values) are excluded from the average calculation. This is done to avoid extraneous data that could be gathered.

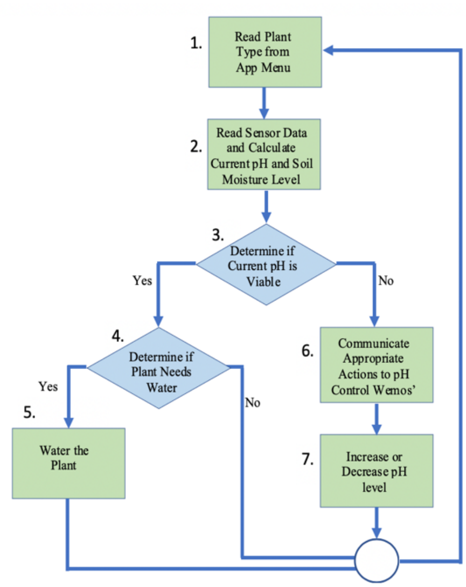

Figure 4. IBSWS Data Processing and Operational Algorithm

```
for(int i=0;i<10;i++)     //Get 10 sample value from the sensor for smooth the value
{
buf[i]=analogRead(SensorPin);
delay(10);
}
for(int i=0;i<9;i++)      //sort the analog from small to large
{
for(int j=i+1;j<10;j++)
{
  if(buf[i]>buf[j])
  {
    temp=buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
  }
}
}
avgValue=0;
for(int i=2;i<8;i++)                     //take the average value of 6 center sample
avgValue+=buf[i];
float voltage =(float)avgValue*5.0/1024/6; //convert the analog into millivolt
phValue = (-5.059)*voltage+19.38;      //convert the millivolt into pH value
```

Figure 5. A code snippet showing the data processing from the voltage reading from the pin which is translated into the pH value

The other six values are then averaged to come up with an average value. This value can then be mapped linearly to a pH value through the use of a pH calibration equation. The equation was created using the output values read from solutions with known pH. Using the output of the pH sensor for three known pH solutions, a linear equation was created by graphing the data, and this equation is used in the program to find the pH value.

Also using the analogRead() function for the analog pin that is connected to the soil moisture sensors, the soil moisture is computed. SMS data is obtained by the AM and BM (each unit has one SMS, which gives a total of two readings). The data is once again read as a voltage from 0-5v which is then mapped out to an integer number between 0 and 1023 (0 being completely wet and 1023 being completely dry). The data from the AM and BM units are sent over the Blynk cloud to the MM, where the two soil moisture readings are averaged to compute the average soil moisture reading.

For each plant type, ranges of pH are designated as safe and unsafe, and in step 3 the stored pH level is compared to these ranges to decide whether the current pH of water in the reservoir is safe for the plant. If it is unsafe, the MM will move on to step 6 and send an integer value of 1 or 2 to the appropriate sublevel microcontroller (AM or BM) to regulate the pH value. For example, if the pH level of the water in the reservoir is too acidic (low) for the current plant type, then the MM will communicate an integer value of 2 to the Base Microcontroller (BM). On receiving this value, the BM will send a signal to the BP pump connected to it to add basic solution from the base reservoir into the main reservoir in order to raise the pH value. Step 7 is the change in pH of the water reservoir that is a result of the BM or AM adding pH solution into the main reservoir.

If the pH is safe, the MM then sends an integer value of 1 to both pH control units to stop any pumping of pH solution into the reservoir, and then it will go on to analyze the soil moisture sensor data in step 4. This average moisture level is compared to a threshold that is programmed into the MM in order to determine if the plant needs watering. If the soil moisture integer is greater than the threshold, the plant is watered in step 5 using water from the reservoir by using the WP peristaltic pump controlled by the MM. This is done by sending a signal to the digital output pin connected to the WP. After the decision on whether the plant should be watered, the correct action is executed and the algorithm runs in an infinite loop to the beginning once new data points have been collected by the soil moisture and pH sensors. The response time for execution of the aforementioned actions is less than 5 seconds.

V. Mobile App Interface and Features

The IBSWS app interface was intended as a method by which users could more easily interact and manage the IBSWS. By using the app, users get the advantage of having full control of the system using the many features of the application. As shown in Figure 6, the IBSWS app interface has several features that help the user navigate the system. The Blynk App provides tools and mechanisms to create new customized apps to use the Blynk Cloud System [8][14]. The IBSWS Mobile App was created using widgets from the Blynk App [8], which can be programmed to work accordingly as desired for each Blynk Project. The following describes the features of the IBSWS app.

pH and Soil Moisture Display: Firstly, there are two main display widgets on the app which are the pH level display and the moisture level display. As shown in Figure 6, they display the current pH level in the main reservoir and the average soil moisture level of the plant's soil in real time, as well as individual SMS readings from the AM and BM. It does this by receiving the pH level and average soil moisture level from the MM and individual soil moisture levels from the AM and BM using the mechanism described in section 2. As seen in Figure 6, the individual soil moisture sensor readings can be quite different, which is why many SMS are needed to get an accurate average in a large-scale application.

Plant Type Selection: Secondly there is a dropdown menu as seen in Figure 6, and a zoomed in image is provided of when the dropdown image is selected. This dropdown menu allows for the user to change the type of plant that is recognized by the system. When users change the type of plant that they are watering using the system, they can use this dropdown menu to allow for customized watering for each kind of plant by selecting the new plant that they are watering. For several different plants, the system has ranges of pH that are programmed in to identify whether the pH of the water reservoir is safe for each plant. This can be useful for home gardeners who change the plants that they plant in an area each season, or for farmers who use crop rotation methods and need to change the type of plant they use in a crop field every growing season.

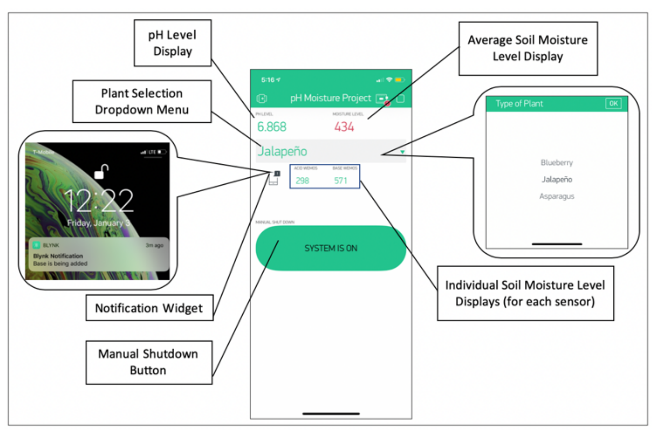

Figure 6. Main app interface with all displays and features labeled

The dropdown menu supports versatility and variability in the types of plant that are grown. The optimal pH range values for certain plants are taken from [15].

System Notification: To the left of the individual soil moisture reading displays is the notification widget. Although the widget has no critical function, it shows that the app is capable of sending notifications. In this application, the app was set up to send notifications whenever the plant was being watered, or whenever the pH in the main reservoir was being changed (by either adding acidic or basic pH solution). On the left in Figure 6, an example of the user being notified can be seen when the basic solution is being added to the main reservoir.

Manual Restart and Safety Features: A system was also set up to count the consecutive amount of times the algorithm (as described section 3) ended in adding pH solution into the reservoir. If added more than six times consecutively, the system would stop automatically and use a notification to ask the user to check for errors. This is to prevent irrigation of plants using unsafe water.

Furthermore, the manual shutdown button is shown at the bottom of Figure 6. If the user feels as if the system is not working properly, they will be able to turn the system off with a click of a button. This allows for the user to be sure that even if the system malfunctions or the pH sensor reads incorrect data, they will have full control of the system and be able to stop it before any damage is done.

VI. Experimental Evaluation and Results

Figure 7 shows the prototype system created by the authors in the process of watering a Jalapeño plant. The prototype system was built as described in section II(A). The system was tested under various conditions in order to test its features and effectiveness. The moisture control portion of the system was tested with 2 conditions, dry or moist.

Average soil moisture sensor readings are given on a scale of 0 (completely wet/water) to 1023 (completely dry) as described in section IV. The dry condition was classified as any value above 675, and moist as any value below 675. The system was programmed to have the same moisture threshold (675) for all of the tested plants. The pH solutions used for the acid and base reservoirs are as follows: The basic solution kept in the base reservoir for this experiment that was used was a bottled water called *Essentia*, which has a pH of 9. The acidic solution used was lemon water, which had an approximate pH of 2.

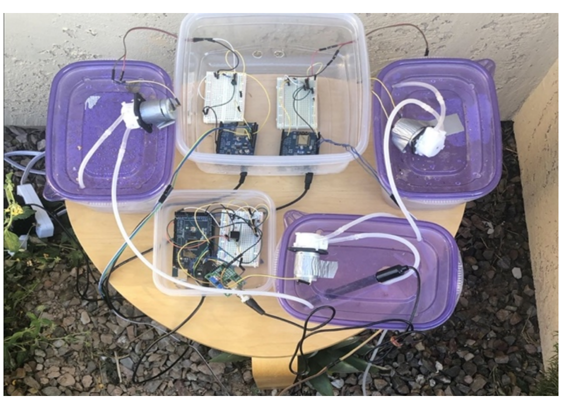

Figure 7. Picture of the IBSWS prototype system

4 trials were conducted: 3 trials were with 3 different plant type settings which included Primrose, Asparagus, and Blueberry plant types and all of the initial moisture levels were approximately 1023 (dry); the fourth trial was with the Primrose plant type setting, which acted as a control for the moisture experiment as it started with an initial moisture level of approximately 550 (already under the 675 threshold indicating that it was wet). These plants were not grown in this experiment; however the app was put on the settings created for the three plants to simulate watering these specific crops. In reality the system was watering plain potting soil. These three distinct plant settings were chosen primarily due to their distinct pH requirements: Blueberry plants require an acidic pH (5-5.5), Asparagus plants require a basic pH (between 7.5-8), and Primrose plants require more of a neutral pH (6.2-6.9). These ranges were taken from [15] and exaggerated for this experiment to be able to interpret data and results more distinctly between plant types.

As expected, we can see in Figure 8 that when tracking the change in soil moisture over time when the IBSWS is in use, in the trials that started with dry moisture levels (Blueberry, Asparagus, and Primrose Dry), the moisture level dropped under the 675 threshold and stabilized as a result of water being pumped by the main pump from the main reservoir. The control trial called "Primrose Control" shows that because the detected soil moisture level was already under 675, it was not watered further. Therefore, the soil moisture stays constant.

In several of the trials, before watering occurred the pH of the main reservoir needed to be modified. As seen in Figure 9, the Asparagus, Primrose Dry, and Blueberry trials had initial pH values out of the programmed safe range. Each of these pH values are modified as time goes on and the system operates, until they each stabilize at a pH level that is safe for the plant (shown in figure 8). Due to the different safe pH ranges for specific crops, different crops will have different responses to pH modification (increase or decrease in pH). For example, the Blueberry initial pH is at about a value of 7.2. This value decreases over time until it goes below 5.5, which was denoted as the upper limit of the safe range for Blueberry. This is why we see a delayed period of time at the beginning in Figure 8 for these three trials where the soil moisture level is not decreasing even though the soil moisture level is high. After time has been taken to modify pH, only then is the plant watered. For example, the test for Blueberry shows a waiting period of over 400 seconds before watering.

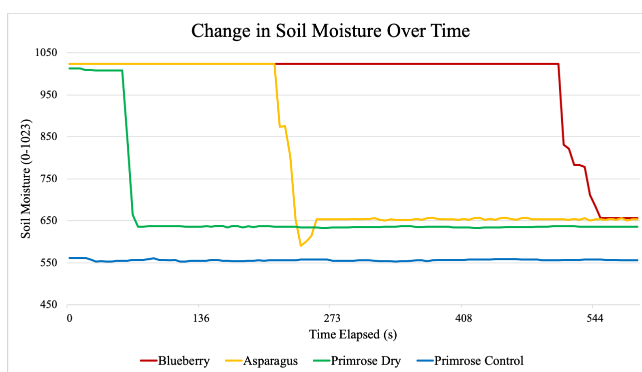

Figure 8. Graph of soil moisture of plant over time using IBSWS

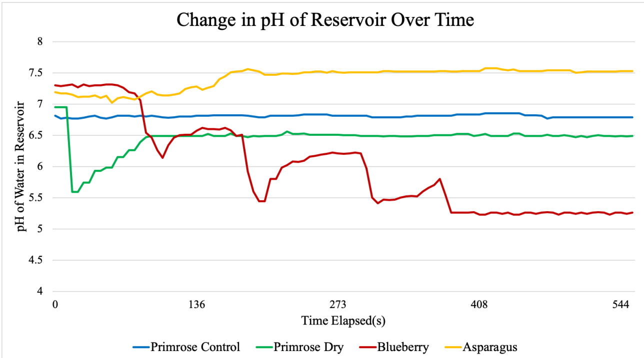

Figure 9. The pH level of the main reservoir tracked over time using IBSWS.

Figure 8 confirms this because it took over 400 seconds for the pH level of the main reservoir to level out at a safe value. We can also see that in the data for Primrose Control, because the initial pH level was already within the safe range, the pH was not modified. The rate of pH change differs between increasing and decreasing pH modifications as the acidic solution was more potent than the basic solution. This is seen in the rapid initial drop of pH in the primrose dry trial as well as the several rapid drops in pH in the blueberry trial. In contrast, the increase in pH is seen in the asparagus and primrose dry trials due to the addition of the basic solution displays a linear increase in pH.

VII. Conclusions and Future Work

The IoT-Based Smart System (IBSWS) is an innovation in the step towards smart irrigation to increase plant health and conserve water by implementing IoT concepts using sensors and microcontrollers over a network to facilitate communication and efficient action. With IoT as its backbone, the IBSWS uses soil moisture sensors to accurately measure and compare the moisture of the soil to a threshold value to ensure that the soil is watered only when the plant needs it. Additionally, if the pH of the water being given to the soil is unsafe for the plant the IBSWS uses the Blynk platform on IoT to communicate to sub-level microcontrollers which use plant-healthy acid and base pH solutions to modify the pH of the water being given to the plant. The system can be managed by the user via the IBSWS App created using Blynk, which allows the user to select the type of plant that they are watering so that the system can accurately manipulate the pH of the water given to the plant. The app can also be used by the user to monitor the pH and moisture levels monitored by the system. From the results of the experimental evaluation it can be concluded that such a system is possible for use in autonomously and efficiently watering specific crops. Overall, the system acts as an efficient method to conserve water by reducing human error and increase efficiency in terms of large-scale agriculture.

In the future, the authors hope to conduct a long-term test on several different plants to monitor the true percent water savings that can be saved by using the IBSWS, while also tracking the effect of using IBSWS on plant health. This would be done by using spectrophotometric analysis of plant leaves. In this process, chlorophyll pigments from cut leaves are dissolved in Dimethyl Sulfoxide (DMSO). The spectrophotometric results and absorption levels of the dissolved solution at certain light wavelengths can be used to determine the health of the plant [16] through chlorophyll density, an indicator of plant health [17].

Acknowledgment

The authors would like to acknowledge Mrs. Debbie Nipar at Hamilton High School for her help and guidance for this research. The authors would also like to acknowledge Professor Nicole Herbots at the Department of Physics at Arizona State University for her guidance in conducting research and scientific writing.

References

[1] G. Devanand Kumar, B. Vidheya Raju, and Durgesh Nandan, “A Review on the Smart Irrigation System,” In Proc. 2nd Intl. Conf. on Recent Innovative Trends in Comp. Science and Applications ‘19, 2019,
[2] A. Chalimov, “IoT in Agriculture: 5 Technology Use Cases for Smart Farming,” Jun. 7, 2018. [Online’ Available: https://easternpeak.com/blog/iot-in-agriculture-5-technology-use-cases-for-smart-farming-and-4-challenges-to-consider/, [Accessed: May 21, 2020].
[3] USDA ERS “Irrigation & Water Use,” *USDA ERS*, 23-Sep-2019. [Online]. Available: https://www.ers.usda.gov/topics/farm-practices-management/irrigation-water-use/. [Accessed: 29-May-2020].
[4] “WATER RESOURCE ISSUES AND AGRICULTURE,” *Food and Agriculture Organization*, 1993 [Online], Available: http://www.fao.org/3/t0800e/t0800e0a.htm [Accessed May 20, 2020].
[5]A. H. Abbas, et al., "Smart watering system for gardens using wireless sensor networks," 2014 Intl. Conf. on Engineering and Technology (ICET), Cairo, 2014, pp. 1-5
[6] United States Botanic Garden, *Soil pH Science Page*, Washington, DC: Office of Headquarters Operation, 2001.
[7] R. McLaughlin, “Will the pH of Water Affect the pH of the Soil?”, Oct. 7, 2016. Available: https://homeguides.sfgate.com/ph-water-affect-ph-soil-74237.html, [Accessed May 27, 2020].
[8] Blynk, “Blynk,” [Online]. Available: https://docs.blynk.cc/. [Accessed: 28-May-2020].
[9] Radi et al. "Design of Wireless Sensor Network (WSN) with RF Module for Smart Irrigation System in Large," 2018 Intl. Conf. on Computer Engineering, Network and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2018, pp. 181-185
[10] M.S. McCready, M.D. Dukes, G.L. Miller, “Water Conservation potential of smart irrigation controllers on St. Augustinegrass,” *Agriculture Water Management*, Volume 96, Issue 11, November 2009, Pages 1623-1632 [Abstract], Available: Science Direct, https://www.sciencedirect.com/science/article/pii/S0378377409001784 [Accessed May 27, 2020]
[11] B. Cardenas-Lailhacar; M. D. Dukes, and G L. Miller, “Sensor-Based Automation of Irrigation on Bermudagrass during Dry Weather Conditions,” *Journal of Irrigation and Drainage Engineering*, vol. 136, Available: ASCE library: https://ascelibrary.org/doi/abs/10.1061/(ASCE)IR.1943-4774.0000153 [Accessed: May 20, 2020].
[12] S. R. Prathibha, A. Hongal and M. P. Jyothi, "IOT Based Monitoring System in Smart Agriculture," 2017 Intl. Conf. on Recent Advances in Electronics and Communication Technology, Bangalore, 2017, pp. 81-84
[13] M. M. Galgalikar and G.S. Deshmukh, “Real-Time Automization of Irrigation System for Social Modernization of Indian Agriculture System,” *Intl. Journal of Computer Applications* Vol 1, No. 22, pp. 0975 - 8887, 2010.
[14] H. Durani et al., "Smart Automated Home Application using IoT with Blynk App," *2018* Second Intl. Conf. on Inventive Communication and Computational Technologies, Coimbatore, 2018, pp. 393-397
[15] C. Boeckmann, “Soil pH for Plants, *almanac.com,* Aug. 13, 2019, Available: https://www.almanac.com/plant-ph [Accessed May 21,2020].
[16] J. D. Hiscox and G. F. Israelstam, “A method for the extraction of chlorophyll from leaf tissue without maceration,” *Canadian Journal of Botany*, vol. 57, no. 12, pp. 1332–1334, 1979.
[17] Y. Li et al. , “Factors Influencing Leaf Chlorophyll Content in Natural Forests at the Biome Scale,” *Frontiers in Ecology and Evolution*, vol. 6, Jun. 2018.