Automatic Irrigation System: a Case Study on Strawberry Cultivation Based on water

[1]Meghana Naik, [2]Moulya M, [3]Nagendra Prasad

Department of Information Science and Engineering,
Alva's Institute of Engineering and Technology, Mijar

***Abstract*: As Human population is increasing it is important to use world resources efficiently.**

One way of achieving this problem is by providing the plants with optimum amount of water when it is required and in a correct time in agriculture.

Here we use cloud-based drip irrigation system, which determines the amount of irrigation system, which determines the amount of irrigation water and performs the irrigation process is presented automatically.

***Keywords*: Automatic irrigation; internet, cloud, water, strawberry, agriculture**

I. INTRODUCTION

As the huge growth in global population, the required amount of food production is increased. In order to achieve this, new approaches such as hydroponics, vertical farming, biotechnology, etc. are being applied.

On other hand, development of methods for effective utilization of water resources is required because of shortage of water supplies, which is one of the leading consequence of emerging climate change. Therefore, it becomes a necessary to include automation and data processing methods in modern agricultural irrigation systems in order to optimize the water usage.

The studies include water needs of plants, determining optimum duration of irrigation accordingly. This is achieved by using daily data meteorological data, soil moisture sensors, drainage lysimeters, evaporation pans.

Here in this case study, a cloud based automatic drip irrigation system that involves measurement of water level in class A evaporation pan is proposed. Measurements from a water level sensor attached to the pan and an environmental sensor are continuously transmitted to a microcontroller that controls the solenoid valves. Through a GSM/GPRS module, the received sensor data and system status related information are sent to a cloud server.

II. THE IRRIGATION SYSTEM

The irrigation system involves four main parts which are power and actuator, data gathering, control and internet connection, and monitoring. We explain you about the MIT App Inventor used in monitoring unit in the irrigation system.

A. MIT APP INVENTOR

Young people should also have opportunities to create with computing that have direct impact on their lives and their communities. We outline two key dimensions of computational action—computational identity and digital empowerment.

We can make computing education more inclusive, motivating, and empowering for young learners. Learners have the capacity to develop computational products that can have authentic impact in their lives Even more progressive CS education that centers around the development of learners' computational thinking has largely focused on learners understanding the nuanced elements of computation, such as variables, loops, conditionals, parallelism, operators, and data handling. This initial focus on the concepts and processes of computing.

a) "'What's conversational AI?' with MIT App

Inventor and Amazon Alexa"

This work aims to democratize conversational AI technology. Currently, MIT App Inventor's block-based coding tools empower anyone to develop their own mobile apps. Soon, users will also be able to program their own Alexa skills, conversational AI, and neural networks using blocks.

b) "On Tools that Support the Development of Computational Thinking Skills: Some Thoughts and Future Vision"

Development of Computational Thinking (CT) is an area of many initiatives in the last years, due to the importance of having CT skills. There are many environments that allow learners to develop such skills, for instance Scratch and MIT App Inventor, in a visual and intuitive way. As in professional software development, assisting tools that help and guide learners are starting to appear. In this paper, we discuss the current status of these tools, based on an analysis of what state-of-the-art CT assessment tools, such as Dr. Scratch for Scratch and Code Master for App Inventor, offer. We report their limitations and envision and discuss future enhancements.

c) "A design-based approach to implementing a computational thinking curriculum with App Inventor and the Internet of Things"

The growing ubiquity of everyday devices connected over the internet, know generally as the Internet of Things (IoT), has opened up new avenues for students to explore their worlds and think and create computationally. Combining IoT with mobile technologies (such as smartphones), enables students to move their designs and computational thinking out of traditional classroom settings and into the real world. This article outlines a design-based IoT curriculum that connects Taiwanese students with the personally-relevant issue of air pollution. The curriculum employs student-driven smartphone application design, using MIT's App Inventor, with Wi-Fi enabled IoT devices.

d) "Constrained Sets: The Effects of Multi-Layered Environments in Learning App Inventor"

MIT App Inventor is a mobile application development platform that seeks to democratize the construction of mobile apps by making app development accessible to people with little to no experience with script-based programming. It uses block-based programming to introduce and teach programming concepts to its users. Users drag and drop functional and visual components onto their planned app in the screen editor, and construct the logic behind those components by using blocks in the block editor. In this thesis, we design and implement Constrained Sets, a system that allows instructors and developers to allow access to only a subset of App Inventor functionality by hiding component and block access. This system allows for the construction of multi-layered interfaces, which we then use to conduct an experiment that explores how novice App Inventor users learn App Inventor in different interface environments.

e) "Expanding Device Functionality For the MIT App Inventor IoT Embedded Companion"

AppInventor allows anyone, even - and especially - children, to create fully functional apps in a quick and trouble-free way. AppInventor+IoT expands this to include communication between the device running the app and remote devices, most notably Arduinos. The next step with AppInventor+IoT is to allow for fully autonomous projects to run on the Arduino. This is what the App Inventor IoT Embedded Companion (hereafter Embedded Companion) makes possible. While this was in development, I created project tutorials to complement the addition of the Embedded Companion into AppInventor+IoT. I made tutorials for two projects, the Autonomous Healthy Plant and the Puzzle Box, to showcase the power of autonomously-run projects. I also made slight adjustments to the existing how-to documents for setting up the Arduino and setting up a BLE connection between the app and the Arduino. Since the Embedded Companion was in development while I was building the tutorials, I extended device functionality to meet the needs of these two projects: light sensor, humidity and temperature sensor, moisture sensor, touch sensor, buzzer, and LCD.

f) "Creating a Database Log History Page for Cloud Components in App Inventor to Support Software Development Education"

CloudDB, an experimental feature of the MIT App Inventor Platform, allows users to send data across several different machines using tag (descriptor) and value (data) pairs. However, currently with CloudDB, users, by deploying several blocks, can only view specific values corresponding to known tags. Since App Inventor, a blocks-based app development interface, emphasizes the usability of its program, in this work I implement a web interface where users can view and search the entirety of their CloudDB data history. The user, for each separate project, can see their tags, the corresponding stored data, and a history of that data. Detailed user studies were conducted to discover efficacy in teaching shared data concepts using this tool. Coupled with educational workshops designed for increasing computational thinking, this extension of CloudDB helps bolster student comprehension of advanced computing concepts and strengthen their independence in design, development, and debugging processes. This work demonstrates that a comprehensive visual tool appended to a cloud database system can increase understanding and thinking about a technology so prevalent in many of today's technologies.

g) "Off the screen, and into the world of everyday objects: Computational thinking for youth with the Internet of Things"

This paper discusses the opportunities presented by the growth of the Internet of Things (IoT) to provide youth opportunities to develop their computational thinking and digital empowerment. This paper argues that to support youth in developing these literacies, we need to develop platforms that reduce the barriers of entry while still allowing them to explore and develop their computational identities. To this end, this paper introduces an extension to App Inventor by MIT that enables students to quickly design, develop, and implement IoT applications.

h) "User-defined blocks for MIT App Inventor"

MIT App Inventor is an open source web platform that lets users build their own Android Apps quickly by using blocks-based programming. MIT App Inventor currently has diverse types of components, from general blocks to specific task oriented components like sensors. However, it does not provide every type of block that an user needs. For example, currently there are no components for intricate graph visualization in App Inventor. However, there is a limitation to keep adding new categories of blocks, not only because the complexity of maintaining the system will increase, but also because an individual user's need is necessarily not a general need for others.

III. APPLICATION

The system is deployed in a high tunnel area like a greenhouse, inside research fields of Cukurova University. Two separate high tunnel areas are used to test the performance of the system. First one is the experimental area on which the automatic drip irrigation system is installed. Second one is the control area which is irrigated using conventional methods. Because of of agricultural research requirements, each area is area is divided into four subsections and four different irrigation regimes are applied to each subsection.

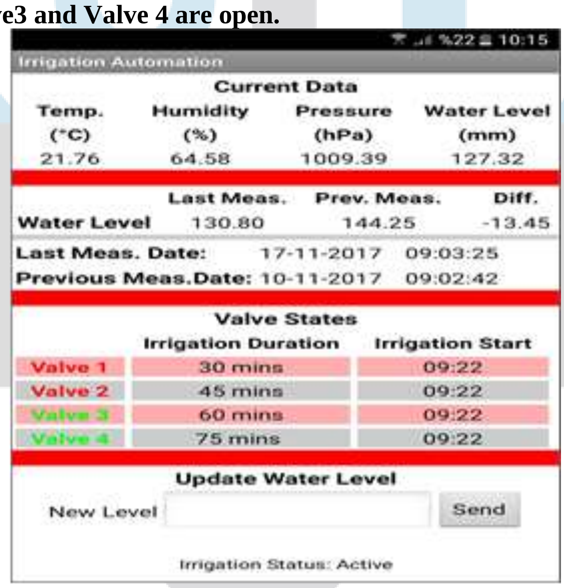
$$t = \frac{E_{pan} K_{cp} P A}{q n} \qquad (1)$$

The above equation is used to calculate the irrigation duration.

where E_{pan} is the cumulative free surface water evaporation at irrigation interval (mm), K_{cp} is plant-pan coefficient, P is plant cover (%), a parameter related with area of plant leaves, is field area (m^2), q is flow rate from the emitters, and n is the number of drippers in the field .

Fig. 2. Screenshot of the Android application taken during irrigation.

Valve 1 and Valve 2 are closed. Valve3 and Valve 4 are open.



IV. DRIP IRRIGATION USING WIRELESS SENSOR NETWORKS

One of the most important application of the WSNs in the PA is the irrigation system control. The interest comes naturally from saving water. For this aim, many researches were conducted to enhance the irrigation control system by coupling novel technologies with the agricultural practices. Among irrigation strategies, the drip irrigation system was considered as the most efficient policy to save water use. Moreover, combining this strategy with the WSNs leads us to have a great benefit from the farmlands. However, the irrigation system reliability need more attention, mainly in the case of general or partial dysfunction. For this aim, we present in this paper a model architecture for a drip irrigation system using the WSANs.

Recent practices in precision agriculture include two main micro irrigation methods which promote interesting water efficiency. The first method is the drip irrigation. It allows water to be dripped to the plants roots through pipes containing several emitters. This irrigation system is composed of the following components: water source (generally is a tank) which is connected with a main tube called main pipeline. To this line, several pipes are connected using manual or electrical valves that control the water flow. The pipes go through the field and distribute water for each plant.

Our proposed system include the following sensors and actuators:

• Soil moisture sensor: It is used to optimize irrigation and to warn of plant stress by controlling some parameters. Such as the electrical conductivity of soil or the underground volumetric water content (VWC).

• Temperature sensor: It is used to monitor the ambient temperature. It can be analog or digital and help farmer to adjust their irrigation schedule according the temperature measured to avoid risk of evaporation.

• Pressure sensor: It is used to measure a pressure of gases or liquids and change it into a quantity that can be processed electronically. It generates a signal as a function of the pressure imposed.

a) Deployment strategy

Deploying the sensor nodes to monitor a farmland is crucial issue. In fact, many parameters must be considered to choose the most beneficial deployment, as the crops characteristics, the micro meteorological parameters, the sensors and nodes specification and obviously the farmer's budget.

b) Priority-based protocol

The first one related to information gathered from temperature and the soil moisture sensors. We classify this traffic type as normal traffic since no need for an urgent intervention is required. The second traffic type is related to information gathered from pressure sensors. We classify this traffic type as priority traffic due to the need for an emergency resolution of the detected problem (shut off the main valve, require human intervention ... etc). Now, in the case when both traffics are active simultaneously, it is clear that the reliability and the timeliness of the priority traffic is more requested than those of the normal traffic. However, in the wireless context, there are many troubles that can occur due to the sharing of the same communication medium. In this paper, we have presented a model architecture for a drip irrigation system using the WSANs. Our model includes the soil moisture, temperature and pressure sensors to monitor the irrigation operations. Specially, we take into account the case where a system malfunction occurs, as when the pipes are broken or the emitters are blocked. Also, we differentiate two main traffic levels for the information transmitted by the WSAN, and based on our previous work, we achieve a high. QoS performance through an adequate priority-based routing protocol. We have performed extensive simulations. The results prove that our solution gives better performances in terms of delay, PDR for the priority traffic. As a future work, we intend to realize a real test-bed to investigate the effectiveness of our approach.

V. CONCLUSION

The Automatic irrigation system application is as per the strawberry field is demonstrated here. Mainly it automatically computes the optimum amount of water required for irrigation. This computation is based on the water level decrement at a certain amount of time in evaporation pan.

Using the optimum amount of water prevents the plants from not only being droughted but even also over watering fungal and bacterial diseases.

Here the Android application which makes it possible to remotely check using the current system status by using required sensors. This makes work easier and takes less time compared to human related work.

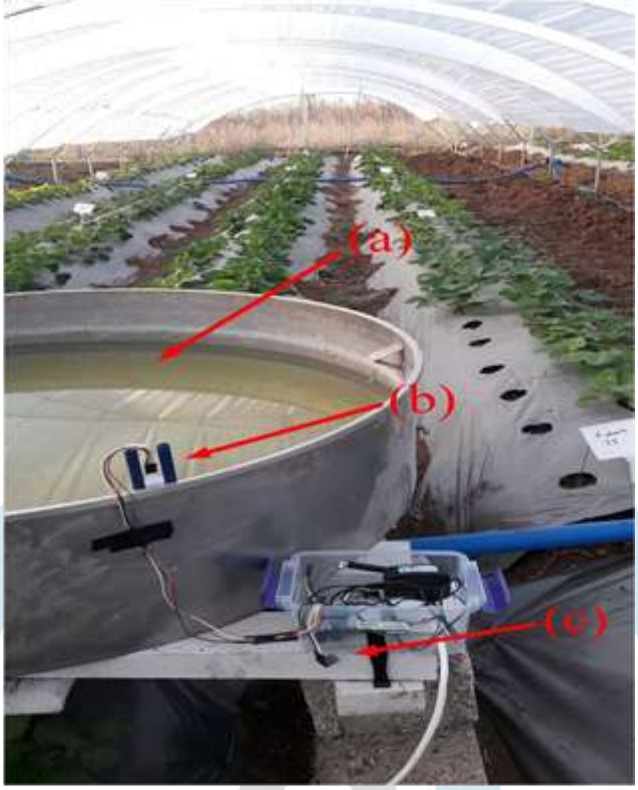


Fig. 3. Data gathering unit inside the greenhouse. (a) Evaporation pan, (b) Water level sensor, and (c) Environmental sensor.

VI. ACKNOWLEDGEMENT

We would like to thank all the lectures who had helped and encouraged us for over work from Information Science and Engineering Department of Alva's Institute of Engineering and Technology.

REFERENCES

[1] T. Ojha, S. Misra, N. Singh Raghuwanshi,"Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," Computers and Electronics in Agriculture, vol.1, no. 18, pp. 66-84, September 2015.
[2] I. Bennis, H. Fouchal, O.Zytoune, and D. Aboutajdine,"Drip irrigation system using wireless sensor networks," in Proceedings of the Federated Conference on Computer Science and Information Systems, 2015, pp. 1297-1302 .
[3] Ercan Avsar ,Kurtulus Bulus, Mehmet Ali Sandas, and Burcak Kapur, "Development of a cloud-based automatic irrigation system: A case study on strawberry cultivation.