Design and implementation of an automated irrigation system using Arduino UNO, sensors and LAMP stack

Despina Misheva, Matej Gorjanov, Ana Mladenovska and Marija Stojcheva
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University, Skopje, N. Macedonia

despina.misheva@students.finki.ukim.mk, matej.gorjanov@students.finki.ukim.mk, ana.mladenovska@students.finki.ukim.mk, marija.stojcheva@students.finki.ukim.mk

***Abstract*—This paper introduces an innovative approach to enhance plant health and development, while minimizing manual care. Automated systems that integrate sensors and regulation mechanisms have emerged as a promising solution for efficient plant care in the agriculture and horticulture sectors. However, current approaches struggle with high expenses, bulky structures and water wastage. To address these issues, we propose a compact and cost-effective plant care system that employs Arduino hardware and software to detect environmental data and adjust watering schedules accordingly. Moreover, the usage of LAMP (Linux, Apache, MySQL, PHP) stack allows for efficient data processing and management. The paper outlines the system's design and implementation, highlighting its distinctive features.**

***Index Terms*—Arduino, Web stack development, plant care, watering system, sensors, LAMP**

I. Introduction

Effective plant care is critical in industries such as agriculture and horticulture, but it can be a resource-intensive and time-consuming process. Automated plant care systems utilizing sensors and control systems have emerged as promising solutions to optimize plant growth and health while reducing the need for manual care and maintenance. However, many existing systems can be expensive, bulky, and wasteful of water resources. Current irrigation systems rely on pre-set timers to water plants, but this approach is not efficient since it fails to consider the actual water needs of the plants, resulting in excessive watering. In terms of cost, competitors offer several irrigation solutions with prices ranging from $90 to several thousand dollars [1] [2]. To address these issues, this paper presents a plant care system built on Arduino hardware and software, which is both cost-effective and compact. The system measures key environmental data, such as light intensity, humidity and soil moisture, and adjusts watering schedules accordingly. The prototype of our irrigation system was built using Arduino boards at a cost of around $60-70 and further cost reduction can be achieved by utilizing more affordable microcontrollers. The paper outlines the design and implementation of the plant care system, highlighting its advantages over existing systems. The system's compact size and low cost make it an accessible solution for plant care in a variety of settings, from home gardens to commercial agriculture operations. What sets apart the proposed system from other alternatives is the variety of sensors used to analyze multiple environmental factors, as well as the usage of LAMP to process and manage the collected data. The rest of the paper is organized as follows. In Section II we review irrigation studies, assess techniques' impact on plant growth and quality in rural/agricultural areas. In Section III, we explain continuous environmental monitoring with Arduino, from preparation to execution. In the IV section we discuss the importance and significance of the gathered data. We go over the challenges we encountered when developing the system in Section V. And in Section VI the conclusion of the research is presented.

II. Related work

Pandurang H. Tarange et al. focused on a farm irrigation system that used an embedded Linux board that provides a web interface to control the irrigation using an on or off button [3]. They described the design and construction of the system, which consists of a solar panel array, a charge controller, batteries, and a submersible pump. M. Yasin et al. discusses the need for an automated irrigation system in India's agriculture sector to meet increasing demands [4]. The proposed system uses a rain gun irrigation mechanism, controlled by a micro-controller and GSM-connected moisture sensors to automate irrigation based on soil moisture levels. K.Kansara et al. propose an intelligent irrigation system based on Internet of Things (IoT) to improve water usage in agriculture [5]. A decision-making algorithm is used by the system to control irrigaton based on the collected data from the temperature and soil sensors. Other attempts to build a working prototype of an automated watering system based on an Arduino Uno have been created. S.Akter et al. develop a smart irrigation system based on a Wi-Fi module and a mobile application [6]. K.Dixit et al. solenoid valves are used to manage the transfer of water, and Y.Shekhar et al. machine learning and Internet of Things technologies are used to address the problem [7] [8]. As aforementioned the solution we provide relies heavily on data for environmental factors that have been found most important regarding plant health. As outlined by M.Qaderi et al., humidity levels affect a plant's ability to absorb moisture and nutrients, with low levels causing dehydration and withering of leaves, and high levels increasing the risk of mold and mildew [9]. They also discussed the role of temperature in plant growth, with high temperature causing stress and hindering water and nutrient intake, while low temperatures cause yellowing and falling of leaves. According to L.Taiz et al. in Plant Physiology and Development, water uptake and circulation of nutrients in the flower are optimal when soil temperatures are cool and transpiration rates are low, making cooler hours of the day optimal for watering [10]. Thus the flowers should receive water in the morning or in the evening when the temperature is cooler, but light is still available since it represents an essential part of photosynthesis, without sufficient levels of light, the plant's growth is limited and discoloration of its leaves may happen.

III. Methodology

A. Requirements

In the initial stages of conceptualizing this project, we engaged in a brainstorming process to identify the key functionalities that needed to be implemented. The following requirements were established as a result and guided us during the development of our solution.

- The system should collect data using multiple sensors connected to the microcontroller board(s).
- The system should provide a suitable algorithm that will determine the further actions based on the collected data from the microcontroller board(s).
- The system should trigger action specified by the algorithm when a signal from the microcontroller board(s) is received.
- The system should send the processed data to a designated server using a standard protocol.
- The system should have a user-friendly interface to configure and monitor the system.

B. Electrical Components and Arduino

To build a functioning prototype, we needed a processing unit that could gather sensory data, perform necessary calculations, and act accordingly. Having considered several different alternatives, we decided to use the Arduino microcontroller, which is easily accessible and familiar to students. The Arduino board was an ideal choice for our system as it allows for easy hardware usage and programming using the Arduino Software (IDE), based on the C++ programming language. Our prototype utilizes two Arduino Uno boards that are based on the ATmega328P microcontroller. These boards have 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button.

- DHT22 sensor for Arduino: the DHT22 sensor is a digital temperature and humidity sensor. The sensor has a range of -40°C to 80°C for temperature measurement and 0% to 100% for relative humidity measurement.
- Soil moisture sensor for Arduino: a soil moisture sensor is a device that consists of two metal probes that are inserted into soil used to measure the moisture content of soil by measuring the electrical resistance between the probes which varies with the moisture level of the soil.
- Light sensor module for Arduino: a light sensor module is a device that can detect light intensity. The module typically consists of a photodiode or phototransistor and an amplifier circuit.
- Arduino Ethernet Shield W5100: an Arduino board add-on that provides Ethernet connectivity capabilities. Provides a full TCP/IP stack and supports DHCP, which means that it can automatically obtain an IP address and other network settings from a DHCP server on the network.
- Relay module for Arduino (220V to 5V): a relay module is an electronic switch that can be used to control high-power actuators. It typically consists of a relay and an optocoupler circuit that isolates the micro-controller from the high-voltage circuit. The relay module is designed to convert high-voltage AC power to a low-voltage DC signal that can be used by the Arduino to switch the relay.

C. Server

The aforementioned Ethernet shield is vital in the process of storing the data since using it allows us to treat the Arduino as a client on a given network. After establishing the connection with the network, we then use the connection to upload the data we have stored to a database. Primarily we used the Ethernet library for Arduino to initialize the usage of the ethernet port and have the Arduino act as a client that connects to a server. Furthermore we utilized the client model from the library which allowed us to directly communicate with a server and transfer data repackaged as HTTP requests. The last piece of the puzzle was setting up a server which will receive the requests and manipulate the data they contain. Using the cloud computing platform Azure we deployed a Linux virtual machine. We opted out for a platform which allows the hosting and remote access to a server in order to make the system portable. On the virtual machine, we used the image Ubuntu 16.04 with a single CPU and 1GB of RAM. We have then configured an application server using Apache2. For the database component of the server, we used the relational database management system MySql. Using its secure installation process we ensured that only authorized access is permitted. Moreover, for easier management of the database we installed the phpMyAdmin web-based tool and created the necessary tables within the database. Lastly, we added 2 server-side scripts written in PHP which are configured to communicate with the database. The first script receives the POST request and stores the data in the database and the other is a simple page that displays the last 10 measurements added using HTML.

D. The system as a whole

The main component of the system is an Arduino Uno board that waits for 10 quadruple of measurements sent from 3 sensors that collect data on the light, temperature, moisture and humidity. Average of the data is calculated and is sent to a server in the cloud using LAN cable to transfer the request to the router. Appropriate algorithm decides if the plant needs water, and if that is the case a second Arduino Uno board gets a signal to activate the water pump. Fig. 1 displays a block diagram that illustrates the system, while Table I contains the pseudo code for the algorithm. Furthermore, Fig. 2 represents the system's decision-making process through an activity diagram. Fig. 3 displays a prototype of our implemented irrigation system, showcasing its efficacy in irrigating a petunia plant in real-life conditions. Since the prototype was tested on the petunia flower the currently employed algorithm matches the measured data points to a threshold values outlined by A. M. Armitage and T. Kowalski that ensures proper and steady growth for petunia [11].

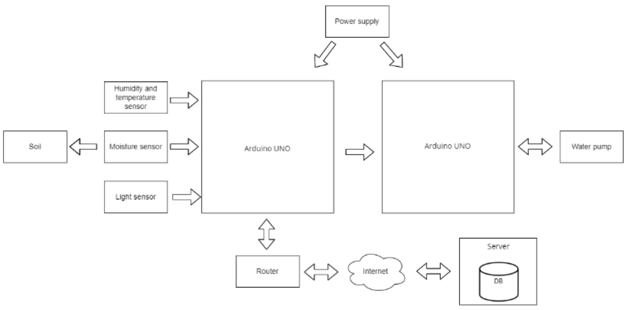

Fig. 1: Block diagram representing the system's structure and behavior

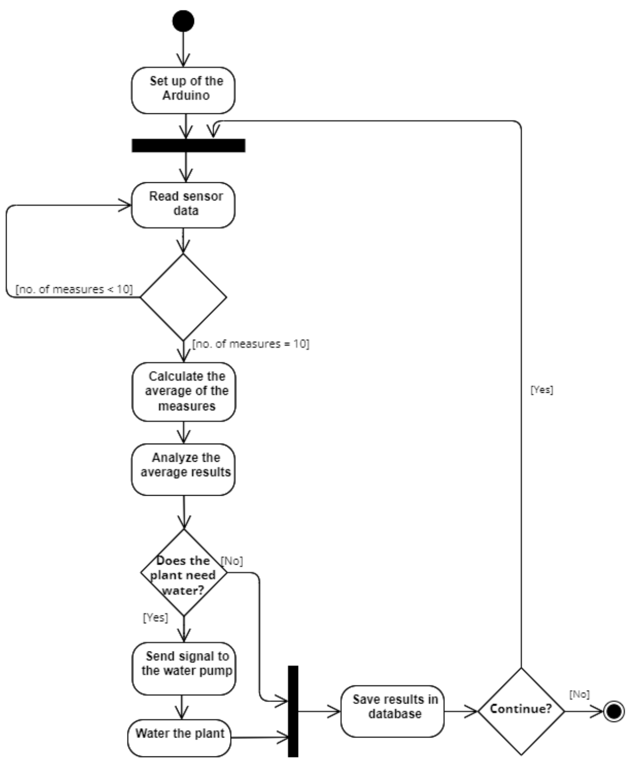

Fig. 2: Activity diagram of the irrigation process of the system

IV. Data Analysis

Our empirical study aimed to establish the threshold values required for optimal growth and maintenance of petunia flowers by collecting sensor data over several days.

TABLE I: Pseudo code for the automatic irrigation algorithm

```
Arduino 1
Begin
define sensors, server, ip and dns addresses,count←0
initialize ethernet client and arrays
for sensor measurements
water(v1,v2,v3,v4)
   needsWater ← algorithm which checks if
the plant needs water
   return needsWater
end of function
average(a[ ], n)
   calculating average value from the data
collected from the specific sensor
end of function
setup()
   start the sensors
   set input pin
   initialize pin for Ethernet shield
   If DHCP fails to assign IP address
     configure using static IP address
   connect to the server
end of function
loop()
   If count ≥ 10
     calculate average value for each of
the four sensors using average() function
     print average values of sensors
     If the client can connect to the server's port 80
       send the average values to the server
via a post request
       print "POST sent"      Else
       print "Connection failed"
     flag ← execute the water function
with the average values
     If flag == true
       write 'sign for pumping' on monitor
     If the server is not connected
       show message that the client is disconnected
       print working time of Arduino
in seconds and rate in kb/s
       count ← 0
   Else
     read information from the sensors
and store in array
     count← count + 1
end of function
End
-----------------------------------------------
Arduino 2
Begin
setup()
   set output pin
   print "Starting. . . "
end of function
loop()
   signal listener ← pin reading
   If signal listener == 'sign for pumping'
     output pin ← HIGH
     print "Finished pumping"
     output pin ← LOW
end of function
End
```

Based on our results, we recommend the following threshold values: 0-511 analog voltage levels for light, 5000-8000% for soil moisture, 21-24°C for temperature, and 40-60% for humidity. Further refinement of these values could lead to increased efficacy in petunia cultivation. Table II presents records from the database that displays the sensor readings and watering status for the petunia at a specific time. However, it is crucial to note that the threshold values we determined are specific to petunia flowers and may not be applicable to other flower species due to potential variations in their properties.

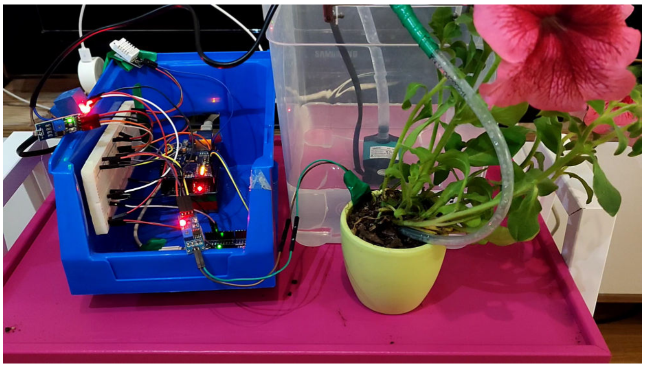

Fig. 3: Prototype of the irrigation system

TABLE II: Records from the database

Light	Soil	Temp	Humidity	Time	Flag
509.50	7180	22.00	46.30	02.05.2023 15:56:36	Yes
620.30	4230	22.00	46.00	02.05.2023 17:11:43	No
701.43	3750	21.00	45.00	02.05.2023 19:07:43	No

V. Limitations and discussion

One of the challenges we faced was connecting the project through a Local Area Network (LAN), which provided reliable data transmission but had limitations in terms of accessibility. We recognized that using Wi-Fi would have allowed for greater mobility, particularly for transmitting the project throughout a larger environment. Regarding the watering system, we faced several potential limitations such as an insufficient pump, pump burnout due to running out of water, and the amount of water discharged. We plan to install a water sensor above the pump to detect low water levels and ensure reliable operation. Additionally, to ensure optimal plant development and growth, flower placement relative to light sources must be considered. The light sensor's location is crucial, and we need to develop an algorithm to notify growers when plants require more light to prevent wilting.

Despite encountering several limitations due to time and resource constraints, we were able to make progress towards implementing a functioning prototype. Even though we successfully fulfilled the aforementioned requirements, our prototype has still place for improvement. Firstly, a mobile application for remote monitoring and maintenance of the flowers can be developed. In addition, machine learning can be used to predict water requirements, plant growth, and perform real-time monitoring. By making these modifications, we can enhance the project's value and provide optimal conditions for plant growth and development.

VI. Conclusion

To summarize, the plant care system presented in this paper offers an effective and efficient solution for optimizing plant growth and health while minimizing the need for manual care and maintenance. With its use of Arduino hardware and software, this system is both cost-effective and compact, making it an accessible option for plant care. By utilizing sensors to measure key environmental data the system adjusts the watering schedule accordingly. Overall, these modifications of existing solutions demonstrate the potential for innovation in the field of plant care.

References

[1] Click and Grow. The smart garden 3. https://eu.clickandgrow.com/products/the-smart-garden-3, Accessed May 2, 2023.

[2] Click and Grow. Wall farm indoor vertical garden. https://eu.clickandgrow.com/products/wall-farm-indoor-vertical-garden, Accessed May 2, 2023.

[3] Pandurang H. Tarange, Rajan G. Mevekari, and Prashant A. Shinde. Web based automatic irrigation system using wireless sensor network and embedded linux board. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5, 2015.

[4] M. Yasin, S.R.M.Zeebaree, and I.M.I.Zebari. Arduino based automatic irrigation system: Monitoring and sms controlling. In *2019 4th Scientific International Conference Najaf (SICN)*, pages 109–114, 2019.

[5] K.Kansara, S. Shah V. Zaveri, S.Delwadkar, and K. Jani. Sensor based automated irrigation system with iot: A technical review. *(IJCSIT) International Journal of Computer Science and Information Technologies*, 6, 2015.

[6] S.Akter, P.Mahanta, M.Haque, R.U.Ahmed R.Hasan, and M.Billah. Developing a smart irrigation system using arduino. *International Journal of Research Studies in Science, Engineering and Technology*, 6(1):31–39, 2018.

[7] K. Dixit, A. Ranjan, K. A. Faizan, and P. Kiran. Automated irrigation system based on soil moisture sensor. *International Journal for Research in Applied Science and Engineering Technology*, 10(7):475–479, 2022.

[8] Y.Shekhar, E.Dagur, S.Mishra, R.J.Tom, M.Veeramanikandan, and S.Sankaranarayanan. Intelligent iot based automated irrigation system. *International Journal of Applied Engineering Research*, 12(18):7306–7320, 2017.

[9] M. Qaderi, A. Martel, and S. Dixon. Environmental factors influence plant vascular system and water regulation. *Plants*, 8(3):65, 2019.

[10] L.Taiz, E.Zeiger, I.M.Moller, and A.Murphy. *Plant Physiology and Development*. Sinauer Associates, 6 edition, 2015.

[11] A. M. Armitage and T. Kowalski. Effect of irrigation frequency during greenhouse production on the postproduction quality of petunia hybrida vilm. *Journal of the American Society for Horticultural Science*, 108(1):118–121, 1983.