SMART PLANT WATERING SYSTEM

Table of Contents

Introduction

In the last 50 years, the agriculture sector has undergone a significant transformation. Modern machinery has increased farming equipment's size, productivity, and speed, enabling more land to be farmed more effectively. The next agricultural revolution is just getting started, and data and connection are at its core. Artificial intelligence, analytics, networked sensors, and other developing technologies may help enhance yields, even more, increase the effectiveness of water and other inputs, and foster sustainability and resilience in crop farming (Goedde et al., 2020). To make sure any new pests and illnesses are kept to a manageable level, horticultural crops in particular need more direct care and monitoring per plant than arable crops. The adoption of this intense but integrated management strategy cleared the path for technical advancements that mechanized and optimized formerly manual farm activities, providing additional precision and reliability (Horticulture, n.d.).

Crops were grown for horticulture range from annual and perennial species to fruits, vegetables, indoor houseplants, and landscape plants. Plants that are grown indoors are known as indoor plants. Many tropical plants, including palms, may flourish in enclosed spaces. Numerous studies have demonstrated that indoor plants increase indoor air quality, promote productivity, and reduce stress (Stanborough, 2020). The four factors that have the most impact on a plant's growth are water, light, temperature, and nutrients. When one of these four factors is altered, the plant may experience stress, which may alter, stunt, or increase growth. Water and humidity can promote plant development (ECOgardener, 2018).

However, giving plants too much water can potentially result in their demise. One of the most common mistakes people make with their indoor plants is overwatering, which can refer to either providing too much water at once or watering too regularly. A plant's health may be impacted by either kind of over-watering, which can result in problems including yellowing leaves, root rot, and damp soil. In rare circumstances, excessive watering might result in mold growth or insect infestations. Many individuals underwater their plants, which is similar to overwatering plants. This behavior may result from carelessness or a failure to comprehend the requirements of a specific plant. Plants that aren't given enough water develop dark leaf margins, wilt, or grow slowly (Birdsall, 2022). To encourage healthy growth, it is crucial that you research how much water your plants require and make sure to provide them with this amount. We intend to implement smart wireless sensors utilizing Arduino Uno to automatically water plants without human involvement due to those reasons and reasons detailed in depth in the project idea section. To monitor the soil moisture of the plant, we used wireless sensors in this project, in particular soil, humidity, temperature, and ultrasonic sensors using Arduino Uno. A plant's quality is greatly influenced by water. Therefore, through this project, we guarantee continuous watering for sustainable plant growth.

Project Idea

Problem Statement

By creating oxygen and organically purifying the air, plants contribute to a healthy environment. Many people adore having plants in their backyards. However, due to urbanization and lack of space, many people used to cultivate plants in a mold or pot of dirt that was set on a windowsill. To support life and growth, this plant needs standard breeding practices such as watering and adequate sunlight. Many often, due to their busy daily schedules and vacations, people neglect to water their plants, which causes them to experience a variety of diseases before dying. What's more, the world's most serious issue in present day society is the deficiency of water assets, farming is a requesting employment to devour a lot of water. It is exceptionally fundamental to use the water assets in appropriate way.

Proposed Solution

The idea is to create smart plant watering framework that works by measuring the soil moisture, temperature, and humidity and then waters plants based on the results.
This project is important solution step for sustainability and safety. It maximizes nutrition for plants, reduces labour and danger, minimizes water wastage and it helps people economically in saving their water bills.

Project Scenario

We can all agree that plants have numerous advantages for all people. The smart plant watering system being developed and used in this project is primarily intended for indoor plants. In this project we use a wireless sensor network by implementing Arduino Uno along with four sensors to monitor and water plants. We selected the soil moisture sensor, humidity sensor, temperature, and the ultrasonic sensor to create a full working automatic watering system. The specified range of the soil moisture sensor and the ultrasonic sensor is adjusted specifically for the needs of individual plants, and the system is run in accordance with that. The temperature and the humidity sensor are measured for the purpose to determine the environmental conditions of the plant. The soil moisture sensor is measured to determine whether the plant is dry or not. If the plant is dry which means the soil moisture sensor shows it is below the given threshold, then the plant needs watering. Then ultrasonic sensor will measure the water level in a bucket. If the water level in the given bucket is below the provided range (cm) then the system will notify the user in three ways. First an LED will turn on and piezo speaker will be activated to make a sound. In case the user is not in their home, an SMS message will be sent to remind the user that the water tank is empty. However, if the water tank was full, then the system will start watering the plant every 3 seconds until the moisture reaches the right amount of moisture. After the watering is finished a piezo speaker will alert us with cheering sound. This project will require electricity to work, that is why we limited to indoors. However, based on our experiment a user can use a power bank for this system to work for about almost 3-5 days.

In this project we used an Arduino IDE to program the Arduino board. For easier and better experiment of the project we provided the ranges based on the time we can see the results. These results are dynamic and will be also displayed on an LCD.

Design and Implementation

For the design process we used bottom-up approach, first considering the components individually and the system as whole. The reason we chose this is because in our case, some of the components work together and can be linked with other components to form one of the results. Therefore, by dividing components of the system into groups we were able to achieve the network of the system as whole. For the time being, this project will be only implemented indoors, the reason for this is because this project requires power, and we believe it is only sustainable to use low power for few plants which are the indoor plants. For effective and reduced labour, the only thing our system requires is filling water in a tank, so if the users used large tanks, it will be enough. We did not create any firm protective covers to some of the sensors, however, when they are used the user should purchase protective plastics for the critical equipment's such as the ultrasonic sensor which is always found near water. We would like to highlight that different plants need different amount of watering. However, in this project we will not differentiate the plants based on the amount of water they need rather we will just show how the smart plant watering system works in general for all plants. For effective implementation, it will be better to differentiate the threshold of given soil moisture based on the type of the plant. This will increase the usability of energy and water in the world.

Design

1. ***<u>System Block Diagram</u>***

   The Arduino Board is programmed using the Arduino IDE software. The function of the moisture sensor is to sense the level of moisture in the soil. The motor/water pump supplies water to the plants by using Relay. If the water pump found the bucket empty, it will send SMS message by using the GSM module. To design the block architecture of our system we used the Lucid Chart.

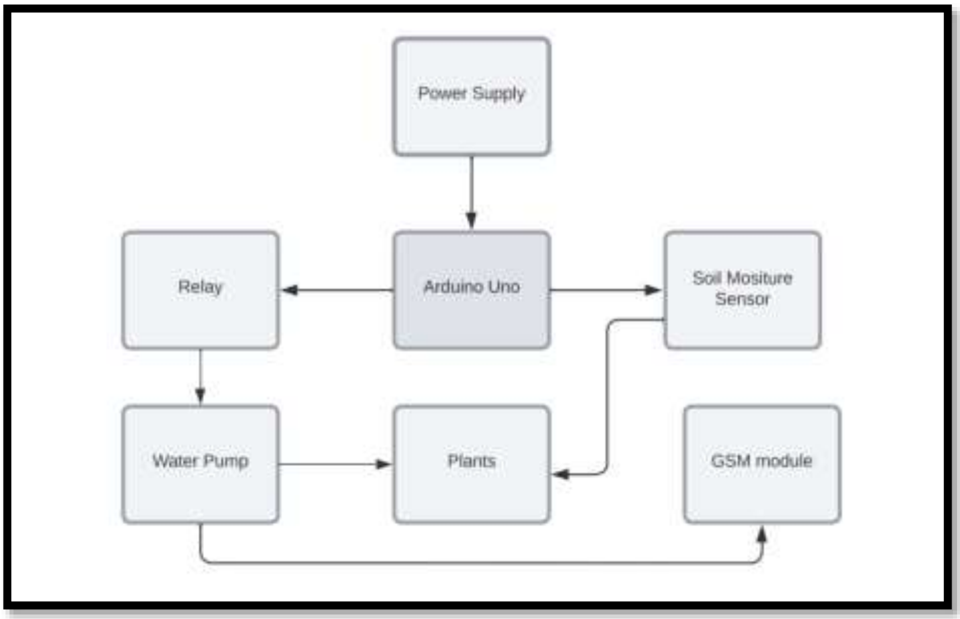
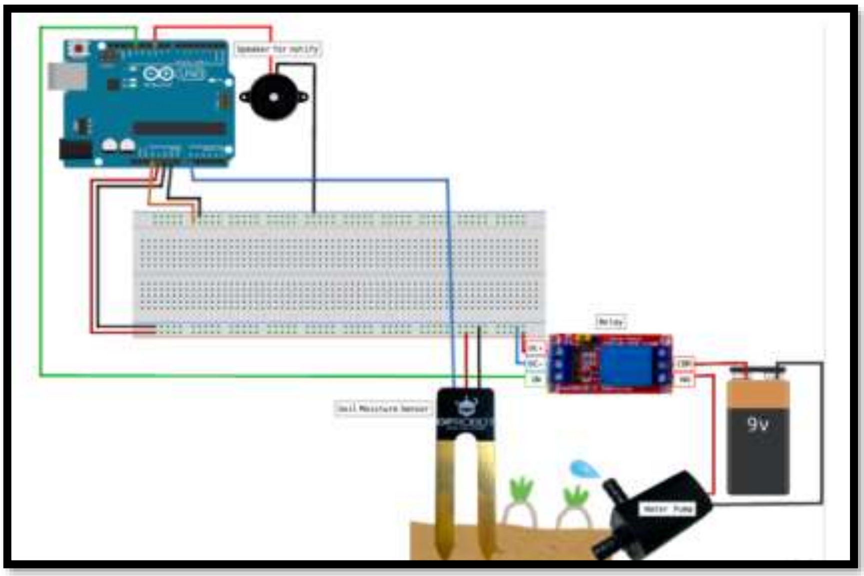
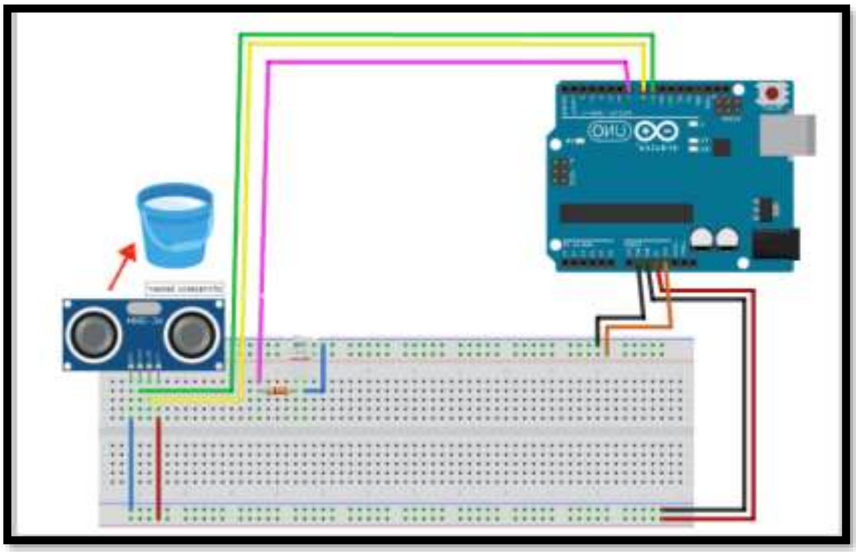


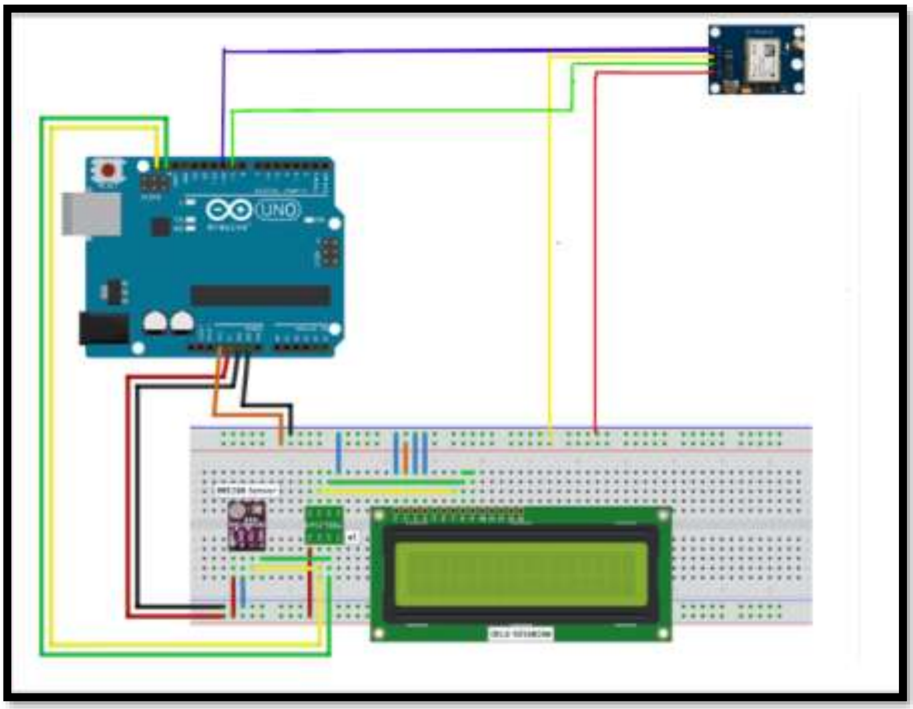
Figure1: Block Diagram of Smart Plant Watering System

2. Software Design of Smart Plant Watering System

To design the software parts of this project, we used tinker cad software which is a web app for coding and designing electronic components. We used most of the component in this software, such as the breadboard, Arduino uno, jumper wires, ultrasonic sensor, temperature and humidity sensors and many others. However, there were also materials we could not find such as water pump, the relay, we needed water bucket and plant in which we could not find in this software. As a result, we used photo editor called photo pea to include the left components. This software design is just to provide the general idea of the setup project. Because in practical we do not follow exactly as the pictures because of size differences and the connection difficulty between components.



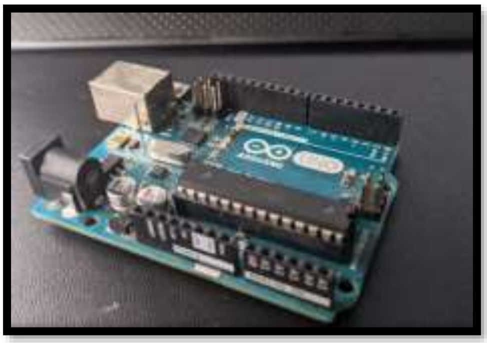
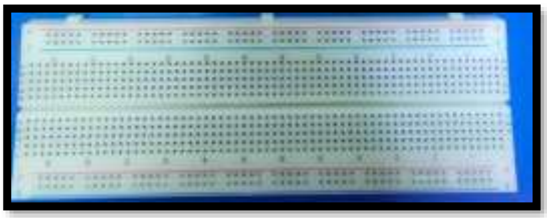




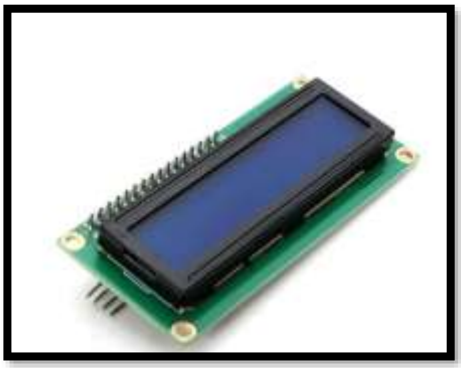
Implementation

1. ***General Components Used***

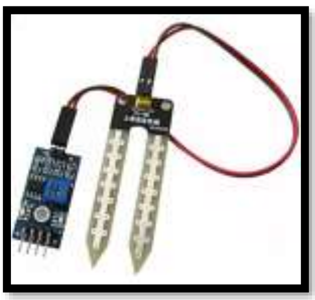
Arduino Uno is a microcontroller board based on the ATmega328. It contains 6 analog inputs, a 16 MHz ceramic resonator, 14 digital input/output pins (of which 6 can be used as PWM outputs), a USB connection, a power jack, an ICSP socket, and a reset button. The Arduino can be powered by connecting a USB to computer or by using power bank. For projects involving microcontroller boards, such as Arduino, a breadboard is a solderless construction base used for designing an electronic circuit and wiring (seedcone,2019).

LCD character Display
There are 16 pins altogether on the 16x2 LCD utilized in this experiment. Eight of the pins are data lines (pins 7–14), two are for power and ground (pins 1–16), three are used to manage the operation of the LCD (pins 4-6), and one is used to change the brightness of the LCD panel, as can be seen in the table below (pin 3). The backlight is powered by the last two pins, (15) and (16).

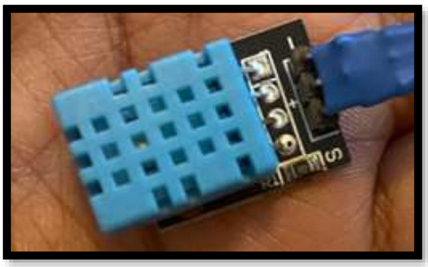


Soil Moisture Sensor
The quantity of moisture in the soil is measured by the soil moisture sensor. Immersion Gold is used in the new soil moisture sensor to prevent nickel from oxidizing. The two probes on this soil moisture Arduino sensor are used to conduct current through the ground, and the resistance is read to determine the amount of moisture present. While dry soil conducts electricity poorly, additional water makes the soil conduct electricity more readily (with less resistance) (more resistance).



Ultrasonic Sensor
The ultrasonic sensor used in this project is HC-SR04 ultrasonic distance sensor. The purpose of this sensor is to detect distance levels by sensing ultra-waves. There are four pins on this HC-SR04, Trig (Trigger), Echo (Receive), and GND (Ground). You will find this sensor very easy to set up and use for your next range-finding project.

DHT 11 Sensor
DHT 11 is commonly used for temperature and humidity sensor. A typical temperature and humidity sensor is the DHT11. The sensor includes a dedicated NTC for temperature measurement and an 8-bit microprocessor for serial data output of temperature and humidity information. Additionally, factory calibrated, the sensor makes it simple to integrate with other microcontrollers. The sensor has an accuracy of 1°C and 1% and can measure temperature from 0°C to 50°C and humidity from 20% to 90% (Components101,2021).



Relay Module
Relay is a switch that is activated by electricity. There are numerous relays that switch solenoids mechanically, but they can also work according to different operating principles. Early computers made extensive use of relays to operate telephones and carry out logical operations.

Water Pump
For a specific duty of artificial pumping, water pump is used. An electronic microcontroller is capable of controlling it. It can be switched between being on and off at will by sending a signal.

GSM module
With the help of the Arduino GSM Shield 2, an Arduino board may access the internet, place and receive phone calls, as well as send and receive SMS messages. There is antenna that comes with the GSM module which help the module to access internet. A GSM Module is essentially a GSM Modem (such as the SIM 900) attached to a PCB with various forms of output taken from the board, such as TTL Output (for Arduino, 8051, and other microcontrollers) and RS232 Output to directly interact with a PC (personal computer) (Components101,2019). For the effective use of the GSM module, in this project we added two **capacitors.** The capacitor hold energy drawn from batteries. The storage of the energy can help the GSM module or the water pump to work in case there is low power supply.

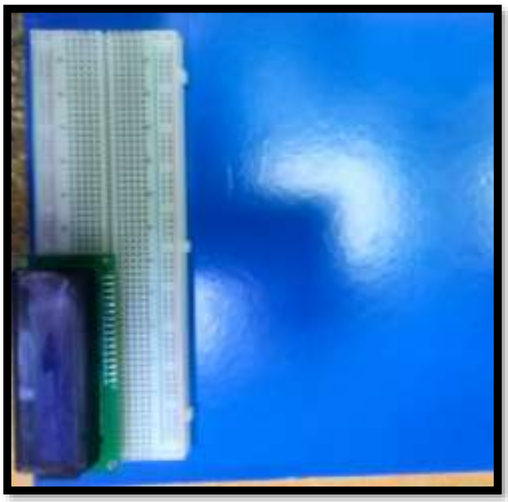
 

Configuration Steps

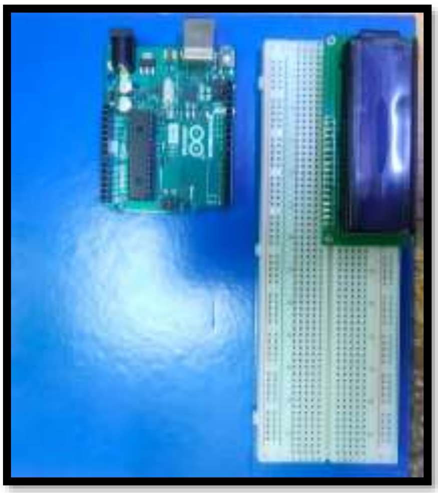
We followed the below detailed steps to successfully set up our project:

Step 1: Prepare a breadboard.

Step 2: Insert the LCD legs into “e13” up to “e28” socket of the breadboard.



Step 3: Add the Arduino Uno to set up.



Step 4: Insert one leg of the resistor of 330 ohm into “a15” and the other leg into “+” of the breadboard.

Step 5: Insert one leg of the resistor of 1 kilohm into “b26” and the other leg into “-” of the breadboard.

Step 6: Jumper wire 1(white): Connect one end of the wire to the “b13” and the other end to “-“of the breadboard.

Step 7: Jumper wire 2 (white): Connect one end of the wire to the “b24” and the other end to “-“of the breadboard.

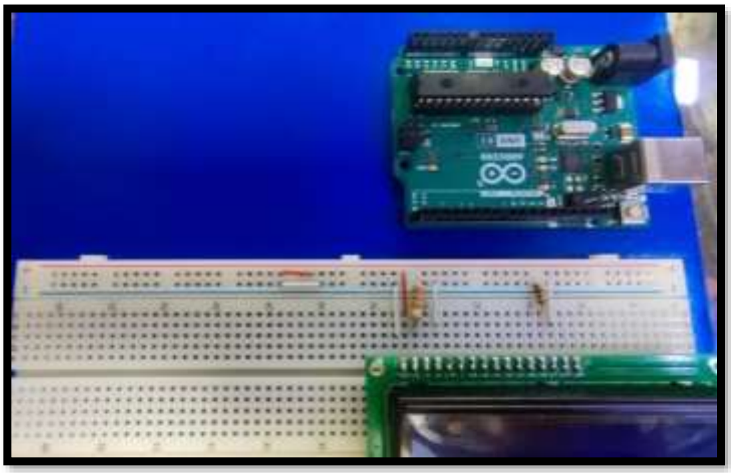
Step 8: Jumper wire 3 (red): Connect one end of the wire to the “a27” and the other end to “+“of the breadboard.

Step 9: Jumper wire 4 (white): Connect one end of the wire to the “c28” and the other end to “-“of the breadboard.

Step 10: Jumper wire 5 (white): Insert the two ends of the wire into the “-“of the breadboard.

Step 11: Jumper wire 6 (red): Insert the two ends of the wire into the “+“of the breadboard.

NB: Refer the below screenshot to see step 4 – step 11.



Step 12: Jumper wire 7 (white) : Connect one end of the wire to the “5v” pin on the Arduino. Connect the other end of the wire to the row marked “+” on the Breadboard.

Step 13: Jumper wire 8 (brown) : Connect one end of the wire to the “GND” pin on the Arduino. Connect the other end of the wire to the row marked “-” on the Breadboard.

Step 14: Jumper wire 9 (blue): Connect one end of the wire to the “pin7” on the Arduino. Connect the other end of the wire to “a15” on the Breadboard.

Step 15: Jumper wire 10 (green): Connect one end of the wire to the “pin6” on the Arduino. Connect the other end of the wire to “a16” on the Breadboard.

Step 16: Jumper wire 9 (yellow): Connect one end of the wire to the “pin5” on the Arduino. Connect the other end of the wire to “a17” on the Breadboard.

Step 17: Jumper wire 10 (orange): Connect one end of the wire to the “pin4” on the Arduino. Connect the other end of the wire to “a18” on the Breadboard.

Step 18: Jumper wire 11 (red): Connect one end of the wire to the “pin3” on the Arduino. Connect the other end of the wire to “a23” on the Breadboard.

Step 19: Jumper wire 12 (brown): Connect one end of the wire to the “pin2” pin on the Arduino. Connect the other end of the wire to “a25” on the Breadboard.

NB: Refer the below screenshot to see step 12 – step 19.

Step 20: Jumper wire 13 (brown): Connect one end of the wire to the “pin 10” on the Arduino. Connect the other end of the wire to “RXD” **on the GSM module**.

Step 21: Jumper wire 13 (white): Connect one end of the wire to the “pin 9” on the Arduino. Connect the other end of the wire to “TXD” **on the GSM module**.

Step 22: Jumper wire 13 (red): Connect one end of the wire to the “-” on the breadboard. Connect the other end of the wire to “GND” **on the GSM module**.

Step 22: Jumper wire 14 (yellow): Connect one end of the wire to the “+” on the breadboard. Connect the other end of the wire to “5VIN” **on the GSM module**.

Step 23: Jumper wire 15 (Red): Connect one end of the wire to the “+” on the breadboard. Connect the other end of the wire to “VCC” on the **soil moisture sensor**.

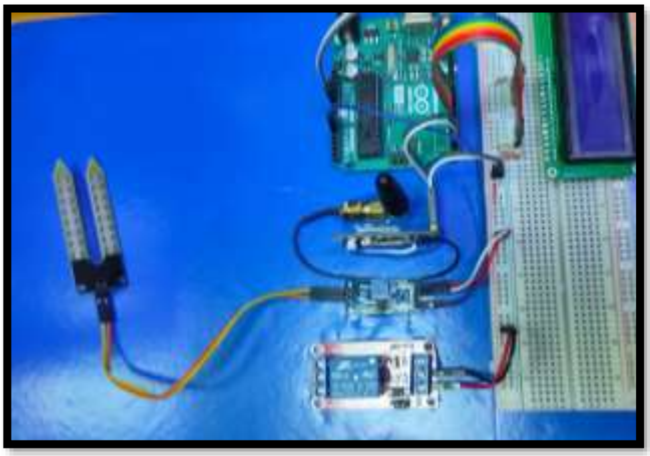
Step 24: Jumper wire 16 (white): Connect one end of the wire to the “-” on the breadboard. Connect the other end of the wire to “GND” on the **soil moisture sensor**.

Step 25: Jumper wire 15 (black): Connect one end of the wire to the “A1” on the Arduino. Connect the other end of the wire to “A0” of the **soil moisture sensor**.

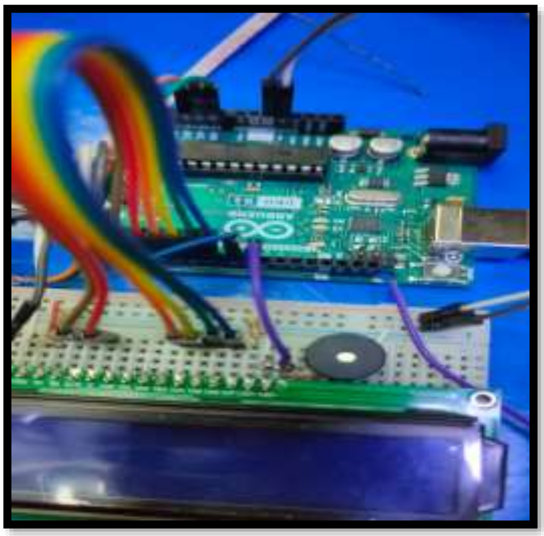
Step 26: Jumper wire 16 (red): Connect one end of the wire to the **relay module**. Connect the other end of the wire to “+ “ of the breadboard.

Step 27: Jumper wire 16 (black): Connect one end of the wire to the **relay module**. Connect the other end of the wire to “- “ of the breadboard.

NB: Refer the below screenshot for step 20 – step 27.



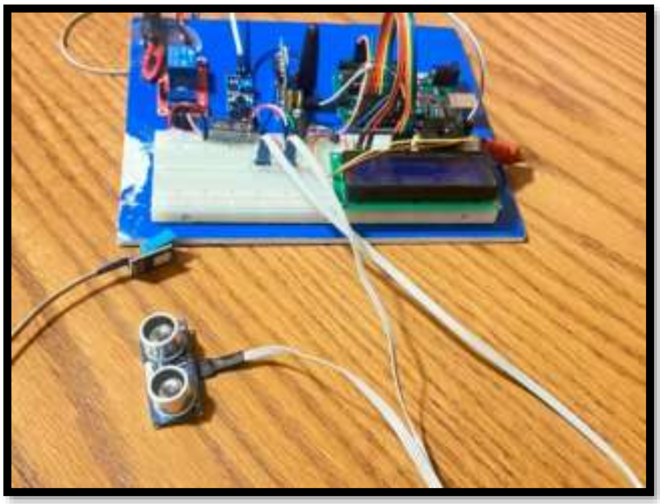
Step 28: **Piezo Speaker**: First, open the legs of the piezo speaker so that the legs can fitly inserted into the breadboard. The piezo speaker has two legs with positive and negative sign. So, enter the positive signed leg into “c3” and the negative signed leg into “c5”.



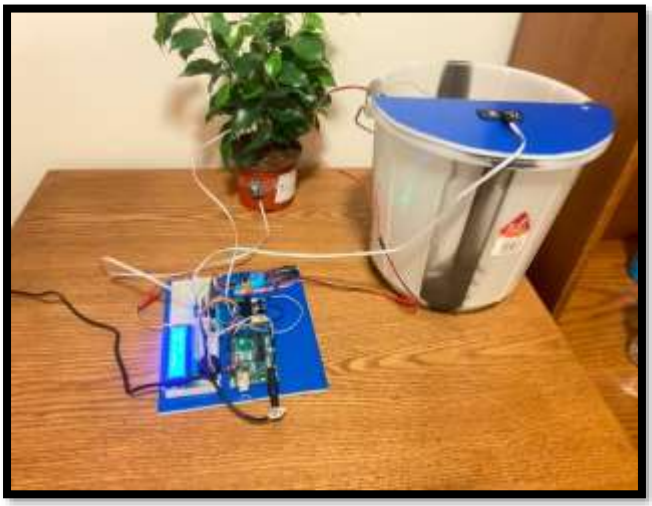
Step 29: **Ultrasonic sensor:** connect the three legs of the connecting wire to “d43-d45” on the breadboard. And connect the other three ends of the connecting wire to the ultrasonic sensor.

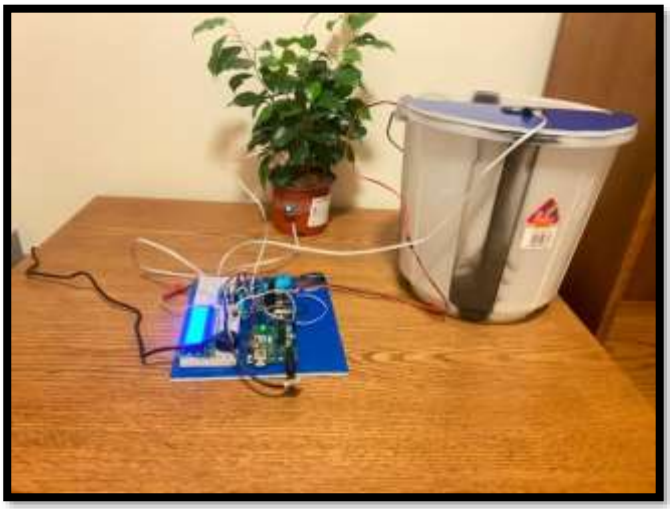
Step 30: **DHT 11 sensor:** connect the three legs of the connecting wire to “d38-d40” on the breadboard. And connect the other three ends of the connecting wire to the temperature/humidity sensor.

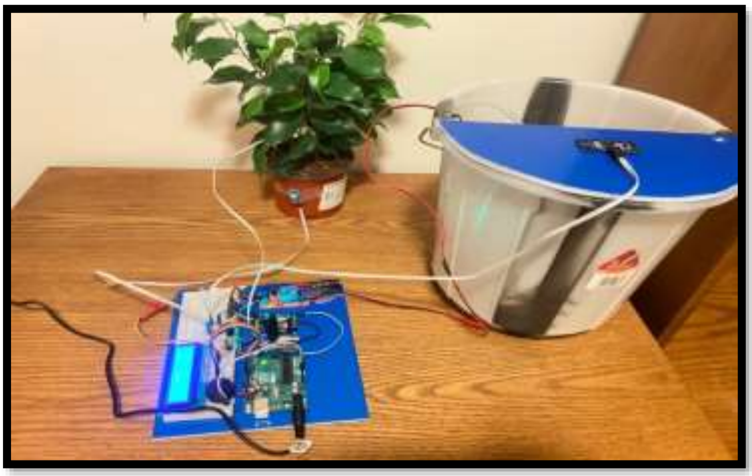
NB: Refer the below screenshot for step 29 – step 30.



Step 31: Final Hardware setup







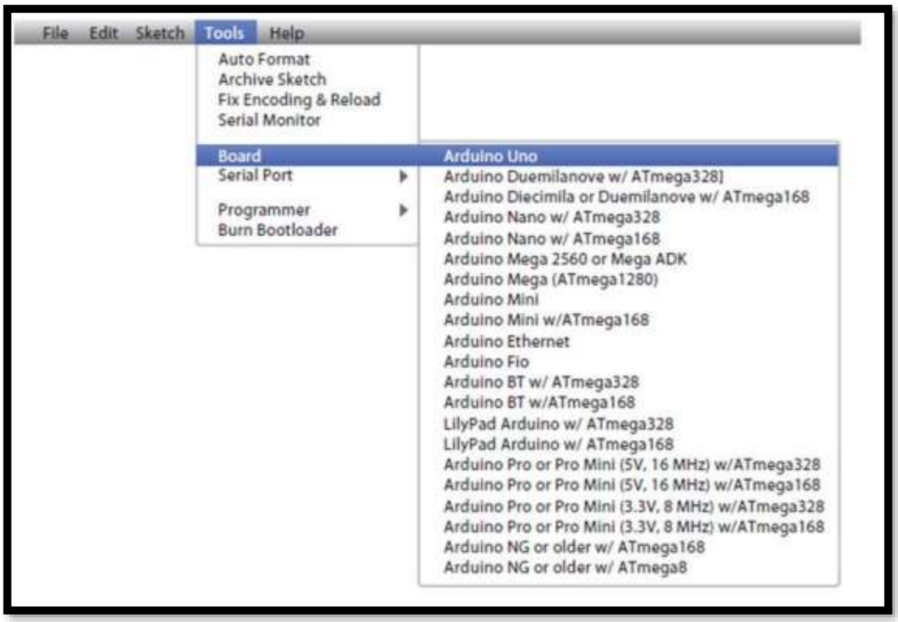
Code Module

A library is a ready-made set of software components that you can quickly add to your code. The code for this program includes four libraries. These are the liquid crystal, ultrasonic, the serial software and the DHT library.
The user should select from tools: the board type as Android Uno and the port, will be automatically shown which will have an extension with the Arduino uno name.

Once we included the Liquid Crystal library, which allows an Arduino board to control the LCD based on Hitachi HD44780. The parallel interface of LCDs requires simultaneous manipulation of many interface pins by the microcontroller to control the display. Therefore, in our case, we adjusted the pin connection based on our circuit. The following pins make up the interface: a register select (RS) pin that manages where data is written to in the LCD's memory. You can choose between using an instruction register, which is where the LCD's controller looks for instructions on what to do next, or a data register, which stores the information that appears on the screen. The Read/Write (R/W) pin that chooses between reading and writing mode. The writing to the registers-enabling Enable pin. The other pins are 8 data pins (D0 -D7). When you write bits to a register or read values from a register, the states of these pins (high or low) represent the respective operations.
We have also included the DHT library which is for the humidity and temperature sensing in which it is connected to pin A0. The software serial is for assigned pins to sim card which is places in the GSM module in this case we connected it to pin 9 and 10. The ultrasonic library is for the ultrasonic sensor to assign pins to the sensor so that it detects the water level. Then we assigned and defines variables for the ultrasonic and the music respectively.

The setup function initializes and sets the initial values. We have initialized here the serial for the SIM and we have set initialized the buzzer, relay and the LED (if the water finished) as outputs. In the loop() function, we have set four functions, the TempHum(), CheckWaterlvl() and CheckMosisture(). There are conditional statements in this part which explains the function of the system. For example, if the level of the moisture is less than 20 then the lcd will display “watering” and then the relay will pump water every three seconds until the moisture reaches 80. However, if there is no water in the tank, then it will turn off the relay and will send SMS message to the user. In addition to that, the LED will turn on and the piezo speaker will be activated. After it finished watering, it will make a sound to show that the result is successful. The TempHum() function reads the data from the sensor (DHT11) and then prints the data in the LCD. The CheckWaterLevel(), check the result of the ultrasonic sensor which measures the distance of the water level and the prints the results on the serial monitor.

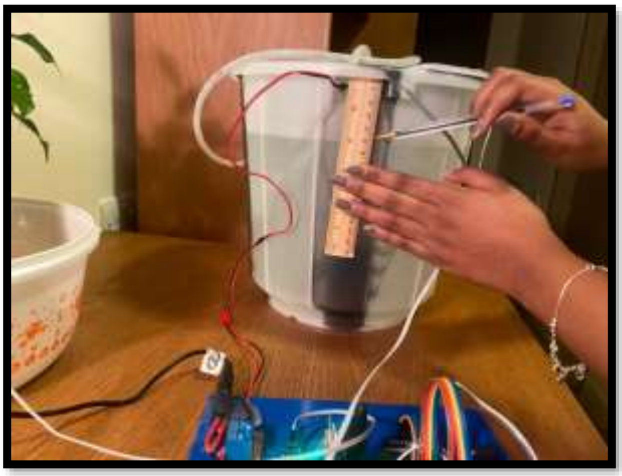
Then to run the codes, we selected tools from the Arduino, and the board: from the board we selected Arduino Uno and the port showed in the screen with an extension form of Arduino Uno.



Results

This experiment measures the soil moisture, humidity, and temperature of the plant to water it if some conditions are not meet. The conditions will be shown clearly using screenshots. We used soil sensor, DHT 11 sensor, and ultrasonic sensor to carry out this experiment.

Condition 1: If soil moisture is < 80, and the distance between the water in the bucket and ultrasonic is < 18 cm, then the plant will get watered every 3 sec until the moisture reaches its maximum level (80). Result is displayed in the LCD as well. The details are shown in the screenshots below.

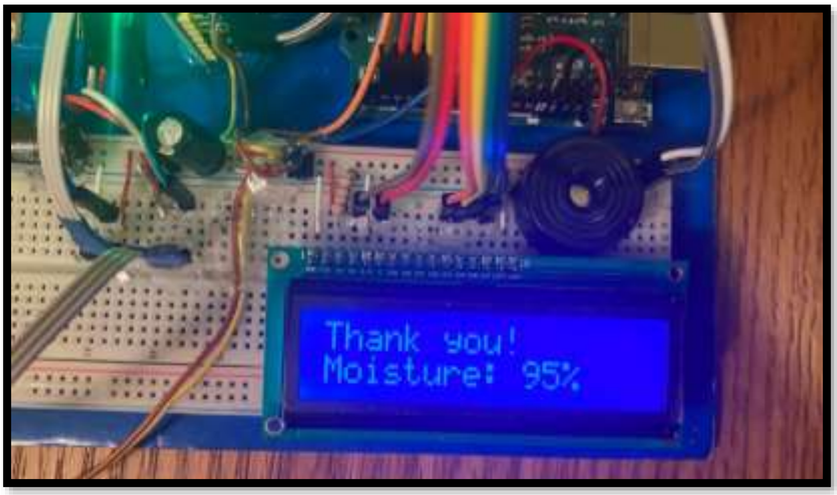


Distance between the water in the bucket and the ultrasonic sensor is 5 cm. Which means it is less than 18cm.

Soil sensor measuring the moisture of the plant

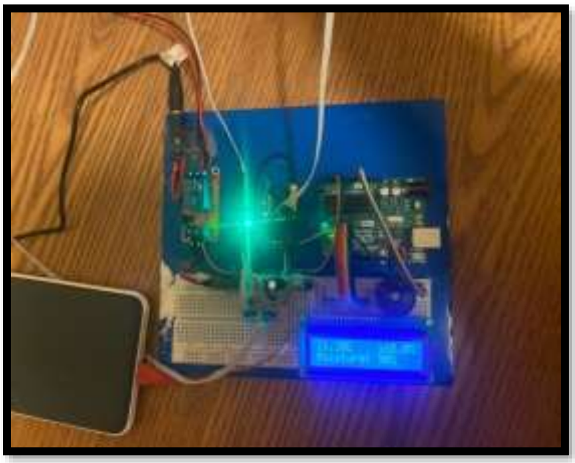
Hence, it will water the plant in every 3 seconds.

The moisture level is displayed in the LCD, and it is 7% which means less than 80%.

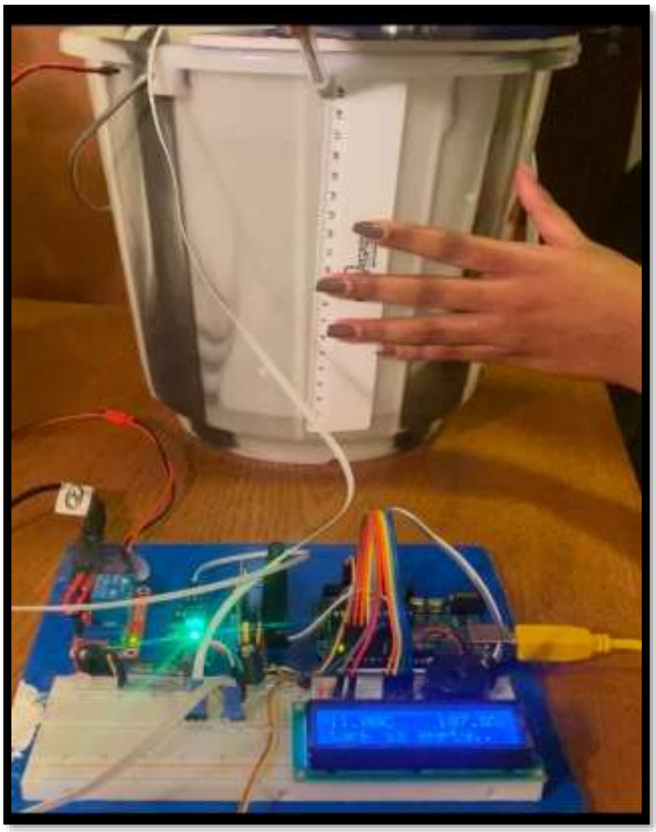


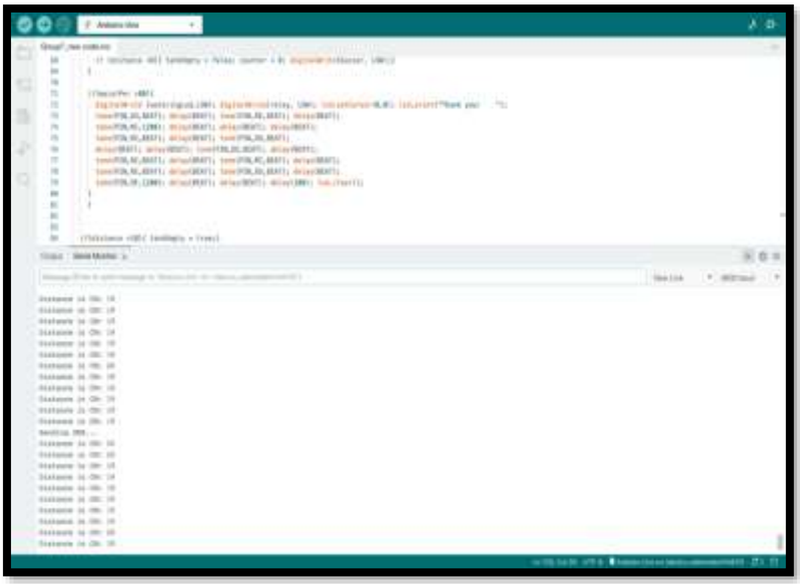
Once the moisture level reaches greater than 80%, a "Thank You!" message is displayed in the LCD, and the piezo speaker will make a sound to notify us that the plant is watered sufficiently. Then it will stop watering.

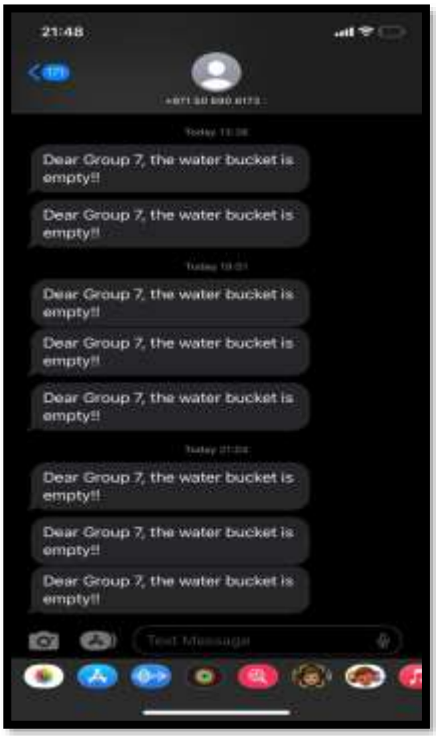
Condition 2: If the moisture level is > 80, then the plant will not get watered. For a quick demo, we have inserted the soil moisture in a small cup assuming that the moisture level is greater than 80% because our plant will take long time to get moisturized above 80%.



Condition 3: The amount of water in the bucket is measured according to its distance with the ultrasonic sensor. If the distance is > 18cm, then the user will get SMS notification every 3 sec that the bucket is empty and needs to be filled. Additionally, a light will turn on the Arduino for the notification. The distance output will also be displayed in the serial monitor.






21:48
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Dear Group 7, the water bucket is empty!!
Text Message

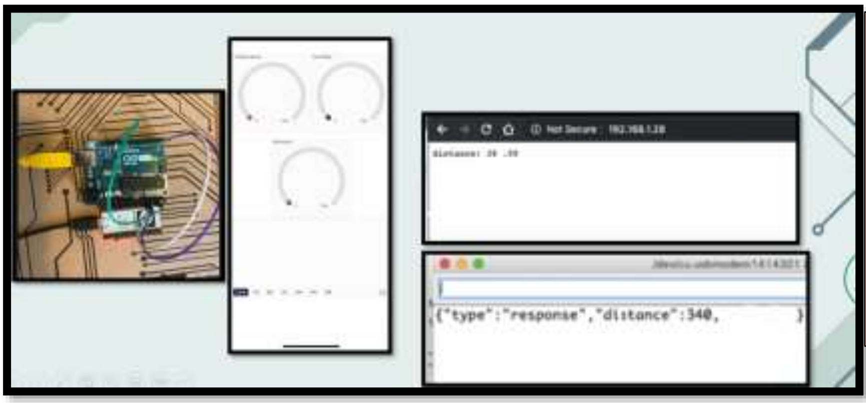
Discussion

Challenges

One of the main challenges we have encountered while working on this project is coming up with the network of the system. Almost all the components we used in this project were new to us, and as result we had problems on connecting the wires to the components. For example, one of the main challenges for us was to connect wires to the relay, which we learned it after watching you tube movies. We also had problems with components as general because we could not find them easily, we had to buy almost everything. This took time for us, and we believe we were little late to start the project because of these reasons. We also decided to follow same steps as other project except in our project we added additional sensors. However, it was hard to follow the projects because there was no detailed explanation, therefore, we came up with our own designs and code to work on the project.

Main Challenge: Connecting Arduino to ESP8266

One of our main objectives for this project was to create an app that shows the users the status of the plant based on the three sensors which are the soil moisture sensor, temperature, and humidity sensors. However, our project was built on the Arduino Board which do not include wireless module. Therefore, to add wireless module, we brought the ESP8266. Most project that consists of the wireless module are built of the ESP, however, in our case we want to integrate both the ESP8266 and Arduino to transmit data between each other. We want to transmit the data from the Arduino to the ESP8266 and then display it to the app we created using ESP8266. We first tried if it works on another Arduino by using the UART communication protocol. The term "UART" (universal asynchronous receiver/transmitter) refers to a protocol, or set of guidelines, allowing serial data exchange between two devices (Rohde & Schwarz, n.c). However, when we tried it on the Arduino, we created our project it could not work. We believe the reason is because we have the GSM module, and it works with serial software. We need to comment the serials, but we could not do that because we need the SMS notification.



This figure shows the setup of the Arduino Uno with ESP8266 and the app we have created it using Blynk software. The others are the results we found to test if there is connection between the Arduino and ESP8266.

Recommendation

Connecting many devices to create such projects on Arduino uno is very entertaining, satisfactory and lesson for us. But we also believe that this course should have the lab as a separate course. The reason is because we believe that we have to learn deeply about sensors. We are not familiar with C languages, so it was very challenging for us to code for a project from scratch. Therefore, we thought if we are working with Arduino IDE, then we should also learn deeply how to use it without the need to follow steps or guidelines. In addition to that, we recommend if there are more complicated projects on the lab, that could help us sometimes to face challenges with new materials, specifically on the connection part. To make the course also more entertaining we believe that there should some gamification related to the sensors since most of us students love to learn in visual environment.

Lesson Learned

The key lesson we took away from this project was real Arduino Uno programming and design implementation. We now understand that every stage is crucial to the development of the project's design and execution. We also discovered how to combine various sensors to create a workable method for automatically plant watering. We gained a better understanding of how the Arduino uno operates and how to include various sensors into a single piece of code that can be executed from the Arduino uno IDE. The sensors can be extremely sensitive, and we have found that great care must be taken to preserve their longevity.

Conclusion

To sum up, we can see how the wireless sensor network could be simply incorporated into our daily lives. By considering the soil moisture, humidity, and temperature of a plant, we have suggested an autonomous plant watering system. We were able to integrate and condition the setting according to our design by using a built-in library of how each of those sensors operates on the Arduino IDE. Additionally, the project successfully tracked the changes in soil moisture as a result of the many experiments we ran on the system and presented the data.

Future works

In the future, we aim to expand our scope to water different types of indoor plants by monitoring their soil moisture. Because we believe, different plants need different amount of watering. As we mentioned in our project idea section, we did not differentiate the plants based on the amount of water they need rather we only showed how the smart plant watering system works in general for all plants. Moreover, by developing it further, this smart plant watering system could be utilized in a variety of horticultural fields, saving labor and time. We also intend to use solar energy instead of the non-renewable power supply because power consumption has been one of the major limitations of employing wireless sensor networks.

Additionally, for user convenience and ease of use, we intend to create an application that will allow users to track the progress of their plants by accessing all the data to their phones. Currently, users only receive SMS notifications when there is not enough water in the bucket.

References:

Automatic Watering System for my plants. Arduino Project Hub. (n.d.). Retrieved December 2, 2022, from https://create.arduino.cc/projecthub/lc_lab/automatic-watering-system-for-my-plants-b73442

Birdsall, L. (2022, November 11). 8 common mistakes you're making with your houseplants. Reviewed. Retrieved November 27, 2022, from https://www.reviewed.com/home-outdoors/features/how-to-care-for-houseplants

Dht11–temperature and humidity sensor. Components101. (n.d.). Retrieved December 2, 2022, from https://components101.com/sensors/dht11-temperature-sensor

ECOgardener. (2018, March 4). 4 factors that affect plant growth and how to deal with them. ECOgardener. Retrieved November 27, 2022, from https://ecogardener.com/blogs/news/4-factors-that-affect-plant-growth

Goedde, L., Katz, J., Ménard, A., & Revellat, J. (2020, November 11). Agriculture's connected future: How technology can yield new growth. McKinsey & Company. Retrieved November 27, 2022, from https://www.mckinsey.com/industries/agriculture/our-insights/agricultures-connected-future-how-technology-can-yield-new-growth

Horticulture: Top 5 technologies for sustainable crops. FutureLearn. (n.d.). Retrieved November 27, 2022, from https://www.futurelearn.com/info/courses/explore-how-farmers-produce-food-sustainably/0/steps/60773

Michigan State University. (n.d.). What is horticulture? A modern applied plant science! Department of Horticulture. Retrieved November 27, 2022, from https://www.canr.msu.edu/hrt/about-us/horticulture_is

Rohde & Schwarz GmbH & Co KG. (n.d.). Understanding uart. Rohde & Schwarz. Retrieved December 2, 2022, from https://www.rohde-schwarz.com/us/products/test-and-measurement/oscilloscopes/educational-content/understanding-uart_254524.html

Seedcone. Arduino Official Store. (n.d.). Retrieved December 2, 2022, from https://store.arduino.cc/collections/seeed

Stanborough, R. J. (2020, September 18). 7 science-backed benefits of indoor plants. Healthline. Retrieved November 27, 2022, from https://www.healthline.com/health/healthy-home-guide/benefits-of-indoor-plants