DEVELOPING A PLANT HEALTH MONITORING, MANAGEMENT AND AUTONOMOUS WATERING SYSTEM USING IOT

Anita Joshi
Department of Engineering, Sciences and Humanities Vishwakarma Institute of Technology
Pune, India
anita.joshi@vit.edu

Arya Pimpalgaonkar
Department of Engineering Sciences and Humanities Vishwakarma Institute of Technology
Pune, India
abhir.arya21@vit.edu

Arman Nandeshwar
Department of Engineering, Sciences and Humanities Vishwakarma Institute of Technology
Pune, India
rajesh.arman21@vit.edu

Arpit Mahalle
Department of Engineering, Sciences and Humanities Vishwakarma Institute of Technology Pune, India
arpit.mahalle21@vit.edu

Arnav Dhobale
Department of Engineering, Science and Humanities Vishwakarma Institute of Technology
Pune, India
dhobale.arnav21@vit.edu

Arwa Bhojawala
Department of Engineering, Science and Humanities Vishwakarma Institute of Technology
Pune, India
arwa.bhojawala21@vit.edu

Manish Arote
Department of Engineering, Science and Humanities Vishwakarma Institute of Technology
Pune, India
manish.arote21@vit.edu

***Abstract*—In today's busy times, everyone aspires to do more but can manage less. This is clearly seen in the modern trends in gardening with people forgetting to take care of their plants once they have planted them. This results in a general unenthusiasm in maintaining plants and gardens in today's youth, but modern problems require modern solutions. Therefore, we propose a modular system which will automate the most crucial of all processes involved in plant maintenance: Irrigation. The system automatically monitors the soil for moisture levels and irrigates whenever required, or according to a preset schedule which the user can program through a simple android app. The scope of this low power solution is not only limited to small home gardens but can also be scaled up to suit more efficient and automated management of large public gardens and even full-fledged smart agricultural farms with just a few upgrades. The provisions of such upgrades have been made in this solution.**

***Keywords*—automatic watering, Bluetooth, DIY, gardening, green beautification, green movement, home automation, home gardening, irrigation, smart agriculture, smart appliance, smart city, smart greenhouse, soil moisture, IoT**

I. INTRODUCTION

With rising population, there is a need for increased agricultural production. To help produce large amount of production the need of fresh water also arises. As of now, the agriculture sector consumes almost 83% of the total amount of water consumption. The use of IoT in this field will be helpful to reduce the wastage of water. It gives an important message that it is the need of the hour to develop such systems which will save water without putting pressure on farmers. Apart from agricultural farms, plants are widely grown in houses and offices. Due to busy lifestyles, people tend to neglect their plants and they end up drying.

The unique point of our project is that it is midway between a traditional irrigation system and a complete autonomous plant maintenance system. People also require a simple automated watering system (without all the costly sensors used in full-fledged maintenance systems which are practically not useful e.g., ambient humidity sensor, temperature sensor, luminous intensity sensor, soil composition sensor, etc.). Agriculture is one such sector in which a lot of research work is done. Most projects signify the use of wireless sensor network which collects data from different sensors deployed at various nodes and sends it through a wireless protocol. The collected data provides information about the various environmental factors. It proposes a low cost and efficient wireless sensor network technique to acquire the soil moisture and temperature from various locations of farm and as per the need of crops if the irrigation is required or not.

Through this paper, we intend to provide a solution to this highly polarized market trend and offer users as well as developers a median between the highly sophisticated, complex and specialized plant management systems which mainly focuses on commercial application, and the fundamental, immutable, schedule-based irrigation systems.

II. METHODOLOGY

A. Components

The following components were used:

i. Arduino UNO R3 development board
ii. Soil moisture sensor
iii. HC-05 Bluetooth module
iv. SPDT Relay
v. Mini 9V submersible water pump
vi. Water pipes
vii. Jumper wires
viii. Plastic housing
ix. Android smartphone

B. *Algorithm*

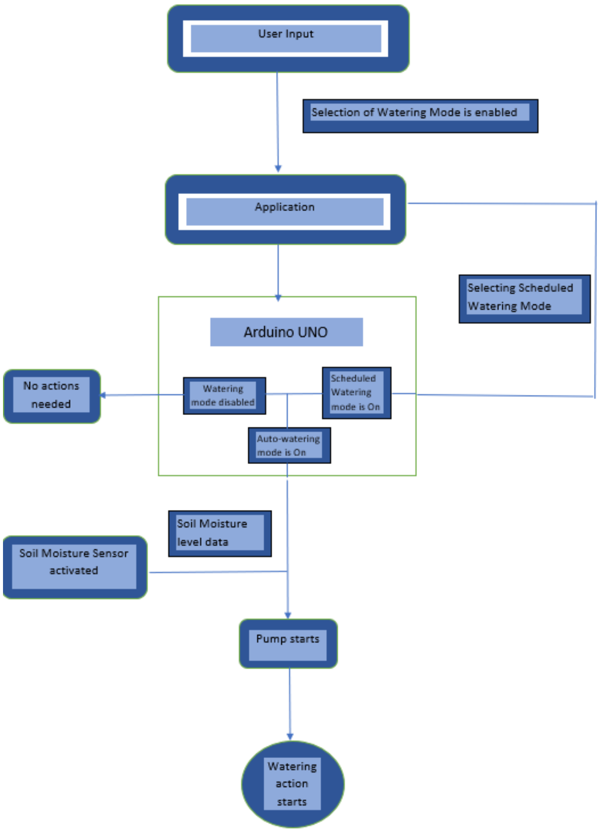

Fig. 1. Flowchart of Watering System using an application.

III. RESULT AND DISCUSSION

The following discussion is structured into 5 sections according to the fundamental heads of our project: App, Module (actuator device), Communication, Calibration, and Testing and debugging.

A. *Android application*

We developed the app for our project using the MIT App Inventor online tool. It is a website developed by students from the Massachusetts Institute of Technology, Boston which offers handy and easy-to-use tools for developing apps. It has a basic graphics designer and a code-blocks based programming section/tool.

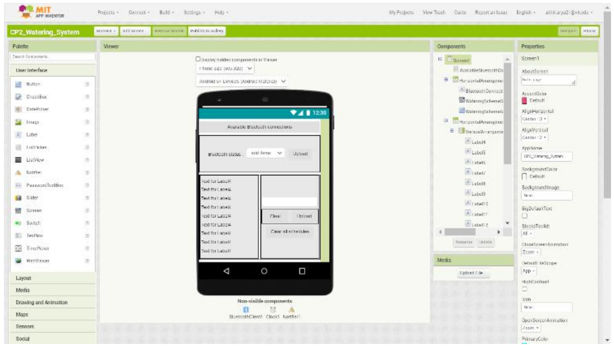

Fig. 2. MIT App Inventor project page (designer)

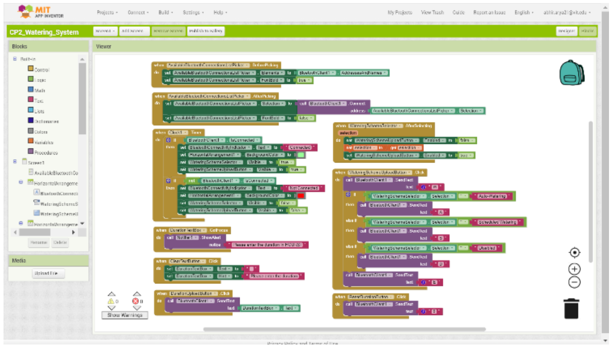

Fig. 3. MIT App Inventor project page (code blocks)

Our aim was simple and clear. We had to develop an app that performs the following functions:

1. Connect to the module via Bluetooth
2. Allow users to select the watering mode and send this information to the module via Bluetooth
3. Allow users to select the schedule of watering while using scheduled-watering mode and send this data to the module via Bluetooth
4. Receive the acquired sensor data from the module and display this on the application screen for users to see.

Using this tool from MIT, we have developed a single-page app consists of 3 primary sections: a Bluetooth connectivity management and watering-mode select section, a schedule-setting section to be used in ‘Scheduled-watering’ mode, and a section for displaying the sensor data used to monitor the plant health in anticipation of future development on those lines.

So let us consider the first section of our application. It consists of an indicator that displays whether Bluetooth connectivity with the module has been established or not. A list picker has been provided for selecting a Bluetooth device to connect to. Users can browse through the list of available connections and find their module here. A spinner along with a button become visible once connectivity has been established. The spinner is for selecting the watering mode and the button is for sending this data through to the module. The spinner offers 3 options to the users: ‘Auto-Watering’, ‘Scheduled watering’, and ‘Disabled’.

The second section of our application is for setting the schedule in case of 'Scheduled watering'. Users would find a text box asking for input and 3 buttons underneath: 'Clear', 'Upload', and 'Clear all schedules'. The app directs users to enter the number of hours after which the module should water the plant again into the text box and then send this to their module by clicking on the 'Upload' button. The user may clear the text box by clicking on the 'Clear' button, and clear the data stored in their module regarding any previously uploaded schedule by clicking on the 'Clear all schedules' button.

The third section of our application is an area reserved for displaying sensor data acquired by the module from the plant's surroundings. Since we have not yet used any plant health sensor, this section is currently empty. We look forward to developing this further.

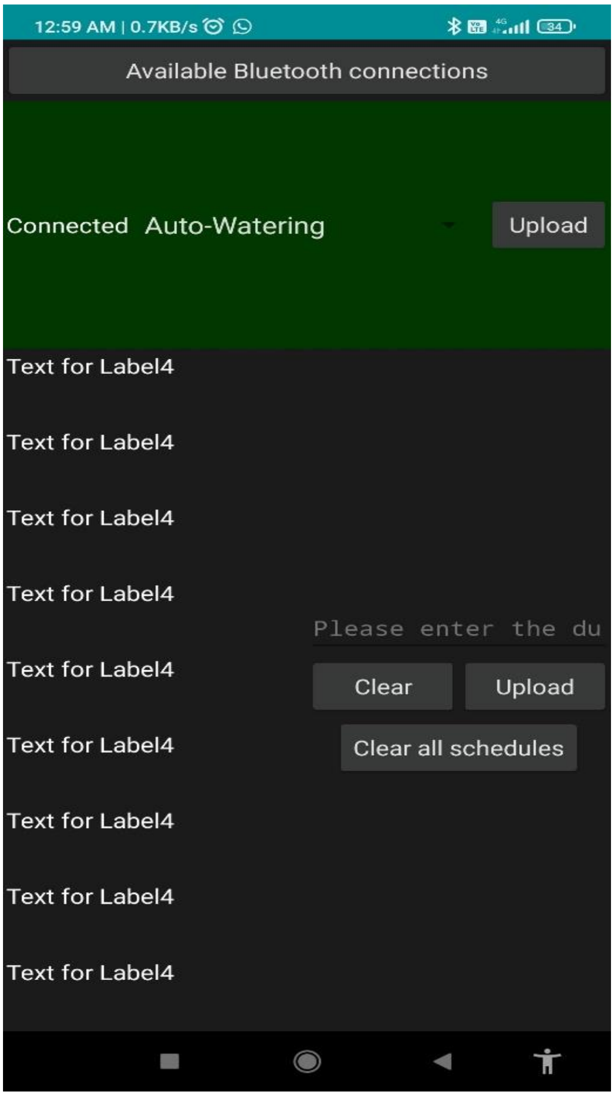

Fig. 4. Screenshot of the mobile application

B. Module

The module consists of a water-tight box made up of plastic containing all the core electronic components (we shall be referring to it as the 'main pod' throughout the course of this paper), with another plastic box-like water tank with a submersible water-pump in it. This assembly is mounted over the edge of the plant's pot. We have used an Arduino UNO R3 development board for this project. The water pump is connected to the Arduino through an SPDT relay, and the water is directed to flow to the base of the potted plant through a small pipe leading through the output nozzle of the pump. Another wire exits the main pod connecting the Arduino to the soil moisture sensor which is inserted into the soil near the base of the plant as close to its roots as possible without harming them.

The main pod houses the Arduino, Bluetooth module, SPDT relay, and the battery. Let us consider each component:

1. *Soil Moisture sensor*: It detects the moisture level of the soil when inserted into it by measuring the resistance between its two electrodes. It has 4 pins: VCC, GND, AO, and DO. VCC and GND are for supplying power and AO gives analog output, while DO gives a quantified digital output. We have only used the analog output pin for this project. The VCC and GND are connected to an analog pin and ground pin of the Arduino respectively, and the AO pin is connected to another analog input pin of the Arduino. The reason we have connected the VCC to an analog pin is to have the ability to power the sensor only when required.
2. *Bluetooth module*: We have used an HC-05 Bluetooth module in our project for enabling communication between the smartphone application and the effector module. The HC-05 module has 6 pins of which we only need to connect 4 for the purpose of our project: VCC, GND, Tx, and Rx. VCC and GND are power pins, while Tx and Rx facilitate serial communication. VCC and GND are connected to the 5V output power pin and ground pins of the Arduino respectively while the Tx and Rx pins are connected to the Rx and Tx pins of Arduino's serial port respectively.
3. *SPDT relay*: The water pump requires 9V for proper functioning, while the digital high level of the Arduino digital I/O pins is 5V. Hence this necessitated the use of an SPDT relay to separate the pump's power circuit from the overall control circuit.
4. *Battery / power source*: Arduino UNO R3 has an onboard voltage regulator which allows a range of 7V – 12V (recommended) or 6V – 20V (limit). Meanwhile the pump requires 9V. Thus, we have used a 9V dry battery to power the entire module. This setup is also compatible with using a 12V lead-acid battery for longer battery life. This aspect of the project is quite flexible.

The module performs the effector functions of watering the plant appropriately as per selected mode, receiving information from the Bluetooth module and interfacing with the sensors. The flow of operation can be characterized as such:

1. Anticipating information regarding watering modes from the user via Bluetooth (application on the smartphone),
2. Accordingly, setting variables that store the current state of watering mode,
3. Redirecting control to appropriate function defined for each mode,
4. Constant monitoring of soil moisture data if auto-watering mode is selected.
5. Keeping track of time and the schedule if scheduled-watering mode is selected,
6. Anticipating information regarding set schedule from user via Bluetooth.
7. Disabling and turning off the motor as well as sensor in case of disabled mode selected.

C. Communication

All wireless communication and subsequent IoT generated in our project is in the form of a basic Bluetooth connectivity between the module and the users' smartphone.
The HC-05 Bluetooth module outputs data using serial communication protocol. Thus, the data is received serially, and one byte is transferred at a time. This byte could be any alphanumerical character. The Arduino reads this data in the form of characters or strings. Thus, we can use the Serial.read() function for reading a single character or Serial.readString() function for reading a full string of received characters from the serial buffer.

Now, there are two types of messages that need to be transferred between the app and the module: Watering mode selection and schedule setting data.

After selecting the watering mode from the spinner, the user needs to click on the Upload button to send this message to the module over Bluetooth. The watering mode message is encoded in the following way: Auto-watering is encoded as 'WAE' standing for Watering mode message – Auto-watering – End of message, similarly, Scheduled watering is encoded as 'WSE', and Disabled as 'WDE'. This form of encoding will come in handy when using multiple modules in the future where the messages should not interfere with each other.

Like the watering mode, there is a text box as well as an upload button provided for specifying the number of hours after which to water the plant again in scheduled watering mode. This message is encoded simply by sending the number of hours over Bluetooth i.e., if the user wants to water the plant once a day, '24' hours can be entered. This message would be sent in this format itself: '24'.

Now these messages are received by the module. The Arduino receives these messages received by the HC-05 through its serial port. The timing at which the Arduino reads this data is very crucial. In the code, receiving the Bluetooth data is the very first thing that is done in the loop() function. The data from the serial buffer is stored in a variable and is checked to see whether it is a number. If it is indeed a number, then this means that the message received is the watering schedule and is subsequently stored in another variable used in the watering mode function. Otherwise, the message received is to specify the mode and is thus, stored in a variable used to select the watering mode.

D. Calibration

After the project was working practically, it needed to be calibrated in a lot of parameters, namely:

1. Setting the threshold moisture level to be such that it would not be continuously watering, yet also not allowed to get too dry,
2. Placing the moisture sensor at the correct depth and location inside the pot,
3. Setting the time for which to keep the pump on when in scheduled watering mode.

This was done by testing it out in the field and in actual use-cases.

E. Testing and debugging

We conducted testing in the form of prototypes. We isolated each subsystem of operation and built a prototype starting with the most basic operation and onwards. The next prototype would have the previous tested operation and the next operation added to it. In such a way, each prototype would be an incremental cumulation of the subsystems until all subsystems are added, i.e., the final product is made. This also happened to align with the design thinking process implemented throughout the project. Following are the prototypes: -

I. Along the lines of our objectives, the first and most fundamental function that must be tested is the automatic watering system which would water the plant according to the soil moisture. So, the first prototype was a simulation consisting of an Arduino connected to a soil moisture sensor and pump through a relay. The desired function was that the pump must turn on when the detected moisture drops below a threshold value and turns off when the moisture returns to threshold level.

II. After successful runs of the first prototype, we realized a problem. Not all plants need, nor can they sustain a constant, continuously maintained moisture level in their surrounding soil. The solution to this problem is the traditional method of watering in which plants are watered at some particular time of the day, week or month depending on the type of plant. We wanted to give our users this comfort of being able to set a schedule according to which the plants would be watered, but without having to physically go to the plant. This meant that the commands should be communicated wirelessly. This meant the involvement of Bluetooth and a smart phone application. Hence, the 2nd prototype tested the application, the Bluetooth communication, and the 'Scheduled watering' mode. The app and the mode worked perfectly, but the Bluetooth communication required a rework of the timing and contents of the messages which were sent and received. Another problem faced was with using the timer to keep track of the schedule. We have used the millis() function to keep time and the number of milliseconds naturally tend to be very large numbers when working with large amounts of time. This necessitated the numbers to be stored in unsigned long integer data type rather than the integer data type.

III. After the success of the 2nd prototype, we noticed the need of a way to turn off the module without disconnecting the power or even having to physically go to the module. Aligned with this necessity, we developed the 'Disabled' mode in which all watering functions are disabled. The module only receives and interprets the Bluetooth messages in this mode. With the integration of a 3rd mode, the logic used to store the currently selected mode through each iteration of the loop() function had to be changed. The new system involved the use of Boolean variables to store the status of mode selection and redirected the control to appropriate functions of each mode. This caused a problem for the soil moisture sensor since we turn the sensor on and off with each measurement of the sensor i.e., each iteration of the loop. Such a high frequency switching of the sensor does not give stable readings. A local delay of 1 second added to the function of the code responsible for reading the sensor data solved this problem. The demerit of this solution is the increased probability of missing a Bluetooth command received during the delay time. This could be solved by adding a confirmation system which sends a message back to the user when the given command is received by the module.

IV. The final prototype included all the subsystems working in tandem with each other. This also involved the hardware and form factor of the module. In this prototype, the actual working of the module was tested with a plant. Now, until the last prototype, the actuator was only tested with an LED glowing in place of the motor turning on. In this phase, even the motor was added (through an SPDT relay). This revealed a problem in the power management. The battery used was not able to sustain the power requirements of the motor added to the Arduino. This problem was solved by adding separate batteries for the power circuit and control circuit. This also gives the advantage of a longer battery life. In the future, a battery eliminating circuit (BEC) might be better suited. The design of the hardware and module was developed to be waterproof, rugged, and cost effective. Mounting was a tricky aspect since it is subject to the users' setup, but a general solution was proposed where the module hangs off the edge of the pot, while the water tank can either do the same or can be placed on the nearest sturdy surface with appropriate adjustments in water pipe and wire lengths.

IV. FUTURE SCOPE

More modifications can be made to enhance the functionality of the system, such as setting a delay for the next time the moisture is checked instead of continuously checking it. We can also set a higher threshold of moisture while watering is going on, then reduce the threshold to normal once watering is complete up to the higher threshold. Different moisture thresholds and different schedules during different seasons according to plant specifics can be added. Also, for the convenience of the users, options like 'date' and 'time' can be set.

Developers can generate cloud database of ideal conditions for corresponding plants and get appropriate data from the cloud database according to the plant selected by them, thus setting appropriate parameters for functions.

This system has the potential to be used for plant research by uploading user's plant health data acquired from sensors to cloud and corresponding these data with the plant's surrounding environmental conditions' data and finding the trends of which plant finds which conditions desirable.

We can use Wi-Fi and host a web location to control these modules from anywhere in the world, over the internet and thus have the ability to control multiple modules such as in a big farm, real time and simultaneously.

We can also attach larger motors and interconnected modules for large scale farming coupled with control software which manages watering plans in complex farm layouts.

V. CONCLUSION

Automatic Plant Health Monitoring System is a system developed for monitoring and checking the health status of the plant. The system checks the moisture levels of soil and irrigates whenever required thus providing the moisture required by plants. An Android application is developed for the watering purpose so as to provide enough water needed by plants for their nourishment. This low powered solution is not only limited to gardens but can also be used for more efficient and automated management of large gardens and smart farms with a touch of new improvements.

ACKNOWLEDGMENT

We would like to express our special thanks and heartfelt gratitude to our Capstone Project guide, Mrs. Anita Joshi, for her valuable guidance, keen interest, and encouragement and for giving us this golden opportunity to do this project which has enriched our knowledge and experience. We would also like to extend our gratitude to our colleagues for their constructive input and constant support.

REFERENCES

[1] Ariyaratne, U. & Vitharana, V. & Deelaka, L. & Herath, H.. (2022). IoT Smart Plant Monitoring, Watering and Security System.

[2] T. Ahmad, D. Zhang, "A critical review of comparative global historical energy consumption and future demand: the story told so far," Energy Reports, vol. 6, 2020, pp. 1973–1991.

[3] García L, Parra L, Jimenez JM, Lloret J, Lorenz P. IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. Sensors. 2020; 20(4):1042

[4] Gultom, Judika Herianto et al. "Smart IoT Water Sprinkle and Monitoring System for chili plant." 2017 International Conference on Electrical Engineering and Computer Science (ICECOS) (2017): 212-216.

[5] bin Ismail, Muhammad Ikhwan Hanif and N. M. Thamrin. "IoT implementation for indoor vertical farming watering system." 2017 International Conference on Electrical, Electronics and System Engineering (ICEESE) (2017): 89-94.

[6] Divani, D., Patil, P., & Punjabi, S.K. (2016). Automated plant Watering system. 2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 180-182.

[7] Pavel, Monirul & Kamruzzaman, Syed & Sabuj, Saifur & Hasan, Sadman. (2019). An IoT Based Plant Health Monitoring System Implementing Image Processing. 10.1109/CCOMS.2019.8821782.

[8] "Plant Health Monitoring System using Image Processing: A Review ", International Journal of Emerging Technologies and Innovative Research ISSN:2349-5162, Vol.6, Issue 5, page no.106-114, May – 2019

[9] Dipto, Chonmoy & Sohan, Shipon & Hossain, Md & Khan, Golam & Anannya, Rifat. (2020). Plant Health Monitoring System using IoT. International Journal of Computer Applications. 175. 37-42. 10.5120/ijca2020920890.

[10] Bhuiyan, Md. Mosharraf & tengku azmi, Tengku. (2015). Development of an Automatic Plant Monitoring System. Advanced Materials Research. 1115. 458-461. 10.4028 /AMR.1115.458.

[11] Hovhannisyan, Davit & Khalifeh, Kareem & Fei, Peng & Eltawil, Ahmed & Kurdahi, Fadi. (2019). Feasibility Study of Plant Health Monitoring. 1-4. 10.1109/ISCAS.2019.8702569.

[12] Sathiyan, Paul & Swathi, S. & Sharmini, Mariya. (2021). Automated Plant Nutrient Monitoring System for Better Plant Growth. 1-7. 10.1109/ICAECT49130.2021.9392607.

[13] Monje, Oscar & Nugent, Matthew & Spencer, Lashelle & Finn, Joshua & Kim, Moon & Qin, Jianwei & Romeyn, Matt & Orourke, Aubrie & Fritsche, Ralph. (2021). Design of a Plant Health Monitoring System for Enhancing Food Safety of Space Crop Production Systems.