Spring 2024

Self-Watering Plant Device

Nick Sabetta
njs114@uakron.edu

Self-Watering Plant Device

Project Design Report

Design Team 14

Allen Jagic

Madhav Khanal

Nicolas Sabetta

Faculty Advisor: Dr. Ida

December 3rd, 2023

Table of Contents:

11. Appendices

List of figures:

List of tables:

1. Abstract (AJ,MK,NS)

In response to the increased demand for houseplants, an automated plant care system was designed to ensure the health and vitality of indoor plants. This system employs an array of sensors and an embedded microcontroller to monitor and manage essential plant care aspects, including soil moisture levels, sunlight exposure, and water distribution. The device features a compact design, making it suitable for indoor and outdoor settings, and is powered by a 4.5-volt DC battery pack. Key components of the system include soil moisture sensors that continually

assess soil moisture levels, a water level sensor to detect reservoir levels, light level and temperature sensors for monitoring plant health, and a water pump for automated watering. To facilitate user interaction and data transmission, the system incorporates an HC-05 Bluetooth module and a mobile application written in Java. Users can connect to the system via Bluetooth, view sensor data through the app, and receive notifications about low water levels and insufficient sunlight exposure. This automated system presents a way for individuals of all experience levels to ensure that their houseplants receive the right amount of care and attention for optimal growth and well-being.

1. Problem Statement (AJ, MK, NS)

- **Need**

Due to the advent of the COVID 19 pandemic, people spend more time in their homes than ever. In fact, many jobs have transitioned from the office to working from home. As a result of being home for longer times, people purchase houseplants to bring them comfort. Not only do they bring comfort, but according to a study done by the University of Hawaii, they also increase humidity and decrease dust particle accumulation in a house [1]. This can lead to individuals living a healthier life. However, according to statista.com, about 30 percent of all purchased houseplants die. There are several reasons that contribute to this, including improper watering and lack of sunlight. Therefore, a device is necessary that can address all three of these concerns.

- **Objective**

A device able to automatically water a plant and monitor sunlight exposure time is needed. The device should be able to gather information on exposure to sunlight, it should be able to detect moisture levels, and it should be able to independently water the plants. This device should also be able to communicate with a user such that any problems it is unable to solve, such as inadequate sunlight exposure. It should have some method to alert the user to the problem.

Background

The basic theory behind the active automated plant watering system involves a system comprised of sensors and a water dispersion system which will distribute water to a house plant when the moisture levels of the soil are too low. In a sense, the automated plant watering system is a smaller scale crop irrigation system; the key difference is the amount of water necessary to sustain a single house plant as opposed to a field of crops. According to André Glória et al., automatic irrigation systems based on manual controllers are designed to distribute water to fields or gardens at specific times and for specific durations each day, and these systems are used in 80% of irrigation projects today [2]. In research conducted by Volodymyr Pastushenko and Anastasia Stetsenko during their efforts to create an automated control system for a water well, they report that plants consume water for both plant tissue formation (approximately 0.01-0.03%), with the rest used for transpiration, which is the vaporization of liquid water [3]. Using a control system to dispense fixed amounts of water at specified times is a common practice in irrigation, and a variation of this concept would be employed in the automated plant watering system. But while the irrigation systems used in crop fields are concerned with only the dispersal of water, the house plant system would also take the amount of sunlight being received by the plant into consideration. While crop irrigation systems are in fields, where the crops are constantly exposed to sunlight, indoor house plants are not always guaranteed that luxury. Light sensors are used to determine whether the plant is receiving an adequate amount of sunlight, which is just as important to the plant as receiving an adequate amount of water. While watering plants is essential to their survival, a well-watered plant in an area with too little sunlight will not be nearly as likely to survive as a plant in a well-lit area. A system which combines a water distribution system with a light detection system will allow plant owners to ensure the health of their plants. Another key issue with irrigation systems is the issue of water pressure or flow level. As explained by Christine Joy T. Dinio et al., valves are needed to control the flow in irrigation, and pumps are required as a water source [10]. Although the proposed system is on a much smaller scale than a standard crop irrigation system, a pump to provide water to the soil is an important component in the proposed system.

Currently, other automated plant watering systems for house plants are making use of various drip or pump systems to automatically water plants throughout the day. Active solutions make use of a device, such as a microcontroller, with moisture sensors to detect when the soil within the plant does not have an adequate amount of water and a pump to distribute water to the plant, whereas passive solutions utilize various drip systems in which water slowly drips into the soil moisture over time. The passive drip solutions tend to be more cost effective than the active solutions, but the active solutions allow for more programmability and do not apply water until a specified moisture threshold is no longer reached. Leonard Goldberg, James P. Romano, and John J. Feketa submitted for a patent for such a system in 2006; their system consists of a cavity in which a pot can be placed, a reservoir to hold water, a pump to remove water from the reservoir, and a host to deliver water to the plant [4]. Goldberg, Romano and Feketa's system is also designed to be programmable to "automatically water a house plant at predetermined intervals, as well as deliver a predetermined amount of water" [4]. Such a system has more components involved in its operation, but the programmability is an important factor in its effectiveness. An example of a passive plant watering system, on the other hand, would be a watering globe. These are globes designed to be filled by the user and inserted into the soil of the plant, and over a period of one to two weeks water will trickle into the soil. These systems keep the moisture level of the soil constant, but they require the user to both clean the globe and pay attention to the water level in the globe [5].

The easiest and most cost-effective option for a self-watering house plant system would be the watering globe. The most common type of watering globes has a reservoir on top of a thin, long tube that is inserted into the soil of the houseplant. This allows for the water to slowly drip into the soil over a period of about one week. However, the amount of time the water will last inside the reservoir will depend on the moisture of the soil. The water might only last a few days in some types of soils while lasting almost two weeks in other types of soils [5]. A second limitation of the watering globe is once the water has run out there is no way for the device to notify the user to refill the water reservoir as there is no programmable feature on the watering globe. Therefore, the user must periodically check the globe to ensure that the water has not run out of water. A third limitation is when the user inserts the globe into the soil there is a high possibility that the soil will go into the tube and clog the watering globe preventing the water from reaching the plant. If the reservoir is full when the watering globe gets clogger the user might not immediately realize there is a problem. This might cause the plant to begin to wilt and eventually lead to the death of the plant. If the goal of the watering globe is to decrease the amount of routine maintenance that the user needs to complete for the house plant to survive the watering globe has many chances of falling short of this goal. A second device that is currently on the market for making it simpler for users to monitor and keep their plants in good condition is the Smart Plant Water Meter and Health Sensor sold by Sage Sill [6]. This device will monitor various conditions of the plant and send the user updated via their app Flower Care. However, a limitation of this device is that the device can only measure the conditions of the plant and soil. This device will not be able to do anything to improve the health of the plant. If the device detects that the plant's conditions are unfavorable to the health of the plant, it will only be able to notify the user instead of being able to alter the conditions of the plant.

One current system that utilizes similar concepts to the proposed system is the Automated plant watering system. This system, unlike the system currently being designed, is intended for industrial agricultural use. It uses the ATmega328 microcontroller and various sensors to detect soil moisture levels, and when the soil moisture is below a certain threshold, it automatically releases a predetermined amount of water. If the water tank does not have enough water, it notifies users to add more water. It also uses a mobile application to notify users of these events [7]. This is different from the proposed system. The proposed system will be designed for individual plants, as opposed to industrial systems. It will use the pic24fj128ga010 microcontroller. One important functionality of the proposed system that is not present in the Automated plant watering system is the photosensor. Using the information from the photosensor, the proposed system will be able to notify the users as to whether their houseplants are receiving enough sunlight, and if the plants are not receiving enough sunlight, it will be able to alert the user to move the plant to an area with more exposure to sunlight. One other difference between the two systems is that the Automated plant watering system is programmed to water plants twice a day, whereas the proposed system will be able to water the plants as required by the feedback from the soil moisture sensor as many times as necessary.

One patent for a similar system is the Automated Plant Watering system [8]. This system has a weight sensor along with a control module that detects the weight of a plant and disperses an appropriate amount of water for the plant. The proposed system will be similar in that it will use a sensor to release a prespecified amount of water. Another patent that exists for a similar system is the Robotic Watering unit [9]. This patent utilizes a robot for watering the plants instead of a hose/nozzle. Like the proposed system, it uses a microcontroller along with various sensors to determine the conditions necessary for watering plants and uses a robot to water them. While these systems have different methods and microcontrollers for the implementation, they have the same underlying principle as the proposed system. All these systems will utilize various sensors (photosensor, soil moisture sensor, etc..) to communicate with a controller of some kind, such as the ATmega328, as used in a system designed by Drashti Divani et al. [7], or the pic24fj128ga010 microcontroller. There will be a feedback system that can automatically recognize certain conditions necessary for the system's activation, such as low soil moisture levels, and will water the plants accordingly.

- **Marketing Requirements**
  - The system should be compact.
  - It should automatically detect low moisture levels within the soil.
  - It should automatically distribute enough water to the plant when the soil moisture level is low.
  - It should notify users when the water level within the reservoir is low.
  - It should notify users of inadequate sunlight exposure.

2. Engineering Analysis

- **Circuits (NS)**

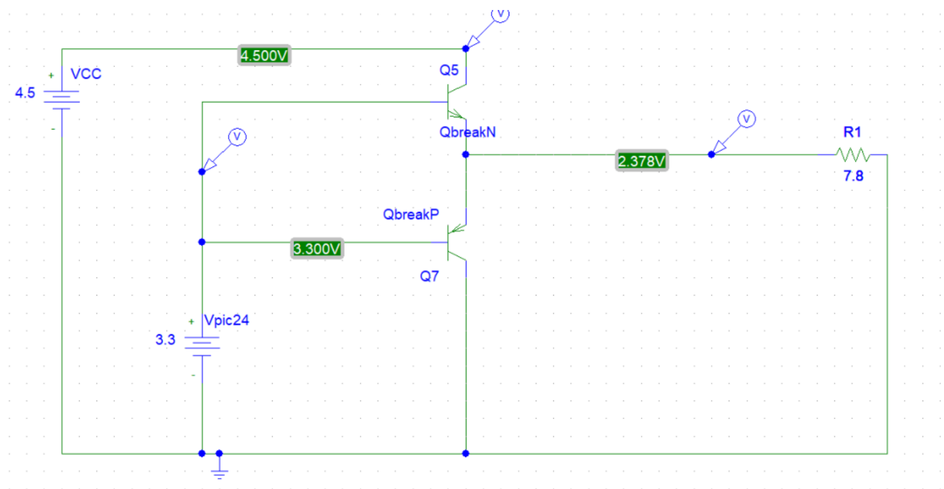


Figure 1:Switching circuit for the water pump.

According to the water pump datasheet, it requires 3 volts to run the pump. After testing the water pump it will turn on between 1V and 4.5V. The PIC 24 has an out signal that can be set between 3 and 5 volts. For this circuit it will be assumed the PIC 23 will supply 3.3V. Figure 1 above shows the switching circuit that will supply the water pump with power. The resistance of the water pump was measured to be 7.8Ω, shown at R1 in figure 1. VCC will be the 3 AA batteries. One PNP and one NPN transistor will be needed for this circuit. When the PIC 24 supplies 3.3 Volts to the circuit the NPN transistor will conduct, allowing the 4.5 volts to be supplied to the water pump. When there are 0 volts supplied from the PIC 24 the PNP transistor will conduct which will connect to water pump to ground. A DC-DC converted will likely need to be connected in between the 4.5V source and the switching circuit to ensure the batteries always supply 4.5V to the system. Figure 2, below, shows the effect of changing Vpic24 on the circuit. Vpic24 is shown in red, voltage of the water pump is shown in blue and the VCC is shown in green. The water pump will activate at around 1 Volt and will be running at around 2.5 Volts when Vpic24 is outputting 3.3 volts.

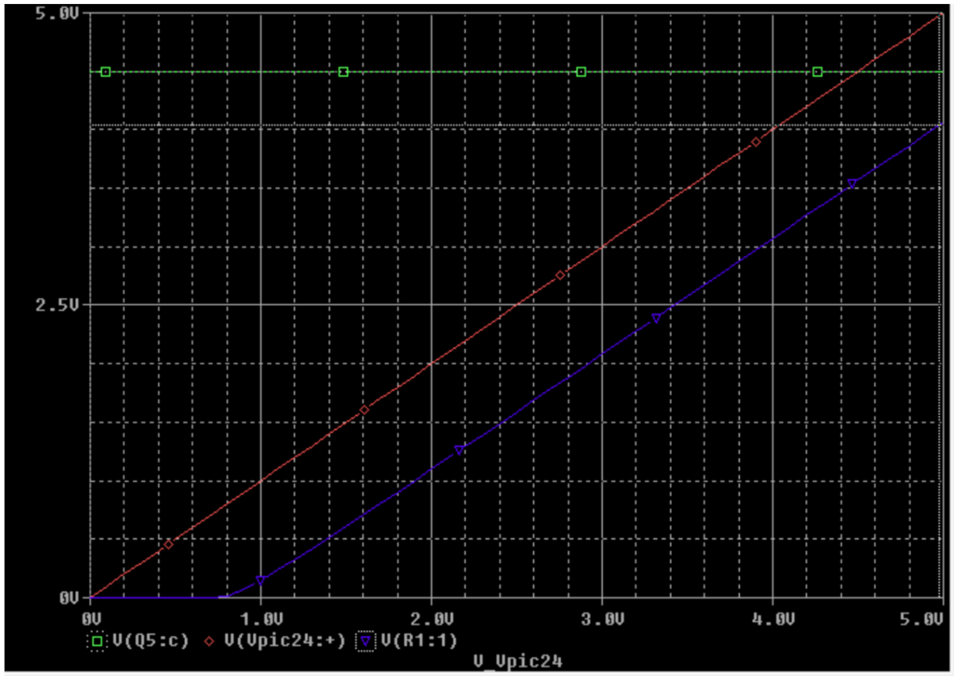


Figure 2: Voltage levels when changing Vpic24

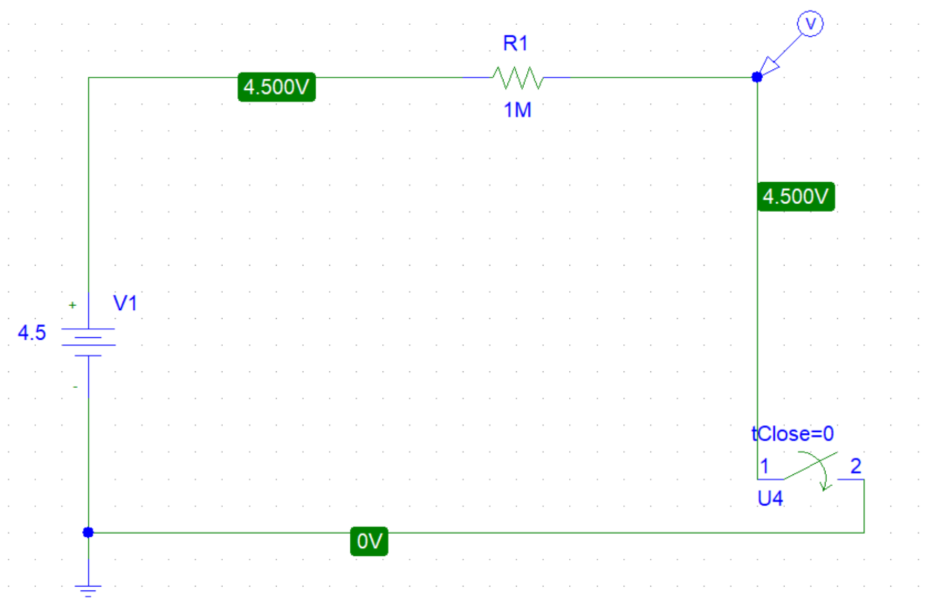


Figure 3: Water Level Sensor Circuit

To ensure the water pump will not run when the water level is too low there needs to be a sensor that will shut off the pump. The circuit in figure 3 shows how the water level sensor will work. The voltage probe is where the output signal is located. When the sensor detects water, the switch will be closed making the voltage at point V 0 Volts. While testing the voltage was closer the 0.5 volts. When there is no water detected the voltage will increase at point V. While testing the voltage was 2.5 volts before entering the water. When pulling the sensor out of the water the voltage would rise to about 1.3 Volts. This shows that it will have to be taken into consideration that the sensor will not show immediately that the water level has fallen below the sensor. The output will also collect to the PIC 24 to allow the user to be alerted when the waler level is low.

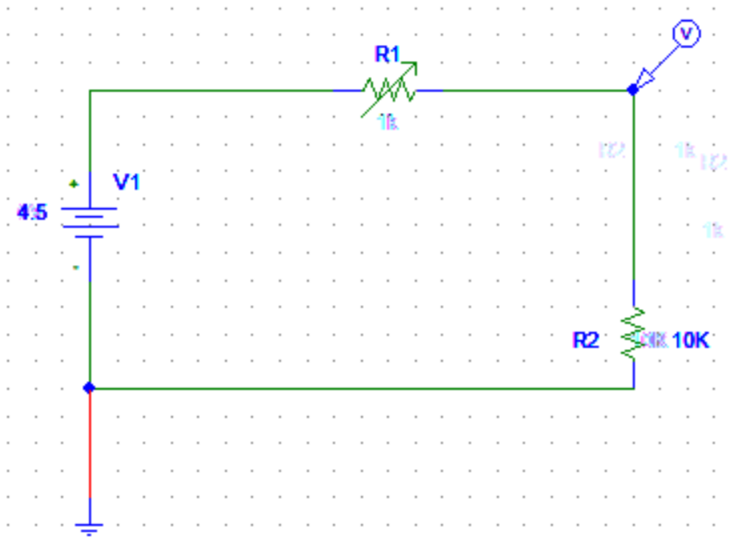


Figure 4: Circuit for the Photosensor

The Photosensor will allow the user to know how often the plant is receiving light. The circuit for the photosensor is shown in figure 4. The voltage probe is where the PIC 24 will be reading the output. When the sensor is receiving light the resistance of the photosensor will decrease, meaning that the voltage across the 10K resistor will increase. When the photosensor is not detecting light the resistance will increase, meaning the voltage across the 10K resistor will decrease. This circuit will not affect the water pump so it will not be tied into the other circuits except for receiving power.

The output for the soil moisture and the temperature sensor will go directly to the PIC 24. These two sensors will not need any additional circuits before connecting to the embedded systems.

- **Electronics (AJ)**
    - Water Level Sensor
    - DC Battery pack
    - Soil Moisture Sensor(s)
    - Light Level Sensor
    - Temperature Sensor
    - Water Pump
    - Explorer 16/32
    - HC-05 Bluetooth Module
- **Signal Processing (AJ)**
    - The processor will use an analog-to-digital convertor and have a reformatting happen within itself to make it readable as well as Bluetooth transmittable and then will be transmitted via UART to the Bluetooth signal

- The embedded processor has a built-in analog to digital convertor within itself that will be used to convert the signals from the sensors to digital to make it the standard used for this device

- **Communications (AJ)**
  - Analog/Digital Input/Output from all sensors, connected to SPI bus on Microcontroller
  - UART Communication to Bluetooth from Microcontroller to connect to database/application
- **Computer Networks (AJ)**
  - Bluetooth 2.0 will be used in this project to communicate from the microcontroller to the database that will store the sensors data once it has been translated by the microcontroller.
  - SPI will be used as the standard of communication via the SPI bus on the microcontroller, where all the sensors will be connected to via jumper cables.
- **Embedded Systems (AJ)**
  - Explorer 16/32 Microcontroller Board by Microchip fitted with PIC24fj128ga010 processor

3. Engineering Requirements Specifications (MK)

1. The system's physical dimensions should not exceed 1 sq ft to ensure it fits easily in indoor or outdoor settings. Need for portability, and it should not take up too much space, as stated in Marketing requirement 1.

2. Implement soil moisture sensors to monitor soil moisture levels every 30 minutes. Water evaporates and plants use it, therefore frequent checks are required. This is done to meet the needs of marketing requirement 2.
3. Develop a water dispensing mechanism capable of delivering water when the soil moisture level falls below 20%. This is to meet the requirements of marketing requirement 3 and distribute water when necessary.
4. Incorporate a water level sensor to detect when the water reservoir reaches 20% of total holding capacity. This much water was necessary to keep the water pump submerged.
5. Utilize light sensors to monitor sunlight exposure every minute. This is to ensure the plant has adequate sunlight levels, meeting marketing requirement 5.
6. Implement a system to alert users via Bluetooth when water reservoir level is low, or sunlight exposure is inadequate. This is to make the system more automatic and keep it from running into problems like lack of water in the reservoir.
7. Create a mobile app/web app interface to enable users to monitor plant status remotely. This is an essential part to keep the plants from dying due to lack of care.
8. Implement a safety feature that stops watering if the moisture level does not increase after 20 seconds to prevent overwatering. This is done as a safety mechanism to make sure that the water does not overflow.

4. Engineering Standards Specifications

- **Safety (AJ, MK, NS)**
  - Lab safety
  - OSHA
- **Communications (AJ)**
  - SPI will be used for the sensors
  - UART will be used for the data transmission with Bluetooth
- **Data Formats (AJ)**
  - Standardized format per sensor/Bluetooth communication
  - Fahrenheit temperatures
- **Design Methods (MK)**
  - Top-Down approach
- **Programming Languages (AJ)**
  - C/C++
  - Java
- **Connection Standards (MK)**
  - Bluetooth 2.0
  - SPI

5. Accepted Technical Design

- **Hardware Design (AJ, NS)**

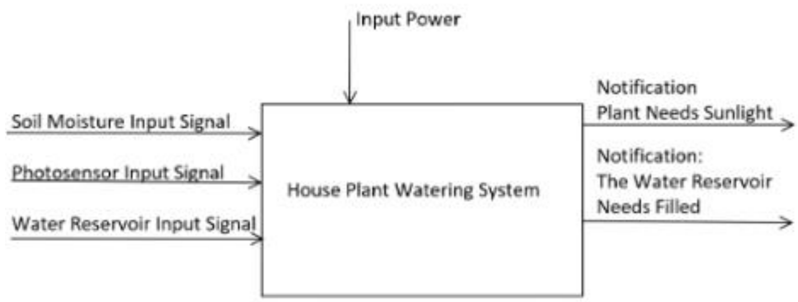


Figure 5: Level 0 Block Diagram

The hardware design began with setting up the level 0 block diagram. Shown in figure 5, the inputs for the self-watering house plant system mainly come from the sensors that will be reading information about the health of the plant. This device will be seasoning the moisture of the soil, the light levels that the plant is receiving and the amount of water that remains in the water reservoir. This device and the sensors will need power as an input. The outputs from this device will be notifications that the plant needs sunlight and when the water reservoir needs to be filled. These notifications will be sent to the user's phone via the app that will be created. All the inputs, outputs, and functionality for the level 0 block diagram are shown in Table 1.

Table 1: Functional Requirement Table for Level 0 Diagram

Module	Self-Watering Plant System
Inputs	• Input Power • Soil Moisture Input Signal • Photosensor Input Signal • Water Reservoir Input Signal
Outputs	Notifications to user

	• Plants need sunlight. • Water reservoir needs to be filled. • Water pumped into the plant from the water reservoir.
Functionality	The House Plant Watering system will help the user to remember to take care of their plant by tracking how much water and sunlight the plant is getting and will send notifications to the user when the plant is not getting the nutrients that it needs.

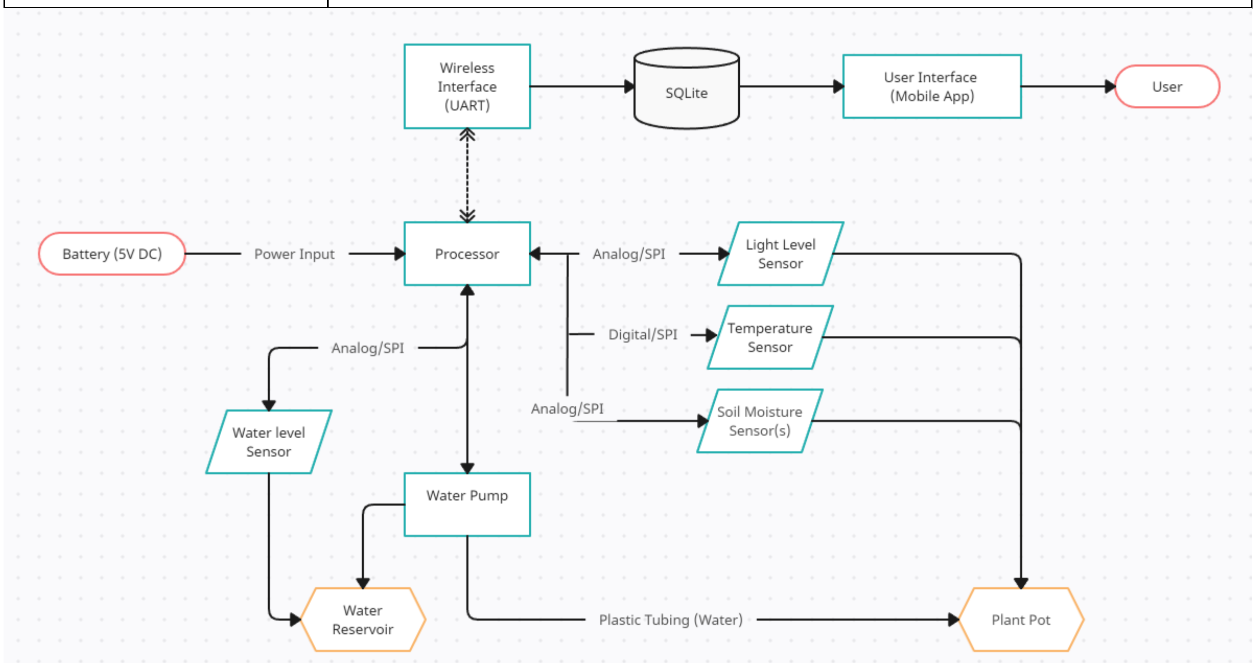


Figure 6: Level 1 Block Diagram

The next step in the design was to set up the Level 1 block diagram. In this block diagram the inner workings of the device were expanded. The main input is power. The device will be powered by a 4.5-volt DC battery pack. This will supply power to the processer and all the sensors. Most of the sensors require between 3.3-5 volts. Each sensor will be described in more detail below. Three of the sensors will be responsible for reporting the monitoring of the health of the plant. The light level and temperature sensor will be mounted on the top of the device and will not have any impact on the water pump. The information from these two sensors will be sent directly from the processor to the app for the user to view. The processor will view the readings from the soil moisture and will activate the water pump when a soil moisture of below 20% is received. More testing is needed to determine the flow rate and the amount of time the pump will be activated when low soil moisture is detected. The last sensor, the water level sensor will be responsible for alerting the user when the water level is low. This will also prevent the pump from functioning when the water level is low. The water level sensor will be in the water reservoir just above the water pump.

Table 2: FR Table for Processor

Module	Processor
Inputs	• 5V DC Power (Battery) • Light Level Sensor (SPI Signal) • Temperature Sensor (SPI Signal) • Soil Moisture Sensor(s) (SPI Signal) • Water Level Sensor (SPI Signal) • Wireless Interface (UART)
Outputs	• 3.3V Stepped down power to sensors and pump • Wireless Interface (UART)
Functionality	The processor is set to take the battery input and use that (stepped down) to power the water pump as well as the sensors that are hooked to it using both

	analog/digital signals. These sensors will send data back to the processor via SPI about their specific duties, in which the processor will then communicate with the water pump as well as the wireless interface in order to get data into the database. All sensors aside from the Temperature sensor will be analog sensors, and the water level sensor will be converted from analog-to-digital itself for proper use.

Table 3: FR Table for Water Pump

Module	Water Pump
Inputs	• 3.3V DC Power • Processor
Outputs	• Water
Functionality	This water pump is designed to take 3.3V DC power from the processor and use it in tandem with the water level sensor as well as the soil moisture sensor to properly release water into the soil of the plant.

Table 4: FR Table for Soil Moisture Sensor(s)

Module	Soil Moisture Sensor(s)
Inputs	• 3.3V DC Power from Processor
Outputs	• The sensor gives an analog output when the soil moisture is low.
Functionality	The soil moisture sensor(s) are going to be set throughout the pot of the plant to measure the moisture of the soil at various depths/areas. These will then be sent to

	the processor to determine whether the plant has been significantly watered by the pump or not.

Table 5: FR Table for Temperature Sensor

Module	Temperature Sensor
Inputs	• 3.3V DC Power from Processor
Outputs	• Temperature
Functionality	The temperature sensor is going to be actively taking the ambient air temperature and reporting it back to the processor for it to determine if it is too warm/cold in the current positioning.

Table 6: FR Table for Light Level Sensor

Module	Light Level Sensor
Inputs	• 3.3V DC Power from Processor
Outputs	• Light Level
Functionality	The light level sensor is going to measure the amount of light going to the plant and tell the user if the light level is significant enough in its current positioning

Table 7: FR Table for Water Level Sensor

Module	Water Level Sensor
Inputs	• 3.3V DC Power from Processor

Outputs	• Sensor will output a voltage when there is no water present.
Functionality	The water level sensor is going to be placed inside of the water reservoir where the water pump is and is going to inform the user if the water level is getting too low. “Too low” is classified based on water pump positioning as the pump must remain submerged in water or else it will short.

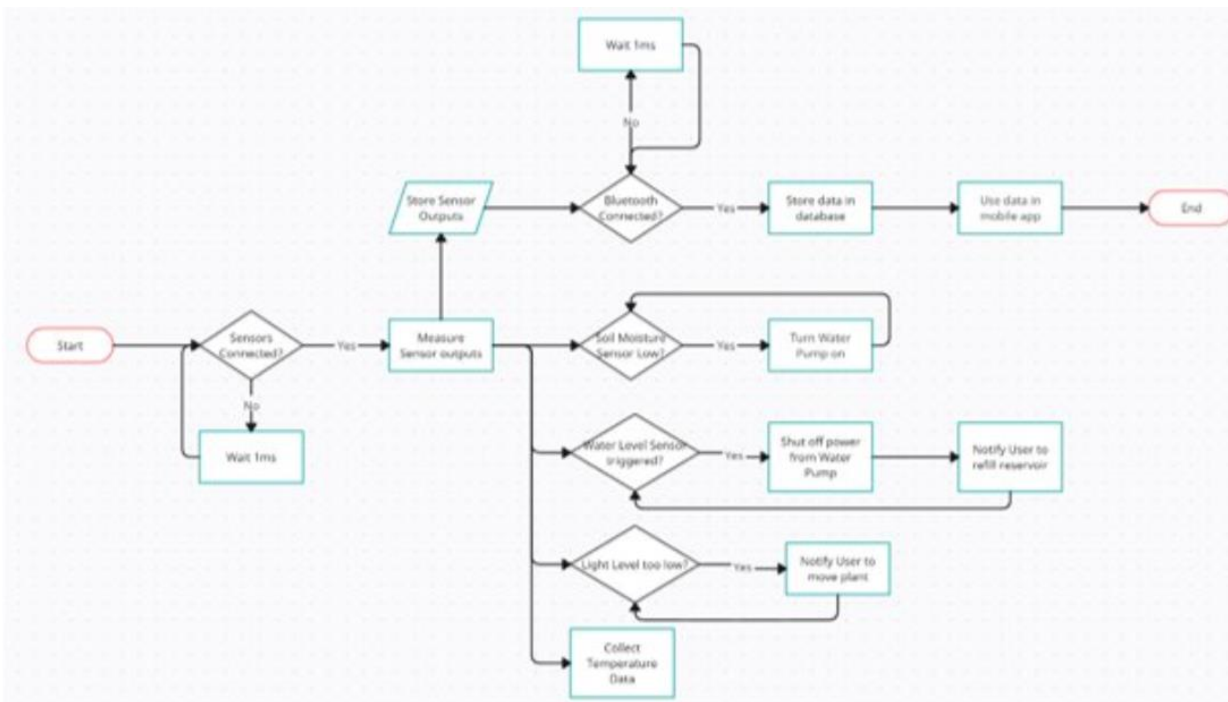


Figure 7: Level 1 Block Diagram for Processor

The figure above shows the functionality of the processor/embedded system in more depth. As can be seen, the ‘N’ number of sensors are connected to the processors SPI bus. The ‘N’ is used because there are a set number of soil moisture sensors needed, and further testing needs to be done to determine that number. The sensors will communicate to the processor via analog and digital depending on the specific settings they are set to (intervals of data communication). The processor will also be working in tandem with the water pump to be able to warn the user if the water level is getting too low and if the user needs to refill the water reservoir as to not short the pump. The other warning that the device will be sending to the user is if the light level sensor is sensing too low light on the plant. While all this data is being communicated back and forth between the sensors, water pump, and processor, it will be then getting translated through code and then transferred up to a database using a Bluetooth connection with a UART communication standard that will be on the board of the processor.

- **Software Design (MK)**

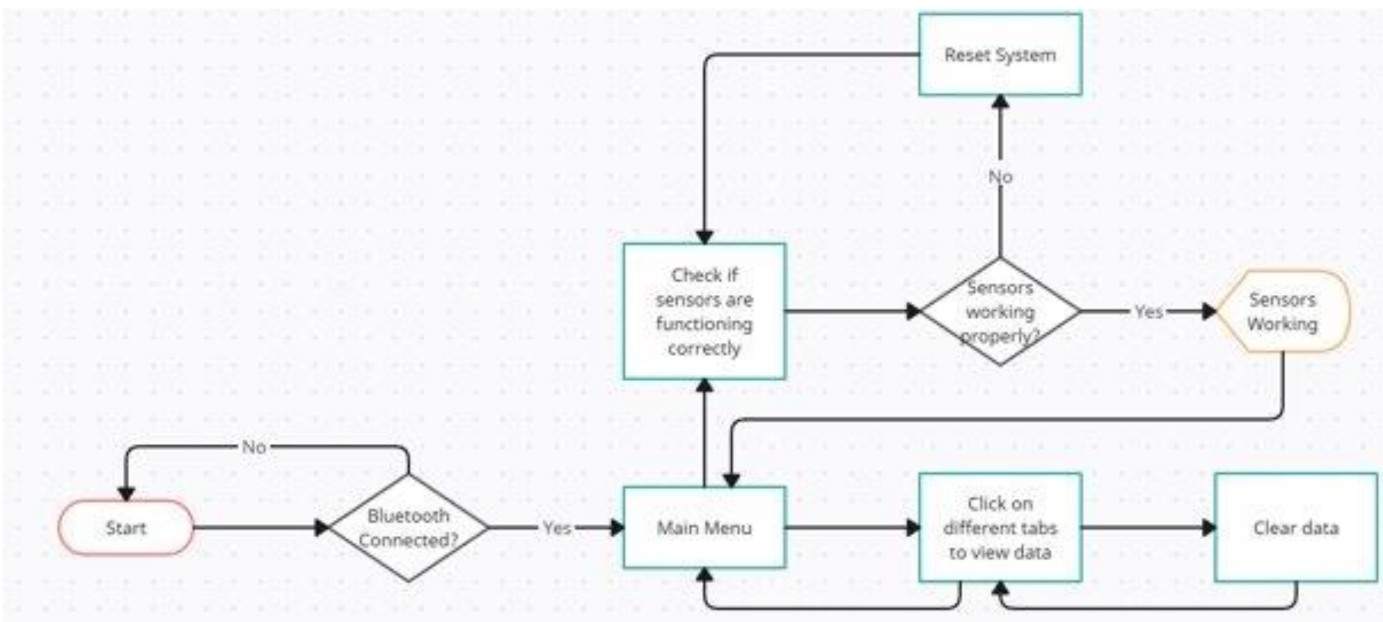


Figure 8: Level 1 Application Block Diagram

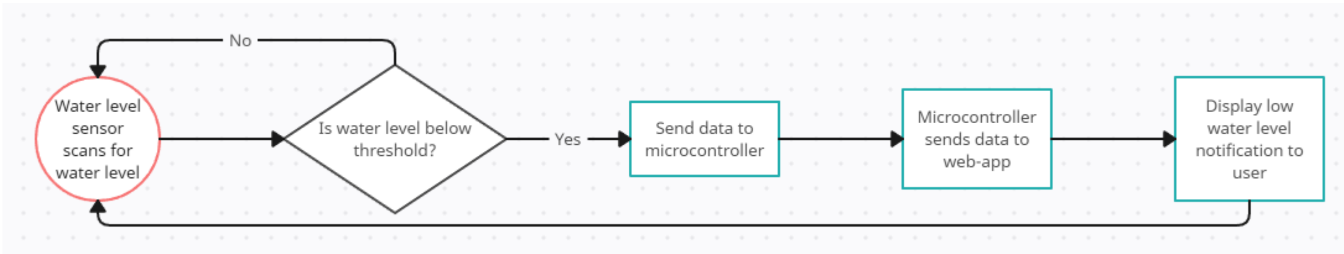


Figure 9: Water Level Sensor Flow Chart

Table 8: FR Table for Mobile Application

Module	Mobile Application
Inputs	• Database (SQLite) • Processor • User
Outputs	• Database (SQLite) • Processor • User
Functionality	The Mobile Application is set to get the data from the processor and store it in an SQLite Database. Once that data is stored, the user will be able to view it on their phone via a mobile application in various forms such as charts and graphs. The user will then do what is needed (refill reservoir, give more/less sunlight, etc.) depending on the information provided.

The software component of the project includes two subsections. One of the subsections is the Mobile application with the other subsection being the Embedded system. The embedded system will be responsible for obtaining the data from the hardware, such as the sensors, and after processing and formatting the data, will relay it to the mobile application via Bluetooth.

One of the common Bluetooth modules used is the HC-05, which is widely used with the PIC family of microcontrollers. This module uses the Bluetooth 2.0 communication protocol. It is also able to communicate using the UART standard. The Embedded system functions as stated above in section 2.7.

The mobile application aspect of the project is shown in Figure 4. As shown above, there will be a mobile application created using java as the programming language. When this application is opened, there will be an option to connect via Bluetooth to the microcontroller. The application will attempt to connect via Bluetooth, but if it is unable to, the user will need to attempt at a reconnection by closing the application and opening it again. One the Application is connected to the microcontroller, the user will be directed to a main menu. This main menu will have multiple tabs. There will be one tab per sensor, which in this case will be a minimum of 4 sensors. There will also be one more tab for checking the connection status of the sensors. Using this tab, the user will be able to check if the sensors are functioning as expected and transmitting data. If the sensors are not functioning as intended, the user will be able to reset the connection to the sensors. Once the sensors have been reset, the user will be displayed a message stating that the sensors are functioning properly. From the main menu, the user will be able to select the different tabs for each sensor, where they will be able to view sensor data in various forms, such as charts, graphs etc... There will also be an option for the user to clear the data shown. They will then be able to redirect themselves back to the main menu.

Table 8 shows the inputs and outputs as well as the functionality of the mobile application. The application will have inputs from the user, the processor, and the database. It will also be able to output data to the user, the processor and the database. The intended functionality for the mobile application is as follows: It will be able to receive data from the processor, which will be directed to the database. The data will be kept in the database until the user decides to view the data, at which point data will be retrieved from the database and displayed to the user. There will also be other data that will need to be displayed such as the low light level notification and the low water level notification.

Figure 5 shows a flow chart for the water level sensor. As shown in the figure, the primary stage is where the sensor will be scanning continuously to sense the water level. Since the water level sensor will be placed in a place that corresponds with 20 percent the total volume of water, once the sensor detects no water, it will forward that data to the microcontroller. The data will then be formatted for Bluetooth transmission and transmitted using the UART communication standard. Once the application receives the data, it will send a notification to the user stating the lack of water and the need to refill the reservoir. Once this process is completed, the cycle restarts again from the water level sensor.

The android app created on Android studio using Java will communicate with Explorer 16/32. The App will be setting up and managing Bluetooth connections with devices with the threads that listen for incoming connection and connecting Bluetooth device, disconnecting Bluetooth device and data transferring between device when it is connected via Bluetooth. The very first step of the app is to make text to let user to know what about the app and button for user to interact in the. Users will be able to see buttons and TextView to read the text and press on button to do certain action.

One package in android studio includes android.bluetooth.BluetoothAdapter represent the local device Bluetooth adapter. It lets the developer perform fundamental Bluetooth tasks such as discovering Bluetooth devices, fetching a list of paired device and creating a BluetoothServerSocket. To set up a server socket and accept a connection, android studio provides a function to call Bluetooth socket by listenUsingRfcommWithServiceRecord (String, UUID). The below snippet creates a new instance of View.onClickListener and wires the listener to the button using setOnClickListener (View.OnClickListener). As a result, when the button is pressed, it will take the user to another screen where you can see the paired device and connect with them.

```
buttonConnect.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // This is an example code to redirect to another page
            Intent intent = new Intent (MainActivity.this, SelectDeviceActivity.class);
            startActivity(intent);
        }
    });
    // Get device information to connect
    deviceAddress = getIntent().getStringExtra("deviceAddress");
```

This is an example of how the package can discover and connect to devices. After that, there are also other packages that allow for handling the transfer of data like InputStream and

OutputStream through sockets using functions like getInputStream and get OutputStream. There will also need to be functions like read(byte[]) and write(byte[]), that will let the user handle the reading and writing data to the Bluetooth transmission.

Embedded System Pseudocode/Schematics (AJ)

```
void bluetoothReceive()
{
    // Simply a check to see if bluetooth is connected
    // If so, data transmission from the board itself is available
    // If not, keep trying until it is
}
```

Figure 10: bluetoothReceive Pseudocode

This function's purpose is to be able to receive data from the Bluetooth device and the phone into the microcontroller itself to make sure that everything is connected properly, and once it is, Bluetooth data transmission may occur.

```
void bluetoothTransmit()
{
    // Sending data from the board to the user(app) via bluetooth
    // Takes in data from the different sensors which is already formatted, and sends it to the app!
}
```

Figure 11: bluetoothTransmit Pseudocode

This function's purpose is to be able to send data from the board to Bluetooth. This data is received and formatted into the board into human readable language, then finally sent up to the user through Bluetooth.

```
void sensorsOn()
{
    // Check if each sensor is responding in its "on" position
    // If so, send the data to the users phone via bluetooth to let user know that all sensors are working and app will start
    // If not, give user error as to which sensor is broken and a possible fix, or restart app message
}
```

Figure 12: sensorsOn Pseudocode

This function's purpose is to check if all the sensors are on and responding. This can be done by checking all of the default values the sensors give off and to check if they are giving those values upon Bluetooth connection, in order for the sensors to be able to give the correct data to the user.

```
void soilMoistureSensors()
{
    // Figure out a way to differentiate between different sensors,
    // possibly name them differently based on what pin they are plugged into and have a corresponding name
    // on the pin as well.
    // Take in data from all of the sensors in raw format, then change that format to human readable values
    // Percentages will most likely be used to get these values (80% moisture, etc)
    // Log this data and send it to the app, if the moisture is too low depending on the plant type (dry, in-between, wet),
    // toggle this as one of the reasons for the water pump to turn on
}
```

Figure 13: soilMoistureSensors Pseudocode

This function's purpose is to be able to get initial data from all the soil moisture sensors that are connected to the board, and to be able to first differentiate between one another. Once this is done, the function will then be able to take in data from the sensors as well as format it to a human readable condition. This data will also be one of the things that determine whether the water pump will turn on and when it will turn on, basing it off the percentage of moisture in the plants soil.

```
void waterLevelSensor()
{
    // Read in value from water level sensor, and determine wether there is enough water in the reservoir for the pump to still turn on safely
    // If not, do not allow the pump to turn on and send a notification to the user via bluetooth to refill water reservoir
    // If there is enough water, do nothing and continue checking after each use of the water pump
}
```

Figure 14: waterLevelSensor Pseudocode

This function's purpose is to see the water level sensor's status, and based on this status, to decide whether the water pump gets turned off until the reservoir is refilled, or, if the pump will stay on. It is a basic mechanism that will tell the user when to refill the water in the reservoir.

```
void photoSensor()
{
    // Read in the data from the photo sensor and if there is not enough light prior to sunset, notify the user
    // Else do nothing and keep checking intervally
}
```

Figure 15: photoSensor Pseudocode

This function's purpose is to simply tell the user if there is adequate light coming onto the plant and if not, notify the user to move the plant to a better location.

```
void temperatureSensor()
{
    // Read in the data from the temperature sensor and send it to the app via bluetooth to be displayed.
    // If the temperature gets too cold/hot(i.e. near a window sill), notify the user on the app to move the plant appropriately
}
```

Figure 16: temperatureSensor Pseudocode

This function's purpose is to calculate the temperature near the plant and to display it to the user. If the temperature is too cold or too warm, the user will be notified to possibly move the plant to a better location.

```
void waterPumpActivation()
{
    // If all of the aforementioned reasons for the water pump to turn on are true, turn it on for a certain amount of time,
    // or until moisture reaches above a certain % value
}
```

Figure 17: waterPumpActivation Pseudocode

This function's purpose is to take in all the factors that will decide whether the water pump should turn on or stay off. This will also determine the amount of time the water pump will stay on for, either a set time or until a set few variables have been met (soil moisture %, temperature, etc.).

```
void LCDScreen()
{
    // Function to display all of the data from the sensors on the LCD screen on the explorer 16/32 so if the user does not want to use
    // the app, they are not forced too and may glance at the screen which will have a scrolling ui displaying information.
}
```

Figure 18: LCDScreen Pseudocode

This function's purpose is to write words out on to the LCD screen that the explorer 16/32 has on it. This will be a user interface of sorts that will scroll through the different sensors along with the data that is being gathered from the sensors per last update. This is to make it easier for users to look at their plants' information without necessarily having to open the app on their devices.

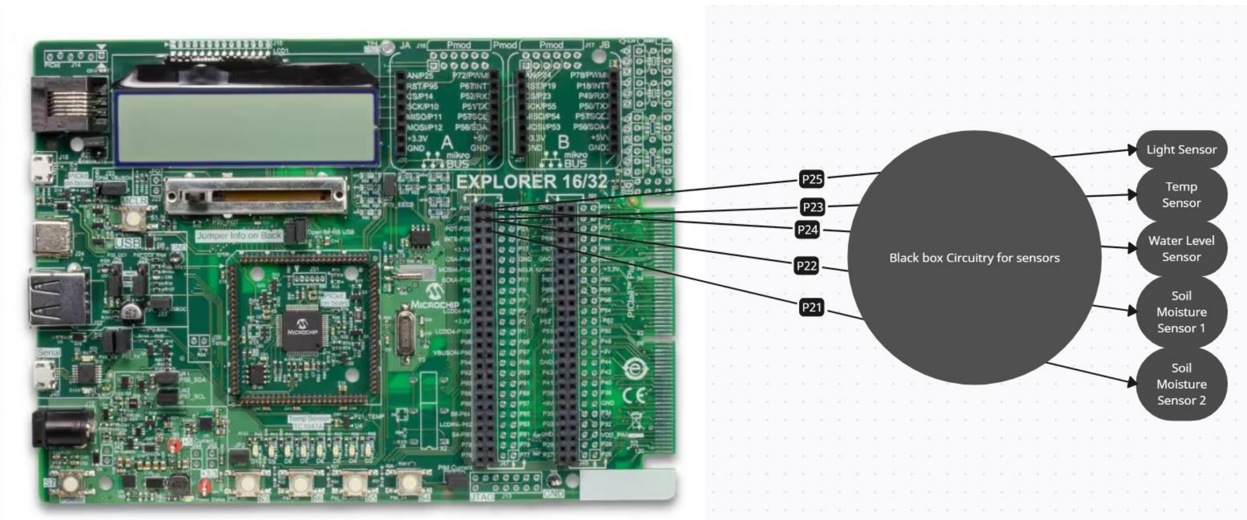


Figure 19: Sensors Schematic

All the sensors that will be used are going to be connected to the digital I/O pins on the embedded system itself. These pins that are shown in the figure above map to PORTB on the PIC24FJ128GA010 Microcontroller, allowing us to get information from the sensors and send information to them if needed.

6. Mechanical Sketch (NS)

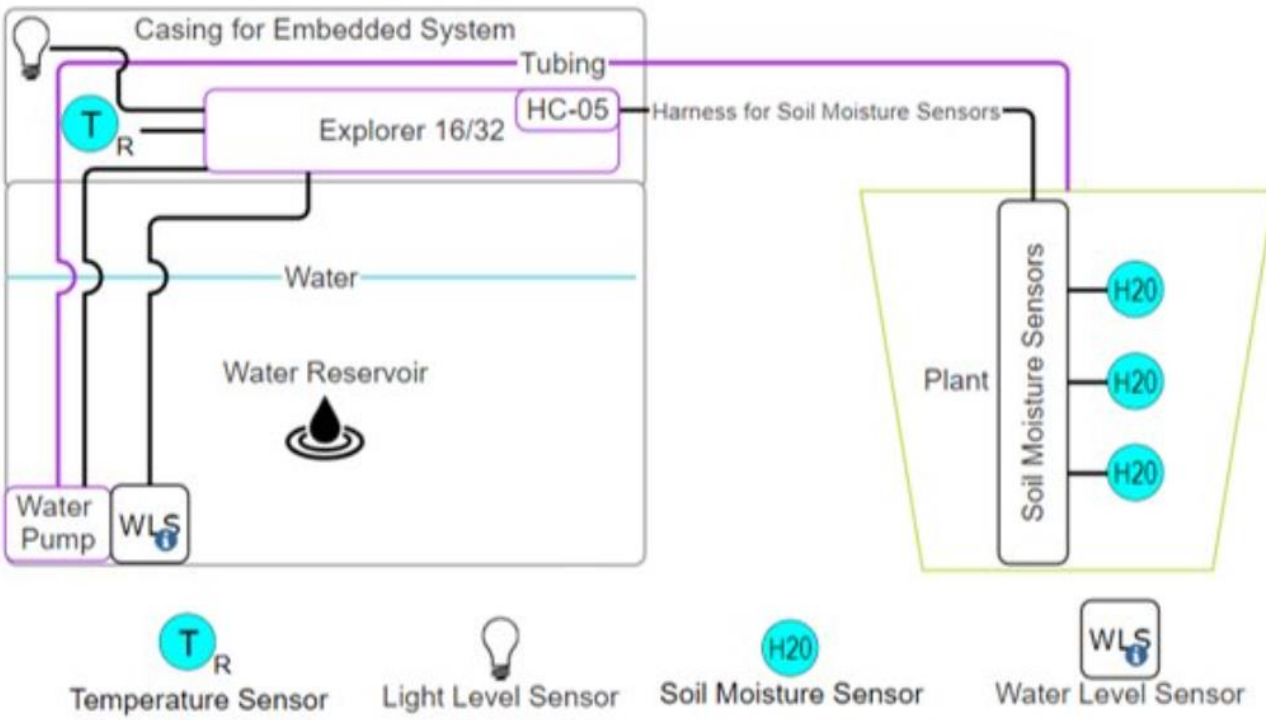


Figure 20: Mechanical Sketch

7. Team Information (AJ)

- Allen Jagic – Computer Engineering
- Madhav Khanal – Computer Engineering
- Nicolas Sabetta – Electrical Engineering

8. Parts List (AJ,MK,NS)

SENIOR DESIGN PARTS REQUEST ORDERING FORM

DT # 14 Project Leader: Nicolas Sabetta   Email: njs114@uakron.edu

Date Submitted: 12/3/2023

FA Approval ____________________

Date Ordered: ____________

SDC Approval ____________________

Ordered By: ____________

Qty.	Refdes	Part Num.	Description	Suggested Vendor	Vendor Part Num.	Catalog #/Page #/Hyperlink	Unit Cost	Total Cost
2		101020018	GROVE - WATER SENSOR-RC0603JR	DigiKey	1597-1127-ND	https://www.digikey.com/en/pr	$3.20	$6.40
2		161	Photo cell(cds Photoresistor)	DigiKey	1528-2141-ND	https://www.digikey.com/en/pr	0.95	1.90
1		4547	Submersible 3V DC water Pump- 1 Meter Vertical	Adafruit	NA	https://www.adafruit.com/prod	2.95	2.95
2		101020614	Grove- Capactive Moisture Sensor	DigiKey	1597-1010206-ND	https://www.digikey.com/en/pr	6.50	13.00
1		101020011	Grove- Temperature & Humidity Sensor	seeed studio	NA	https://www.seeedstudio.com/	5.00	5.00
1		Cuifatizviyfx8bgn	Bluetooth Module UART RS232 Serial Converter (	Amazon	B08Z3J9Y8T	https://www.amazon.com/Blue	7.75	7.75
1		lmus-revh-10	LeMotech Junction Box with Reserved Holes ABS	Amazon	B09VDBN4LP	https://www.amazon.com/LeM	20.99	20.99
2		B00009Y351	Water reservoir	Amazon	HPL827	https://www.amazon.com/Lock	16.08	32.16

Figure 21: Parts List

9. Project Schedules (AJ, MK, NS)

SDP I 2022				
Project Design	**46.38 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Midterm Report	46 days	Wed 8/30/23	Sun 10/15/23	
Problem Statement	**399.38 days**	**Wed 8/24/22**	**Wed 9/27/23**	
Need	46.38 days	Sat 8/12/23	Wed 9/27/23	
Objective	46.38 days	Sat 8/12/23	Wed 9/27/23	
Background	46.38 days	Sat 8/12/23	Wed 9/27/23	
Marketing Requirements	46.38 days	Sat 8/12/23	Wed 9/27/23	
Engineering Requirements Specification	33.38 days	Wed 8/30/23	Mon 10/2/23	
Engineering Analysis	**46.38 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Circuits (DC, AC, Power, ...)	93.38 days	Wed 8/30/23	Fri 12/1/23	
Electronics/Sensors (analog or digital)	46.38 days	Wed 8/30/23	Sun 10/15/23	
Signal Processing	93.38 days	Wed 8/30/23	Fri 12/1/23	
Communications (analog and digital)	93.38 days	Wed 8/30/23	Fri 12/1/23	
Embedded Systems	93.38 days	Wed 8/30/23	Fri 12/1/23	
Accepted Technical Design	**46.38 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Hardware Design: Phase 1	**46.38 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Hardware Block Diagrams Levels 0 thru N (w/ FF	33.38 days	Wed 8/30/23	Mon 10/2/23	
Software Design: Phase 1	**46.38 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Software Behavior Models Levels 0 thru N (w/F	33.38 days	Wed 8/30/23	Mon 10/2/23	
Mechanical Sketch	93.38 days	Wed 8/30/23	Fri 12/1/23	
Team information	28.38 days	Wed 8/30/23	Wed 9/27/23	
Project Schedules	**46 days**	**Wed 8/30/23**	**Sun 10/15/23**	
Midterm Design Gantt Chart	28.38 days	Wed 8/30/23	Wed 9/27/23	
References	46 days	Wed 8/30/23	Sun 10/15/23	
Midterm Parts Request Form	50 days	Wed 8/30/23	Thu 10/19/23	
Midterm presentation file submission	33 days	Wed 8/30/23	Mon 10/2/23	
Midterm Design Presentations Day 1	0 days	Wed 10/4/23	Wed 10/4/23	
Midterm Design Presentations Day 2	0 days	Wed 10/11/23	Wed 10/11/23	
Project Poster	14 days	Tue 10/17/23	Tue 10/31/23	
Final Design Report	47 days	Tue 10/17/23	Sun 12/3/23	3
Abstract	47 days	Tue 10/17/23	Sun 12/3/23	3
Hardware Design: Phase 2	**47 days**	**Tue 10/17/23**	**Sun 12/3/23**	**3**
Modules 1...n	**47 days**	**Tue 10/17/23**	**Sun 12/3/23**	**3**
Simulations	47.38 days	Tue 10/17/23	Sun 12/3/23	3

Figure 22: GANTT Chart Part 1

		Schematics	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Software Design: Phase 2	**47.38 days**	**Tue 10/17/23**	**Sun 12/3/23**	
		Modules 1...n	**47.38 days**	**Tue 10/17/23**	**Sun 12/3/23**	
		Code (working subsystems)	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		System integration Behavior Models	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Parts Lists	**47.38 days**	**Tue 10/17/23**	**Sun 12/3/23**	
		Parts list(s) for Schematics	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Materials Budget list	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Proposed Implementation Gantt Chart	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Conclusions and Recommendations	47.38 days	Tue 10/17/23	Sun 12/3/23	3
		Parts Request Form for Subsystems	32 days	Wed 9/27/23	Sun 10/29/23	28
		Subsystems Demonstrations Day 1	0 days	Wed 11/15/23	Wed 11/15/23	
		Subsystems Demonstrations Day 2	0 days	Wed 11/22/23	Wed 11/22/23	
		Parts Request Form for Spring Semester	0 days	Fri 12/8/23	Fri 12/8/23	31

Figure 23: GANTT Chart Part 2

10. Conclusions and Recommendations (AJ)

Overall, this project is an automated plant watering system especially meant for household plants. This project will consist of a microcontroller, sensors, a water reservoir, as well as a water pump that will all work together to gather data and send that data to a database via Bluetooth, which will then be able to send notifications to a user if their plant needs different amounts of sunlight or if the water reservoir needs refilled. The next steps in this project are to get all the sensors in and to commence hardware testing to get everything working together correctly and to commence software testing/production.

11. References (AJ, MK, NS)

12. [1] K. D. Kobayashi, A. J. Kaufman, J. Griffis, and J. McConnell, "Using houseplants to Clean Indoor Air," *ScholarSpace*, 01-Dec-2007. [Online]. Available: https://scholarspace.manoa.hawaii.edu/items/71abecd6-3d80-46fc-a05c-3cab08c3d970. [Accessed: 19-Mar-2023].

13. [2] Glória A, Dionisio C, Simões G, Cardoso J, Sebastião P. Water Management for Sustainable Irrigation Systems Using Internet-of-Things. Sensors (Basel). 2020 Mar 4;20(5):1402. doi: 10.3390/s20051402. PMID: 32143482; PMCID: PMC7085535.
14. [3] V. Pastushenko and A. Stetsenko, "Development and modeling of automated control system of agricultural cultures' water well being," *2009 10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics*, Lviv, UKraine, 2009, pp. 121-124
15. [4] L. Goldberg, J. P. Romano, and J. J. Feketa, "Automatic watering apparatus for houseplants," 27-Oct-2009.
16. [5] D. Lipford, "Tips on using Houseplant watering globes," *Today's Homeowner*, 26-Jan-2023. [Online]. Available: https://todayshomeowner.com/lawn-garden/guides/tips-on-how-to-use-houseplant-watering-globes/. [Accessed: 11-Mar-2023].
17. [6] "Smart Plant Water Meter and Health Sensor." *Sage & Sill*, Available: https://sageandsill.com/products/smart-plant-moisture-light-nutrient-and-temperature-sensor. [Accessed: 11-Mar-2023].
18. [7] D. Divani, P. Patil and S. K. Punjabi, "Automated plant Watering system," 2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), Melmaruvathur, India, 2016, pp. 180-182, doi: 10.1109/ICCPEIC.2016.7557245.
19. [8] AUTOMATED PLANT WATERING SYSTEM, by J.Marsh. (2013, Nov. 19). US 8,584,397 B1. [Online]. Available: https://patentimages.storage.googleapis.com/d7/14/d6/138603a3f43948/US8584397.pdf

20. [9] Robotic watering unit, by N.W. Anderson. (2011, Oct. 4). US 8,028,470 B2. [Online]. Available:
21. https://patentimages.storage.googleapis.com/94/c6/c7/5eb46fb66f0c73/US8028470.pdf
22. [10] C. J. T. Dinio *et al*., "Automated Water Source Scheduling System with Flow Control System," *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, Baguio City, Philippines, 2018, pp. 1-5, doi: 10.1109/HNICEM.2018.8666253.

23. Appendices (AJ, MK, NS)

Grove- Water Sensor

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/872/101020018_Web.pdf

Photocells

https://cdn-learn.adafruit.com/downloads/pdf/photocells.pdf

Grove- Capacitive Moisture Sensor

Soil Moisture.pdf

Temperature and Humidity Module

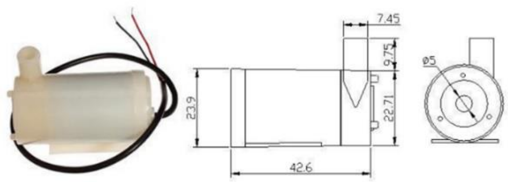
Temp sensor.pdf

Water Pump Data Sheet

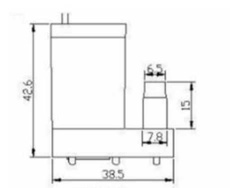
产品详情

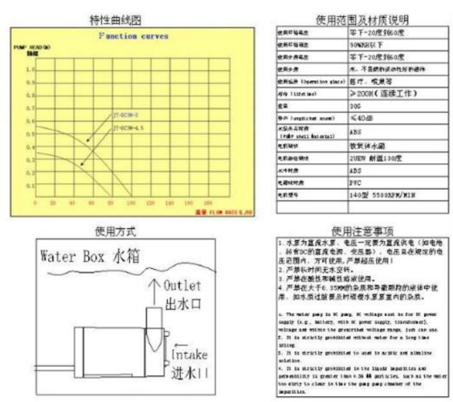
Product details

型号	电压	电流	功率	扬程	流量	启动电压	防水等级
JT-DC3W-3	3V	0.12	0.36	0.35	80	1	IP68
JT-DC3W*4.5	4.5V	0.18	0.91	0.55	100	1	IP68



型号	电压	电流	功率	扬程	流量	启动电压	防水等级
JT-DC3L-3	3V	0.12	0.36	0.35	80	1	IP68
JT-DC3L-4.5	4.5V	0.18	0.91	0.55	100	1	IP68





接口说明

红线：电机正极

黑线：电机负极

适用水管：适用于内径6mm水管

Microcontroller Datasheets:

https://ww1.microchip.com/downloads/en/DeviceDoc/Explorer_16_32_Schematics_R6_3.pdf

https://ww1.microchip.com/downloads/en/DeviceDoc/40001854A.pdf

https://microchipdeveloper.com/boards:explorer1632

HC-05 Datasheets:

https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf

https://components101.com/wireless/hc-05-bluetooth-module