Larissa Umulinga

ECE-499

Capstone Project

Advisor: Prof James Hedrick

Automated Greenhouse Watering System for the Schenectady ARC

Report Summary

Located in Albany New York, the Schenectady ARC is a non-profit with the dedication to support individuals with developmental disabilities. As a way to promote individualism amongst the people at the center, the ARC allows them to participate in activities such as the maintenance of the greenhouse. Currently, the installed water delivery system requires manual work, and knowledge of electronics to troubleshoot and maintain. Therefore, the purpose of the project is to implement a water delivery system tailored to the ARC's greenhouse in terms of use, cost, and intuitive operation for people with disabilities.

The system comprises four subunits. The main controller overlooks all the other subunits by sending instructions to their respective components. The moisture sensing ensures the collection of moisture levels in real time to base on for watering. The user input allows the user to determine the mode of operation: automatic or manual mode. The water delivery enables the safe and even watering respective to the moisture content or the user input. The newly added main controller will allow the future implementation of an easier user interface (touchscreen), a wireless collection of moisture content, and a protector case for the system. With these additional changes, the system would be able to be easily operated by people with disabilities and under no supervision. In comparison to the existing automated water delivery system on the market, the system is cost effective and meets design specifications given by Donna Vincent from the ARC. For the upcoming term, I will continue working on the project to improve the user interface as a touchscreen, collect data wirelessly and mount everything in a proper protector case.

Table of Content

Table of Figures

Table of Tables

1. Introduction

As a non profit organization, the Schenectady ARC provides resources to help individuals with disabilities. Care and support is provided to over 1000 people with a wide range of ages and developmental disabilities [6]. The center is equipped with a greenhouse which serves as a platform for the individuals at the center to acquire greenhouse maintenance skills but is also used as a means to collect funds by selling plants. The watering system of the greenhouse is provided by the Maple Ridge Center to provide enough water to sustain the moisture of the soil for the plants. From the previously implemented watering system, the controller requires manual operation implemented by switches. Currently, the system demands constant supervision and manual work to avoid the overflow of water or any other dysfunction within the watering of the plants. From discussions with Donna Vincent, further implementation is necessary. The main controller board was changed to leave room for extra features such as autonomy, a graphical user interface, and wireless communication amongst units. Each bed's watering system will be independent from its neighbor. In this case, water will be distributed at different intervals to the seedling beds in accordance to their respective plants moisture levels. In order to avoid constant supervision, the watering system will be equipped with an automatic option and, most importantly, more user friendly for the individuals at the center.

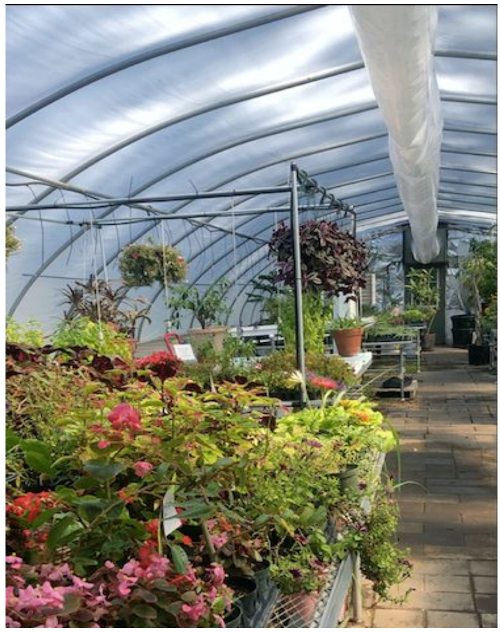

Figure 1. The Schenectady ARC water delivery system setup

2. Background

As part of the ARC New York, the Schenectady chapter was founded in 1952. The services provided by the organization range from seeking employment, acquiring medical or clinical services, finding residential services and so many more. A network of individuals, whose goals are ensuring the advocacy of the importance of people with disabilities along with their rights, was formed. With a strong belief, that with a little help, people with disabilities would be able to take their lives into their hands, and become active participants in society. The ARC acts as a platform to inhibit the idea of independence for people with disabilities by engaging them with activities that help them discover and build on their skills and hobbies. Amongst many activities, one way in which the center advocates for independence is through the participation in the greenhouse maintenance. As a result, the plants and produce, from the greenhouse, contribute to the funding of the center's future activities.

Currently, the market has automated greenhouse watering systems that perform respective to the measured moisture levels of the plants. The price of watering systems with controlled drip ranges from $1800 to $2500 [16]. The systems are expensive, hard to install and not user friendly. Operating such systems requires more intricate ways that are complex for people with disabilities. Supervisors would be required to teach them how to operate and the learning would be time consuming. In addition to the price, the systems bring about additional costs such as the installation, maintenance or replacement. Difficulties are not only financial but manual too. For example, the maintenance of a Gardena Water System (shown by Figure 2 below) requires WIFI, batteries and many more constraints that the center would not easily acquire. The current water delivery system accounts for the financial means of the ARC. but needs to be adjusted in terms of the user interface and the water seedling bed needs.

Figure 2. Gardena Water System

3. Literature review

In 2015, Qianyue Guo built a prototype of an automated watering system which consisted of a solenoid connected to a drip tube and moisture sensors. Reading levels and the calibrations were the constraints to the prototype. Similarly, Kyle Stengel designed a Linux-run controller with a touchscreen GUI in relation to the heating of the seedlings [9]. The constraint was that the user was given access to control the settings of the heating pads. Lisa Gu combined both works into a single, simple and user-friendly controller for the systems that was triggered respectively to the user input with a control of the solenoids and manual switches [10]. The controller was run manually, but the automation, a graphical user interface and a protector case were not implemented. After analyzing the existing watering system, I will be equipping the system with an autonomous mode, a wireless transfer of moisture data to the main controller, a graphical user interface, and a proper mounting in a waterproof protector case.

4. Design Specifications

From the already installed watering system, the new implemented watering system for the greenhouse should satisfy design requirements given by Donna Vincent.

4.1 Interface and accessibility

The watering system is implemented for people with intellectual and developmental disabilities. It needs to account for their needs and be easy to understand, operate and access it. Therefore, the system should not require any special knowledge to be operated, maintained or understood. In order to use the system, no instruction manual should be necessary. No aspect of it should be hard to understand such as locks, or involve a heavy physical aspect to it. Simple operations such as touching a screen or pushing a button should allow the user to control the system. In addition, the user should not have to understand the inner workings, such as the circuitry of the system, in order to run it.

Figure 3. Seedling Bench (8 by 11)

4.2 Automatic and Manual

Given that the system can not be supervised all the time, it needs to be equipped with an automatic option. The user sets a minimum threshold value which is compared to the readings from the moisture sensors. Until the moisture readings reach the threshold value, the water delivery is activated to increase it. The regulation of the water is ensured respective to the readings. A manual option should be provided that allows the user to run the watering for a desired period of time. As the system is not fully supervised, a warning will be sent in case of a malfunction. In addition, there will be a button to turn off the system when need be. The user can alternate between manual and automatic mode using a slide switch.

4.3 Sensing Moisture

Water is regulated according to the measured moisture. Air conditions and soil moisture should be measured within every 3 minutes (as it is the average time the moisture level changes) and send the collected data to the system to control water in real-time. Given that the moisture sensors are inserted in the soil, they should not harm the seedlings' growth. Seedlings are affected by both air and moisture conditions at early stages of growth, hence the systems would control the water accordingly. The moisture sensors should communicate with the main controller board wirelessly to avoid any damage to the overall system.

4.4 Performance

Moisture levels will be measured every 5 minutes to maintain the moisture levels by spreading water onto an 8’ by 11’ seedling bench. Given how large the area is, it was divided into four quadrants (like Donna Vincent had previously requested), and no quadrant should acquire more water than another. Each quadrant’s moisture readings is an indication of the level of moisture on the entire quadrant. The difference between the readings of the moisture and the levels should be insignificant.

4.5 Safety

The main controller that assembles all the components involves a lot of wiring. The controller should be enclosed to avoid any of the wires conducting high voltages that could possibly endanger any person in its surroundings. From the Schenectady ARC, the hardware design should fit in its respective place without causing any harm to other activities carried out in the greenhouse. With a graphical user interface mounted as the main controller, the contact between the user and the controller should not cause any harm such as electrocution.

4.6 Economic

The Schenectady ARC is a non-profit organization which implies that funding for the system is not as highly available. In this case, all the necessary purchases will have to be done through the provided funds from New York State Industries for the Disabled (NYSID) through the CREATE project. CREATE provides assistance to engineers to allow them to put their skills into use and solve a real-life issue through their Capstone projects.

4.7 Water Delivery

The sensitivity of the seedlings is higher during its early stages. The type of the seedlings affects the air moisture and soil conditions. Depending on the current moisture levels, water needs to be delivered at a limited level. In this case, the water coverage needs to account for the area of each quadrant for all the seedlings. The amount of water delivered is limited, or in this case can be determined by the user. Different amounts of water are needed by different seedlings such as flowers or vegetables. Water needs to be rerouted from the existing plumbing systems for the greenhouse without affecting it. The ARC plans on operating their water delivery systems when they start growing plants.

5. Top Level Design

The system comprises four major subunits which are the moisture sensing, the control system, the water delivery and the user interface as shown by Figure 1. All the subunits receive instructions from the main controller. The moisture sensing ensures collecting the moisture levels of the soils within regular times and transmitting the information in analog values. The controller receives and converts the measurements into values that will be digitally output on the interface. The user interface unit gives the user the option to set the automated and manual operation. In case of a malfunction, the unit also serves as a communicating devices by displaying error messages. The automatic control will adjust the watering system respective to the measured moisture levels from the soil. The system accepts input from the user to set the minimum threshold and compares the moisture value from the sensing unit to trigger the watering system until the threshold is achieved. The manual control allows the user to control the beginning and end times of the water delivery system using a push button.

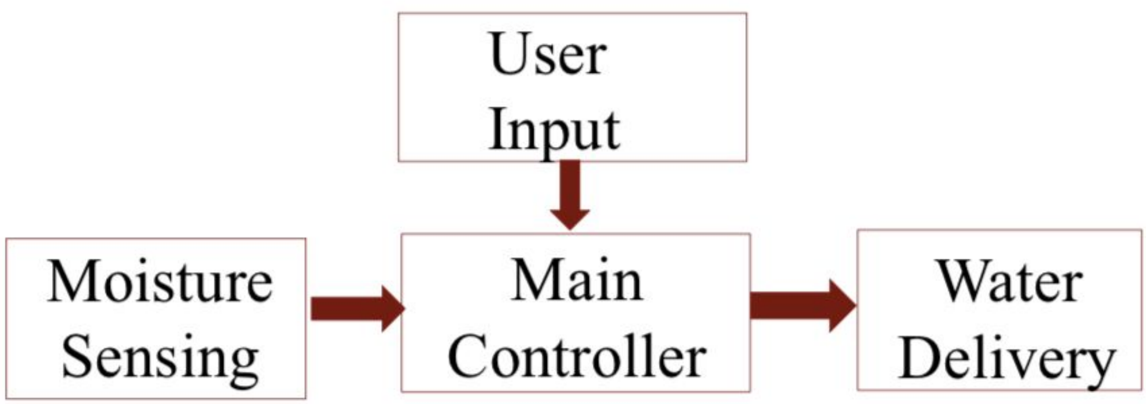

Figure 3. Top Level design of Automated Water Delivery System

6. Design Alternatives

6.1 Water Delivery

The main alternatives suggested by the market are the mat irrigation, the drip tubes and overhead misters. The mat irrigation comprises the growing of plants with the provision of constant moisture and let it stay. However, the control over the moisture levels as a response and a lack of water into air were the main constraints over the method. Drip tubing literally comes from its name. The system is made of a tube system that is positioned on the roots of the plants. The method is cost effective, precise and efficient in the conservation of water. Similar to the mat irrigation method, the growth of the seedlings is constrained by the lack of moisture in its surroundings.

The mister will need to gently provide water to the seedlings bed. The seedlings need to be in an environment with fairly high non condensing moisture level in the air; especially in its early stages. The water is absorbed by the leaves and should not be forming a rot or drying out.From previous research, the most effective method was the overhead misters in terms of use and cost. Each unit costs $10, and its maintenance, testing and replacement is easy. The solenoid valves on the systems allow it to control moisture conditions and provide atmospheric moisture.

Figure 4. Netafim Overhead Mister (Top-Sprinkler and Bottom-Mister)

6.2 Moisture Sensing

The alternatives to measure the moisture levels of the soils are the micro controller resistive moisture sensor, a tensiometer and a micro controller capacitive moisture sensor shown in Figure 5 below. The base values to indicate the desired moisture levels had been provided by Donna Vincent from testing with a tensiometer. The size, the cost, and the compatibility to multiple control systems from both sensors were given priority over the accuracy of the readings of moisture levels of the soil. The sensors were small to fit into the seedling beds, and the cost was within an affordable price for the Schenectady ARC. Due to the exposure of wires to outside conditions, the moisture data will need to be sent wirelessly to the main controller to promote independence of subunits in terms of troubleshooting and maintaining the system.

Figure 5. Sparkfun resistive moisture sensors, Spectrum Technologies irrigation tensiometer, Worldchips capacitive moisture sensors

Non wireless moisture sensing brings about a limitation in the number of moisture sensors per quadrant. With a small number of sensors per quadrant, the accuracy in readings decreases given that there is less data to average over and get a more accurate moisture level reading. In terms of distance, given that the moisture sensors would have to be physically attached to the main controller. Additional components such as wires are necessary to establish the data transmission hence exposing the system to external damages such as water to cause circuitry shortage or unnecessary unplugging from one end to another. Wireless moisture sensing was chosen to tackle all these challenges with an addition of having a lower power consumption, a fast data transfer, acting both as a transmitter/receiver, easy to troubleshoot and cost effective.

Previous trials included reverse engineering an already existing moisture sensor with a communication through analog values, but the implementation required additional components such as transmitters (hard to understand and troubleshoot) and increased the expenses necessary for the Schenectady ARC. The effective implementation involved Xbee modules shown in Figure 6 below. With a lower power consumption of 25%, the data transfer of an Xbee is of 250 kilobytes per second. It is equipped with multiple A/D converters, and has the ability to act as a node (transmitter/receiver) on a mesh network. With a baud rate of 9600, the Xbee module is fast, easy to troubleshoot and cost effective.

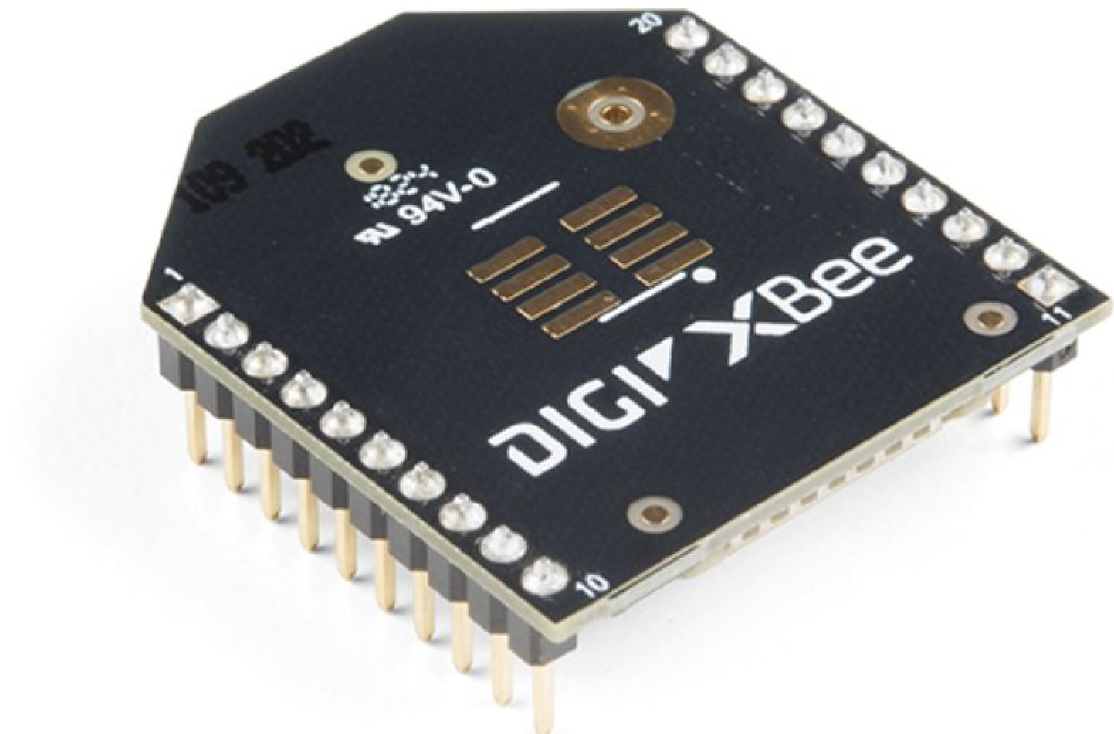

Figure 6. Digi Xbee Module 3

6.3 Automation

The greenhouse will benefit from an autonomous watering system to function under no supervision, in case the manual system is damaged, and account for people not knowledgeable about the threshold to be set to water the plants. The controller board processes data from sensors and converts into output (water being on/off) and error reporting. The controller needs at least 3 UARTs for the user input, the program controller and error reporting along with enough pins to turn on water valves and account for autonomy. The requirements of the system can be satisfied by the Arduino MEGA. It is equipped with 54 digit I/O pins from which 16 analog inputs, 16MHz crystal oscillator, a power jack, 4 UART (hardware serial ports) and more. Similar to the Arduino UNO, the board shares compatibility with most shields. The current Arduino board will be replaced to allow the system to account for extra features such as a touchscreen and more.

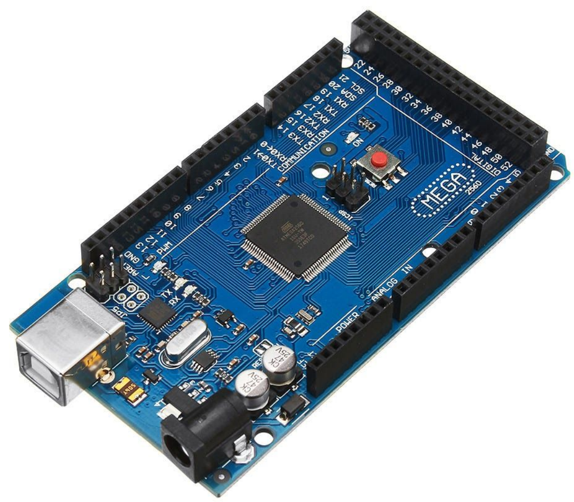

Figure 7. Arduino Mega 2560

6.4 User Interface

The user interface is important to testing and the emergency operation of the system.The implementation of a touchscreen for user interface is the most simple and intuitive alternative to operate the system. Figure 7 and 8 elaborate on the alternatives for user interface design which involve a 4x4 keypad for input from the user and a 16 x 2 LCD to display the output respectively.

Figure 8. 4x4 User Input Keypad

Figure 9. 16 x 2 LCD screen for digital output

Figure 10. Touchscreen Display for Raspberry Pi

The user input and the display of the messages or error messages will be digitally output on the screen. The controller's main board will be changed to accept more features that can handle the touchscreen. The keypad, the switch and push button will all be implemented within the touchscreen to decrease the number of unnecessary additional components. The current implementation holds these components as representation of the graphical user interface within the Raspberry Pi Model B touchscreen.

6.5 Main controller case

As an essential component of the system, the circuitry of the system needs to be protected. The case will provide safety, controlled access to electrical components, and less exposure to damage such as water. In order to avoid energy loss, electrocution or outer damage, the board will be mounted in a waterproof case. The main characteristics to choose the outer case was the material, the cost, and the melting points. The material from which the case is made matters to ensure that no water can go through.

In agreement with the other design specifications, the controller's protector case needs to be cost effective to be affordable to the Schenectady ARC for mounting or in case of replacement. Currently, the most popular cases on the market are made of aluminium or polycarbonate. As a way to promote the robustness of the system and continuous learning for the people, the main controller supports easy mounting and demounting of additional components. The outer case will need to be easily cut or melted for mounting the extra parts. Two different ways to mount extra components are cutting or melting it hence the importance of melting points.

7. Preliminary Design

Figure 9 below elaborates on the specific components of the system and the connection amongst all subunits. Each component is listed in Table 1 along with its respective subunit. The main controller will use an Arduino Mega 2560 as the connector and function instructor to the rest of the system. Figure 9 shows the current design of the system.

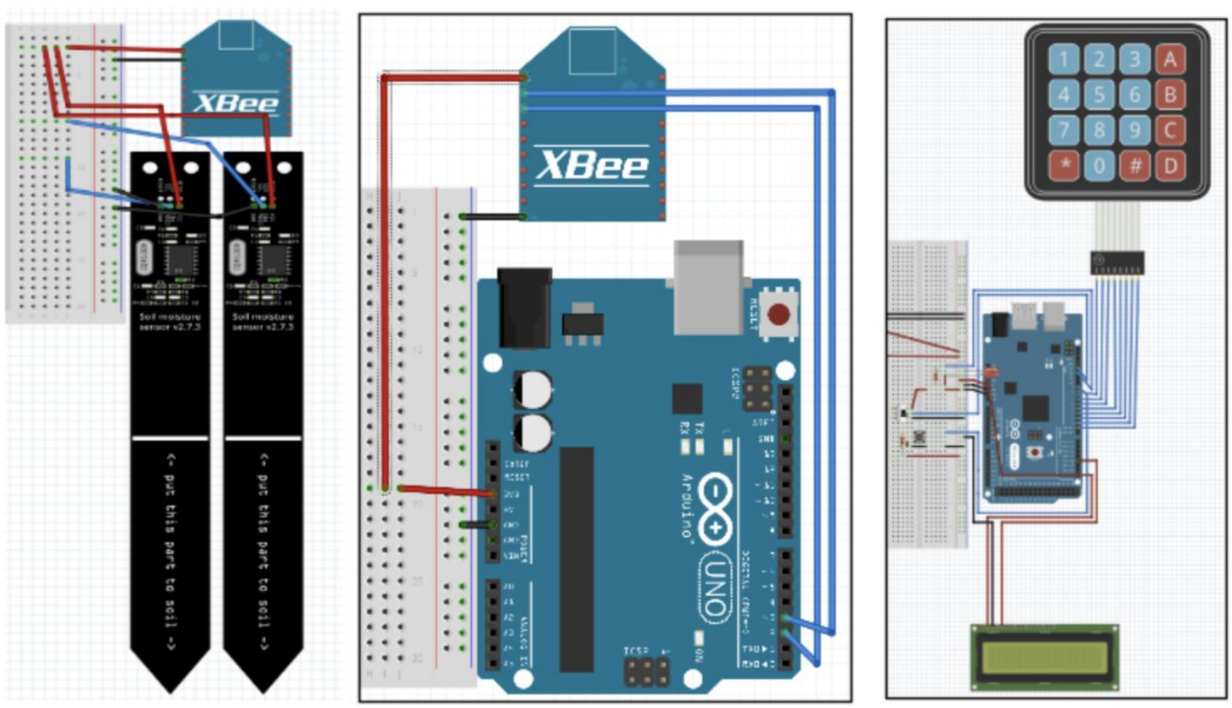

Figure 11. Schematic of Preliminary Design for the system

Table 1. Components used in the design

Subunit	Component
User interface	4x4 keypad

	Touchscreen
	Raspberry Pi Model B
	16x2 Serial Enabled LCD
	Slide switch and Push button
Water Delivery	Solenoid (WR1220) - 110 VAC
	Solid state relay (Crydom D2425)
	Netafim Mister System
Moisture Sensing	Digi Xbee Module 3 Worldchips capacitive moisture sensors
Main controller	Arduino Mega 2560

7.1 User Interface

Given that the project is being implemented in two steps, there are two different types of user interfaces. The first interface comprises an LCD screen, 4x4 keyboard, a pushbutton and an LED. The LCD prints the chosen option (manual or automatic), the moisture levels, the threshold minimum value, and error messages in case of a malfunction. On pin 13, the slide switch enables the user to alternate between the manual and automatic option.

While the system is automated, the keyboard allows the user to set the threshold minimum value from which the sensor's readings are compared to. Under the manual option, the pushbutton serves as a trigger to the water system through pin 10. The second interface is a touch screen which will act as input from the user. A Raspberry Pi Model B 3 will allow the communication between the touch screen and the Arduino board. The keypad, the pushbutton and the switch will be integrated within the touchscreen to decrease the additional components.

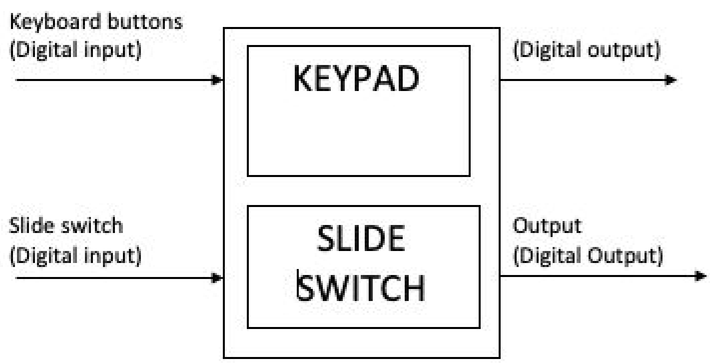

Figure 12. Overall Input and Output Schematic for User Interface Unit

7.2 Water Delivery

The beginning and end times of the delivery of the water are controlled by a 110VAC solenoid within the Netafim sprinkler. Depending on the user’s choice between the manual or automatic option, a comparison is made between the moisture level readings and the minimum threshold value from the soil. If the minimum threshold is not reached, the controller sends a HIGH signal. In order to trigger the subunit, the system requires a Crydome solid state relay. In order to complete an output circuit, the solid state relay as a switch. The solid state is powered by a signal coming from the Arduino board. At a current of 7-12mA, the relay accepts 3-32VDC for input and 24-280VAC at 10-125A. With power, the solenoid triggers the water delivery system.

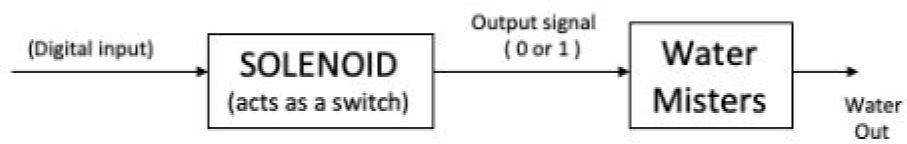

Figure 13. Overall Input and Output Schematic for Water Delivery Unit

7.3 Moisture Data Transmission

The component for moisture sensing is split in two parts : the transmitter and the coordinator. The transmitter requires a power input of 3.3 volts. The moisture level collects the moisture levels in analog values through an input pin on the Xbee module connected to the output pin of the capacitive moisture sensor. The transmitter's address is configured to match the MAC address of the coordinator. The coordinator receives the values from the transmitter through a serial connection established from a mesh network via the XCTU software. The main controller reads the moisture level from the serial connection to the coordinator and reads it through an Arduino code.

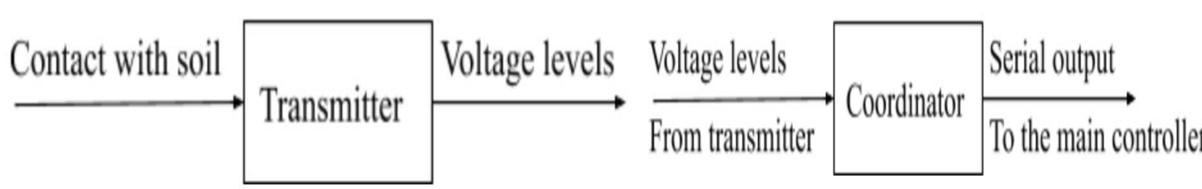

Figure 14. Overall Input and Output Schematic for Moisture Sensing Unit

7.4 Micro Controller

The microcontroller is the hub/point of connection amongst all the subunits and gives instructions accordingly. In order to function as expected, the system follows the algorithm below.

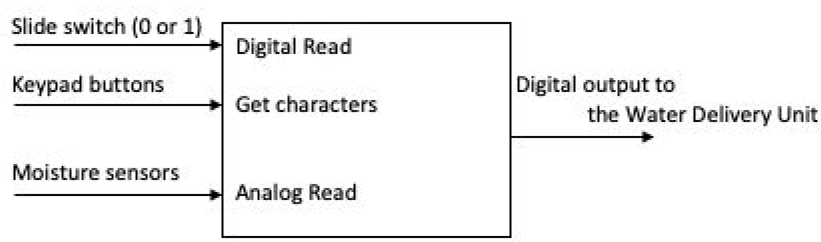

Figure 15. Overall Input and Output Schematic for Main Controller

ALGORITHM OF THE MAIN CONTROLLER

Using the keypad, the pushbutton and the slide switch:

1. Import the Keypad, Liquid Crystal and Software Serial libraries
2. Set up the Keypad instance
3. Declare all the pin variables for the led, switch and capacitor
4. Declare variable to hold the minimum threshold, capacitor values and push button signal

SETUP

1. Set up serial port for 9600 baud

2. Define all the input and output pins for the system

MAIN LOOP

1. Get all the digital readings from the capacitive moisture sensors
2. Read signal from the switch pin to specify the automatic or manual option
3. Generate the strings of the moisture levels to print on the LCD
4. Clear the screen by writing a command on Serial1
5. If operation is automatic:
    a. Set the threshold minimum value
    b. Compare the moisture levels readings to the threshold value
    c. If the moisture level is less, then send a HIGH signal to the solenoid
    d. Else send a LOW signal to the solenoid
    e. Clear the screen by writing a command on the Serial1
    f. Return to the beginning to the loop
6. If operation is manual:
    a. Print on the LCD to ask the user to input the threshold value on the keypad
    b. Get the threshold minimum value from keypad
    c. Compare the moisture levels readings to the threshold value
    d. If the moisture level is less, then send a HIGH signal to the solenoid
    e. Else send a LOW signal to the solenoid
    f. Clear the screen by writing a command on the Serial1
    g. Return to the beginning to the loop

8. Testing

For each subunit, the components were thoroughly tested separately. Each section was divided into the experiment, the results and an evaluation.

8.1 User Interface

Experiment

The slide switch was tested individually by printing the chosen operation mode on the serial monitor. If slid on the left, the serial monitor printed “Auto” and “Manual” for the right side. A similar method was used for the 4x4 keypad where the pressed numbers were printed on the serial monitor. The first pushed key is the 10th second position and the second becomes the last 10th position. The LCD’s function was tested by sending characters to be digitally printed out.

Results

On the serial monitor, the printed values matched the input coming from the keypad. The LCD screen correctly printed out each received character. The correct option was printed when the slide switch was on the left and vice versa.

Evaluation

For the LCD screen, errors occurred when the characters overlapped during the output. Each symbol needed to be delayed to allow enough time on the screen for the user to be able to be seen. Characters and lines are read differently on the screen. While printing character on a new line, similar to the method used for the serial monitor (‘println’), the LCD filled the rest of the line with a symbol. The exported LCD library provides commands to place the cursor which was used to correct the error. For the keypad, the first input added the 10th multiple of the pressed number to the threshold value which started at 0. The second key input only added the input number to the recently updated threshold value. For example, if 8 was pressed first then the threshold became 80. If 4 was pressed in second place, 4 would be added to 80, and the final value would be 84.

8.2 Water Delivery System

Experiment

The subunit was tested in terms of the triggering signal instead of the physical structure to deliver the water. The water delivery system was represented by an LED light on the preliminary design. The LED light, in parallel with a resistor, received a signal from the board. A HIGH signal symbolized the start of the trigger to the water delivery system and LOW otherwise. On the final implementation, when the HIGH signal is sent, the solenoid opens to let the water be misted out. The solenoid requires the same amount of voltage as the LED light to be opened hence the usage on the prototype. In order to ensure that the water delivery system was triggered, the threshold value was set to be higher in both the manual and automatic options. The solid state relay was setup as an additional test for triggering. Both the ssr and the LED use the same amount of voltage that would be provided at the Schenectady ARC greenhouse. While the delivery system was on, the soil was watered to increase the moisture levels.

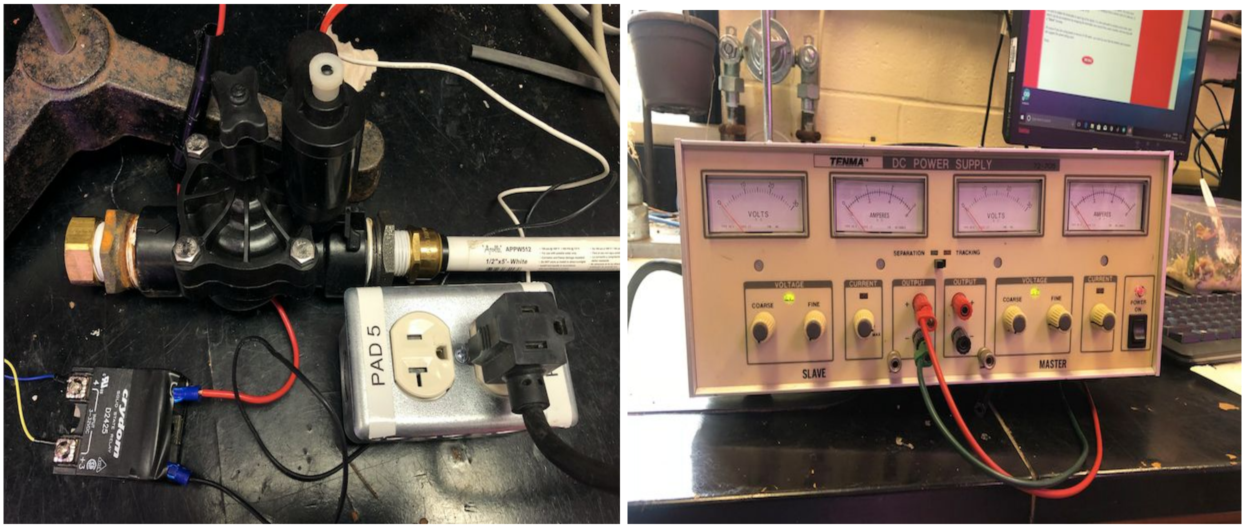

Figure 16. Water delivery system testing setup

Results

The correct signals were sent from the main controller. While the moisture levels were low, the LED light and the ssr stayed on until the moisture levels of the water reached the threshold value. The water system worked in a similar way under the automatic and manual options. Both systems (ssr and LED light) stayed intact when the moisture levels of the soil were equal or exceeded the threshold values.

Evaluation

Similar to all the other subunits, the intention remains to keep the system easy to understand and maintain. Once the water system is triggered, the LED light notifies the user while the values were digitally printed on the LCD monitor. A manual will be necessary for the maintenance of the system. The solenoid that allows the water to be delivered is an external component hence demounting will be fairly simple using the color matching process.

8.3 Moisture Data Transmission

The correlation coefficient of the Worldchips moisture sensors was found higher compared to the Sparkfun, more stable, hence used for the final design. The moisture sensors had been tested in terms of the accuracy of readings by inserting two sensors in the two different types of soil With the new implementation, the Xbee mesh network and communication between the transmitter and the coordinator was tested in terms of the speed, the range and the accuracy in readings.

Experiment

In order to increase the accuracy of the readings, the sensors connected to the transmitter were dipped into a pot acquired from the laboratory. The pot was considered fully dried from being exposed to the sun to absorb all the moisture and water content. The destination address of the transmitter was set to the coordinator's MAC address. In order to ensure that the modules were correctly communicating, both modules were connected to test grove boards (shown by Figure 17). The firmware settings were configured in the XCTU with a known sleep and wake times to transmit and receive data. For each module, the Xbee test grove boards turned on as the module was waking (with a green LED on) and transmitted showing the red led turning on (on the RSSI pin).

The leds were expected to turn on at the same time on both test grove boards for each module.
In addition, the XCTU software, the console log of the coordinator could be accessed to ensure that a frame had been received from the transmitter unit. An existing commercial moisture sensor

was used on the same pot as a way to estimate the moisture level from the analog readings as voltage levels. The transmitter unit was placed in a different room than the transmitter to test the range limitation. In addition, while readings were being received to the coordinator, the capacitive moisture sensors were dipped in water to observe the change in readings.

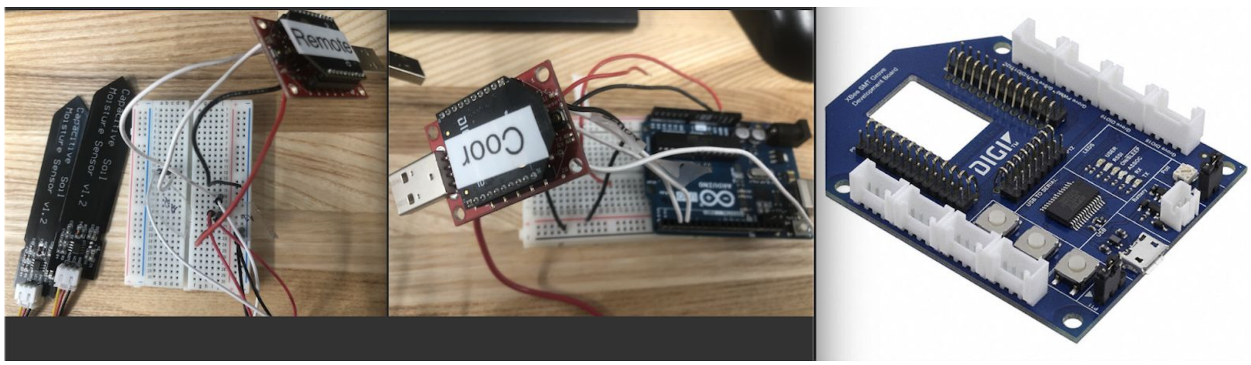

Figure 17. Moisture Sensing Testing Setup (Remote unit, Receiver unit, Xbee Test Grove boards)

Results

The capacitive moisture sensors values had accurate readings from previous experiments while dipped in pots for 6 minutes and observing the output on the LCD. For the Xbee, the range was limited to 2 meters. Whenever the setup for testing the range was further than that, then the communication between both units was completely lost. After establishing the mesh network, the communication between the modules were confirmed using the led pins turning on and off on the test groves. The led pins were synchronized as expected and the coordinator received a frame (from the console log). In terms of accuracy, the transmitter was not sending the accurate moisture levels as analog values. The error in transmission occurred at the power supply required to be 3.3 Volts and was at 1 Volt. As for the coordinator, even though the readings were not accurate, the packet was correctly read from the Arduino code.

Evaluation

The Worldchips capacitive moisture sensors had already been established as cost effective, easy to understand, time saving and easy to operate and troubleshoot. Issues rose with the Xbee mesh network setup. The Xbee module pins are very fragile which requires extreme precaution while soldering. The power was not always supplied to the module which required replacing the solder very often. The module's manufacturer had extremely poor documentation in terms of startup, operation and troubleshooting. The communication between the Xbee modules requires a mesh network which transmitted and received API frames from both ends which failed at times due to poor documentation. The power supply for both the transmitter and the receiver needed to be at 3.3 Volts to function as expected. With the need of at least two moisture sensors per quadrant, the setup will need to have the main controller placed in the middle of the setup for every transmitter to send data for moisture levels More testing needs to be made in regards to the amount of voltage required, the range, the accuracy in readings and how to increase it.

8.4 Main controller

Experiment

In terms of implementation, the initial system's Arduino board was replaced with the MEGA 2560. The new board allows the implementation of additional features such as a touchscreen. All the external components were respectively connected to the board. The controller board was tested to account for the program controller, error reporting, user input, sending the correct signal to the water valves and implementing autonomy of the system.

Results

Compared to the previous implementation, the main controller allowed the user to choose an automatic or manual option. When the slide was on one side, the system ran automatically by setting a threshold value. On the other side of the switch, the system waits for the user's input to set the threshold value. The LCD screen displayed the chosen option; automatic or manual. For both options, the water delivery system receives a HIGH signal when the moisture content levels are lower than the threshold value. The exceptions comes in for the manual option where the pushbutton needs to be pressed to release the water. As expected, the respective values were correctly digitally output on the LCD screen.

Evaluation

The implementation is fairly simple and intuitive for people with disabilities. Similar to the moisture sensing, a manual would be required in case of troubleshooting the system. In real time, the system updates the user on the moisture levels of the seedlings. Manual control allows the user to set the threshold value which properly determines whether the water system will be triggered if the moisture levels are less. The solenoid is opened which allows the water to be evenly spread on the seedlings. Differently, the automatic control sets the threshold value but triggers the water delivery system similar to the manual control. It compares the measured values to the threshold and sends a HIGH signal if less and LOW if otherwise. Upon future implementation, the heating system can be added onto the new board.

9. User Manual for current design

Startup

When the power is supplied to the system, the moisture content of the soil is digitally output on the LCD screen. The main options of running the water delivery system are manual and automatic. The slide switch is slid left for automatic override and right for manual option. Once on the left, the system runs itself hence no further input required. Otherwise, the user needs to set the threshold value on the keypad. The user needs to type one key at a time to set the threshold value.

Operation

While the automatic option is chosen, the user sets the threshold value using the keypad. The mister for delivering the water relies on the comparison between the measured moisture values and the threshold value. If the levels are lower, the Netafim misters will turn on until the moisture content raises to the threshold value. If the levels are lower, the misters will turn on until the moisture content raises to the threshold. When the switch instructs the rest of the subunits of the manual option, the user sets the threshold value. The misters are not turned on until the pushbutton is pressed with no consideration of the readings from the soil. All the values are digitally output on the LCD screen in real time.

Troubleshoot and Maintenance

Electricity and water should be turned off at all times during replacement or troubleshoot. Upon any malfunction, all the components should be easily detachable from the main controller with no effect on the rest of the system. Multiple components will be provided to the Schenectady ARC to account for such situations. A constant check for damage or repair should be done for the moisture sensors given the exposure to external harsh conditions such as ultraviolet light or water. For most of the components, the color at the end of the electrical wiring connector matches where it needs to be plugged on the main controller. In case of a malfunction, turn off the power supply, detach the damaged component and attach a new one by matching the respective colors. The outer case of the controller should be cleaned regularly and checked for degradation. In case of a leak in the system, dry the system properly and restart it. In case of any unknown circumstances, contact the greenhouse administrators or Union College for help.

Winter term (Initial plan) :

- Week 1: Test the manual and automatic option as two separate systems both on the prototype and at the greenhouse.
- Week 2: Visit the greenhouse to test the new main controller with the Netafim Sprinkler system and test the wireless sensors
- Week 4: Ordering the Raspberry PI to implement the touchscreen and the protector case
- Week 5: Setting up the Raspberry to the main controller to implement the touchscreen
- Week 6:Testing the touchscreen both on the prototype and at the greenhouse
- Week 7: Conduct calibration tests and enclosing the system in the protector case

- Week 8: ECE 499 Demonstration and Presentation
- Week 9-10: Assemble all the components together for manual and automatic operation
- ECE 499 Final Report (each part of the paper is written per week)

Winter term (Current plan):

- Week 1 : Visualizing the implementation plan for the term as well as ordering the components necessary for building the prototype.
- Week 2 : Test the manual and automatic option as two separate systems both on the prototype.
- Week 3 : Ordering the Digital Soil Moisture Meter and Freeze Alert (commercial unit) and the RF Link Receiver for reverse engineering an existing wireless moisture sensor.
- Week 3: Reverse engineering the wireless moisture with a 434 Mhz Sparkfun receiver to decode an RF signal from bits to moisture levels.
- Week 4 : Intensive research, testing different setups with the commercial unit, realizing that the commercial unit is not feasible, and redesigning a solution for wireless moisture sensing.
- Week 5 : Ordering the components for the new design and laying out an implementation plan (online schematics). Intensive research and acquiring the documentation for the Xbee modules, test grove boards, and Sparkfun USB dongles.
- Week 6 : Setting up the connection between the Xbee modules through a mesh network.
- Week 7 : Implementing the remote unit (the Xbee module) to collect moisture content as voltage levels, and read them through the XCTU software.

- Week 8 : Implementing the coordinator unit (Xbee with the Arduino Mega 2560) to receive an API frame from the remote system wireless and be read through serial communication from the main controller.
- Week 9: Oral presentation to the ECBE departments and continuing implementation
- Week 10 -11: Reassembling all the units together (the remote and the transmitter), as well as the user interface, and an LED to symbolize the trigger to the water system given that this is a prototype.

11. Cost Analysis

Table 2 below consists of the total cost from all the components used to implement the automated water delivery system. The first prototype was solely built from donated components from previous designs. The upcoming implementation involves the listed parts below with a total of $293.44. Upon completion of the project, the items will be given to the school for further implementation of prototype testing.

Table 2. Detailed Cost of components for the system

Component	Quantity	Cost	Vendor
Xbee 3 Module	2	$71.80	Sparkfun

Xbee Explorer Dongle	2	$51.90	
Worldchips Capacitive Moisture Sensors	2	$77.88	Amazon
	TOTAL		$201.58

The water delivery was not included in the estimation of the total because the former design was not altered. In addition to meeting the greenhouse's needs, the total cost is considerably less compared to other automated water delivery system currently on the market.

12. Reflection

The original design kept fluctuating in order to find the best model of the system fit to the people in the Schenectady ARC. The main focus was towards a replacement of the main to account for automation, error reporting, and a wireless acquisition of moisture content of the water delivery system for the Schenectady greenhouse. In order to increase production efficiency, deadlines were set per week for specific tasks. External conditions such as damaged parts, circuitry and coding errors were not accounted for which delayed the initial timeline of completion. Multiple difficulties arose such as getting all ordered parts, planning a visit to the Schenectady ARC within a feasible time, acquiring measurements from the greenhouse and testing out the prototypes, and separating the manual and automatic options of the system. In addition, the first most fit design ended up not satisfying the design specification and not fully functioning hence new solutions had to be implemented. Additionally, as per Donna's request, the schedule had to be speeded up because it was expected to be functioning in January for the planting of plants.

13. Conclusion

- Lessons Learned

Such projects showcase the appliance of knowledge from class which is an important tool of a Capstone project. The implementation of the project encompasses multiple teachable aspects. I learned valuable skills such as the design process which revolved around dealing with a customer, noting down the requirements and designing a project that meets them respectively. In terms of the hardware, I acquired more circuitry skills in terms of debugging, rewiring, mounting, and testing. New components, that I had not been exposed to before, were used in the design. In this case, I refined my research skills on the usage of various electrical components, calibrating and testing them, as well as wiring in accordance with their datasheet. On the software side, I learned more on coding in Arduino along with serial communication between the hardware and software. Even though setting a timeline is important during the design process, I learned that it is equally necessary to account for testing time for the system. Similarly, it is important to understand the possible changes of the overall design during the implementation process. The original design is open to various tweaks to meet the design specifications.

- Discussion

One of the main goals of the implementation of the automated water delivery system is to promote independence amongst the individuals at the Schenectady ARC. Fair intuition for the users and less supervision of the system were the main aspects pondered upon. Respective to the user's input, the main controller coordinates the instructions sent to the rest of the system. The functionality of the system expanded in terms of how it operates. Currently, no supervision is necessary if the automatic option is activated. Digital values are directly read from the moisture sensing unit to dictate the trigger of the water delivery system. Otherwise, the seedling trays are watered if the moisture levels are less than the threshold value input by the user. The new main controller board promotes the installments of more components for future work. The prototype allows the right signal to be sent to the solenoid, to act as a switch, to allow the water to be misted out.

- Future Work

For the next term, the main focus will be performing more tests on the wireless moisture sensing unit to get accurate readings, implementing a graphical user interface and enclosing the entire system in the protector case. The current prototype, with the new board for the main controller, will need to be tested with the greenhouse watering system for compatibility and measurement purposes. I would like to visit the greenhouse more to acquire measurements for the seedling trays in order to estimate the number of sensors needed per tray. The user interface will be replaced with a graphical user interface; a touch screen mounted on a Raspberry Pi. All the interface subsystems such as the keypad, the slide switch and the LCD will be integrated within the touchscreen. Given the exposure to water and other external conditions, I will focus properly on the GUI, wiring and mounting, and waterproofing the circuitry. The outer protector case will satisfy all these requirements. With enough time, more prototypes could be built to function for multiple different greenhouses.

14. Acknowledgements

Thanks to Prof James Hedrick for impeccable advising and help throughout the project. The project would not be what it is without your expertise.

Thanks to Donna Vincent and the Schenectady ARC allowing us to contribute to the community positively.

Thanks to NYSID and CREATE for providing funds for the implementation of the project.

Thanks to the Engineering Department, especially Prof Helen Hanson, for supervising the planning of the Engineering Capstone projects.

Thanks to Julius Madey for his incredible advising through the implementation of the project.

15. References

[1] Adams, Derek. “The Best Garden Water Systems 2019: from Auto Timers to Self-Watering Planters and Irrigation Systems.” *T3*, T3, 4 May 2019, www.t3.com/features/best-garden-water-systems.

[2] “Polycase.” *Waterproof Boxes & Weatherproof NEMA Enclosures for Electronics*, www.polycase.com/waterproof-enclosures.

[3] “Aluminum Alloy.” *MakeItFrom.com*, www.makeitfrom.com/compare/6061-AlMg1SiCu-3.3214-H20-A96061-Aluminum/Polycarbonate-PC.

[4] *Arduino Mega 2560 Rev3*, store.arduino.cc/usa/mega-2560-r3.

[5] Crydom Datasheet: http://www.crydom.com/en/products/catalog/series-1-240-ac-panel-mount.pdf

[6] *“About Schenectady ARC”,* The Schenectady ARC, , https://www.arcschenectady.org/about.html

[7] “Timeline of the Individuals with Disabilities Education Act (IDEA).” *Online Teaching Degrees from the University of Kansas*, educationonline.ku.edu/community/idea-timeline.

[8] Admin. “The Psychology of Color: How Do Colors Influence Learning?” *The Psychology of Color: How Do Colors Influence Learning?*, www.shiftelearning.com/blog/how-do-colors-influence-learning.

[9] Stengel, Kyle, Automated Greenhouse Heating System. 2018.

[10] Gu, Lisa, Automated Greenhouse for Schenectady ARC, 2019.

[11] CREATE: Cultivating Resources for Employment with Assistive Technology, https://www.createnysid.net/guidelines.htmlhttps://www.createnysid.net/guidelines.html

[12] Raspberry Pi, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[13] Srivastava, Lalit M. “Seedling Growth.” *Seedling Growth - an Overview | ScienceDirect Topics*, 2002, www.sciencedirect.com/topics/agricultural-and-biological-sciences/seedling-growth.

[14] ray. “RAYSHOBBY.NET.” *RAYSHOBBYNET RSS*, 23 July 2014, rayshobby.net/wordpress/reverse-engineer-a-cheap-wireless-soil-moisture-sensor/.

[15]

"Digi XBee 3 Zigbee 3." *Digi XBee 3 Zigbee 3.0 | Digi International*, www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-modules/xbee3-zigbee-3.

[16] "Learn How Much It Costs to Install a Sprinkler System." *HomeAdvisor*, www.homeadvisor.com/cost/lawn-and-garden/install-a-sprinkler-system/.

16. Appendix

```
//https://github.com/andrewrapp/xbee-arduino/wiki/Developers-Guide
// https://www.digi.com/support/forum/54079/arduino-reading-frame-from-xbee-co-ordinator
//https://core-electronics.com.au/tutorials/how-to-network-xbee-and-arduino.html

#define cmd 0xFE
#define cls 0x01

#include <SoftwareSerial.h>
#include<Keypad.h>
#include <LiquidCrystal.h>
//SoftwareSerial mySerial(2, 3); // RX, TX
SoftwareSerial mySerial(18, 19); // RX, TX


/* Setting up for the keypad*/
const byte numRows=4;
const byte numCols=4;
char keymap[numRows][numCols]={
  {'1','4','7','*'},
  {'2','5','8','0'},
  {'3','6','9','#'},
  {'A','B','C','D'}
};

byte rowPins[numRows]={5,4,3,2};
byte colPins[numCols]={9,8,7,6};
Keypad myKeypad=Keypad(makeKeymap(keymap),rowPins,colPins,numRows,numCols);


int dataLength;
```

```
int switchPin=13;
int ledPin=12;
/*Declaring values used in other functions*/
int val=0;
int on=HIGH;
int off=LOW;
int reading;
int var1, var2;
int thresholdmin;
int sensorValue1 = 0;
int sensorValue2 = 0;
int pushButtonPin=10;
char var1string[4], var2string[4];
int x=0;
int m=500;


void setup()
{

  Serial.begin(9600);
  //mySerial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600);

  pinMode(switchPin, INPUT);
  pinMode(pushButtonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}


void loop()
{
  reading=digitalRead(switchPin);
  //Serial2.print("AD");


  /*Print on monitor the measured values*/
  byte dataList;
  byte highByte,lowByte;
  int value; // A/D value from A/D0
  if (mySerial.available()>20){
    if(mySerial.read()== 0x7E){
```

```
      // read 2 bytes which are the length of the packet in bytes
      byte data1 = mySerial.read();
      byte data2 = mySerial.read(); //16
      dataLength = (data1<<8)|data2; //22
      for (int i=0; i<=dataLength; i++){
        dataList = mySerial.read();
        if (i==16)
        {
          //Serial.println(dataList,HEX);
          highByte=dataList;
        }
        if (i==17)
        {
          //Serial.println(dataList,HEX);
          lowByte=dataList;
//          value=(highByte<<8)|(lowByte);
//          Serial.print("A/D value :");
//          Serial.print(value);
//          Serial2.print("A/D value :");
//          Serial2.print(value);
//          Serial.println("");
          clear_screen();
        }


      }

    }
    value=(highByte<<8)|(lowByte);
  Serial.print("A/D value :");
  Serial.println(value);

  }

  //Serial.println();

  /* Automatic mode on */
  if (reading!=on) {
    thresholdmin=1000;
    Serial.println("Automatic Mode");
    Serial2.print("AUTO: ");
    Serial.print("Moisture level: ");
    Serial.println(value);
```

```
    Serial.print("Threshold value: ");
    Serial.println(thresholdmin);
    delay(1000);
    clear_screen();

    trigger_watersystem(value,thresholdmin);
    //m+=10;
  }

  else {
    Serial.println("Manual Mode");
    Serial2.print("MANUAL");
    Serial.print("Moisture level: ");
    Serial.println(value);
    clear_screen();
    Serial2.print("Input Threshold");
    clear_screen();
    thresholdmin=get_thresholdmin();
    Serial.print("Threshold value : ");
    Serial.println(thresholdmin);
    // m=500
    trigger_watersystem(value,thresholdmin);

  }
}

void clear_screen()
{
   Serial2.write(cmd);
   Serial2.write(cls);
}
/* Calculating the thresholdmin from the input key */
int get_thresholdmin ()
{
  //Serial.print("here");
  int thresholdmin=0;
  char threshold1=myKeypad.getKey();
  while(threshold1==NO_KEY){
    threshold1=myKeypad.getKey();
  }
  switch(threshold1){
    case '1':
      thresholdmin+=10;
```

```
    break;
  case '2':
    thresholdmin+=20;
    break;
  case '3':
    thresholdmin+=30;
    break;
  case '4':
    thresholdmin+=40;
    break;
  case '5':
    thresholdmin+=50;
    break;
  case '6':
    thresholdmin+=60;
    break;
  case '7':
    thresholdmin+=70;
    break;
  case '8':
    thresholdmin+=80;
    break;
  case '9':
    thresholdmin+=90;
    break;
  default:
    thresholdmin+=0;
}

char threshold2=myKeypad.getKey();
while(threshold2 == NO_KEY){
  threshold2=myKeypad.getKey();
}
switch(threshold2){
  case '1':
    thresholdmin+=1;
    break;
  case '2':
    thresholdmin+=2;
    break;
  case '3':
    thresholdmin+=3;
    break;
```

```
    case '4':
      thresholdmin+=4;
      break;
    case '5':
      thresholdmin+=5;
      break;
    case '6':
      thresholdmin+=6;
      break;
    case '7':
      thresholdmin+=7;
      break;
    case '8':
      thresholdmin+=8;
      break;
    case '9':
      thresholdmin+=9;
      break;
    default:
      thresholdmin+=0;
  }
  return thresholdmin;

}

/* LED on means water system is on, and off otherwise*/
void trigger_watersystem(int sensorValue1, int thresholdmin)
{
  if (sensorValue1<thresholdmin)
    {
      digitalWrite (ledPin, on);   }

  else
    {  digitalWrite (ledPin, off);
    }
}
```