MOIST (Mesh-Operated Irrigation Sensor Technology)

Ahmed Hamdy, Elange Pierre, Gabriel Santos

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — At the forefront of today's standard irrigation systems for open fields lies the sprinkler system: a both labor and time-efficient group of devices that waters grass and plants on its own without the need of human supervision, for the most part. As technology progresses, new ways to accomplish just about any task are constantly discovered. MOIST (Mesh Operating Irrigation Sensor Technology) focuses on innovating the modern irrigation system. The format in which most sprinklers currently operate has its advantages in terms of autonomy but has major repercussions in terms of efficiency and power consumption.**

***Index Terms* — Mesh network, wireless sensor nodes, wireless sensor network, radio frequency, energy consumption, web services, wireless communication**

I. Introduction

According to the EPA (Environmental Protection Agency), around half of the water currently used to water plants and grass outdoors is wasted because of evaporation, wind, or overwatering [1]. MOIST aims to be vastly more efficient than current sprinkler systems due to its ability to respond to changes in moisture in the grass it is overseeing. Sensors will be placed throughout the plane of grass and will be distributed evenly enough so that moisture in all sections of the grass is accounted for. Instead of the grass being watered for the same amount of time every week like they currently are with sprinklers, the sensors will give out a moisture reading on an hourly basis stating whether each area of the grass has consumed enough water for the day or not. If the moisture readings are too low, the network of sensors will work together to send out a signal to a watering robot detailing what section of the field needs to be watered and how much watering the section needs based on the moisture level. A company we have consulted with, Guard Dog Valves, has already worked with a senior design group in Fall 2017 to produce moisture sensors that report the level of moisture in soil to a central hub. MOIST focuses on implementing this sensor into a mesh network that can be further implemented by adding a mechanical motion of water to the system. Guard Dog Valves intends to, one day, have their technology efficiently watering large areas of grass, such as golf courses and commercial real estate. The area of study for this project, however, will be on a 20 by 30 feet backyard area comprised of grass native to the average backyard of a residential home in Florida for the sake of time and cost efficiency. With that being said our research and development of this technology will take it beyond everyday irrigation processes and impact the environment in a positive manner. With the evolution of the Internet of Things and Smart Automation, current irrigation systems is one more application most landscapes use today that require human supervision unnecessarily, but with implementation of MOIST irrigation will result in less energy consumption, less waste of resources, and less costs overall, making it a solution that will surely become more popular as such systems become more commercialized and prominent in the irrigation market. Although MOIST is considered a stand-alone project, it was initially intended to be a part of a three-group interdisciplinary project consisting of two teams of computer engineering and electrical engineering students, and one mechanical engineering group. The other groups worked to oversee creating and commanding a watering robot that will water different areas of grass at a time based on the readings that MOIST sensors relay out. This system was designed to be an add-on to common sprinkler systems. With considerations taken, MOIST aims to be the best solution for the next generation of irrigation methods.

II. Research

Comparative studies of commercial solutions has been carried out for small scale farms. The solutions that incorporate sensors do not include systems based on fuzzy logic which allow them to establish the watering quantities in a precise way. Aifro WaterEco considers climatology in order to lower or increase irrigation but it is focused on the definition of threshold values and does not include fuzzy logic or sensors, such as soil and land humidity. Blossom encompasses crop irrigation and generation of calendars, depending on the climate these calendars can be edited manually, it has common functionalities but allows for remote management, it also does not include fuzzy logic in its behavior. BlueSpray includes seasonal information to adjust irrigation as in the previous example, it does not include fuzzy logic-based behavior. GreenIQ and

IrrigationCaddy are conventional programs that can be managed remotely from mobile applications and include the feature of creating irrigation calendars. Lono incorporates threshold values and seasonal information and reduces crop watering according to the thresholds, as in the previous cases it does not include fuzzy logic and does not have weather sensors. On the other hand, the Orbit B-Hyve system incorporates a control through smartphones that is able to change some parameters in order to edit the system schedule. The parameters that device takes into account when configuring the irrigation timer are: the slope of the site, the soil type, if it is in the sun or shade, history of rainfall in the area and the current weather. The Rachio Smart Sprinkler Controller system also has a Wi-Fi connection and is able to send the data from the sensor to the user's smartphone. This device requires an initial configuration which is established by indicating the type of crop and the type of soil. In this way, the system can estimate the irrigation time required by the crop. The fuzzy system is not applied, nor are the flexible rules. Rainmachine is another commercial system which incorporates an automatic irrigation program. It is capable of calculating the percentage of evaporation and transpiration of the soil, according to the weather conditions obtained from the data of the meteorological service. This system, like the others, does not include fuzzy knowledge. The Spruce irrigation system combines the information obtained from all the temperature and humidity sensors and rainfall forecasts. Lastly, we list the Rain Commander system for its ease of use and its integration with mobile devices for remote irrigation control. However, this system lacks an intelligent configuration, it has no fuzzy logic rules, and only considers the schedule and the irrigation time that has been configured manually by the user [2].

Past senior design projects like F.A.R.M. (Fundamental Agriculture Resource Monitor) offered a low powered, wireless, plant monitoring system. Although it achieves its goal, its design concentrated more on studying the plant for gardens and pots to print out reports for its users. Our system needed to achieve that key point in addition to automated irrigation aimed for scalable landscape while withstanding weather climates and prone accidents. It is the project that we based the concept of MOIST on. Also, our project focuses less on providing a farmer or gardener with various conditions that their plants are experiencing, and hones down specifically on the moisture level of plants. However, with the loss of generality this project undertakes compared to F.A.R.M., it makes up for in the improvement of water efficiency for grass. MOIST only focuses on water moisture in grass, but the involvement with a mesh network to send the reading levels of each node in a field's moisture to a central hub and have only these specific areas watered is what makes this project the next step toward extreme water efficiency in the future. In addition, this project allows us to only water different areas of grass for as long as the node has a low moisture level. This feature would allow gardeners using the F.A.R.M. system who don't have too much time on their hands to have their plants watered for them for the right amount of time and with the right amount of water while they are away from their gardens. Irrigated Water Monitoring System, a project previously sponsored by Guard Dog Valves was aimed for this purpose. Initially presented by the UCF Department of Mechanical and Aerospace Engineering focused on the material of the sensor probe. Their design research involved extensive testing of possible material used for an anode and a cathode of the device. The study and design included a correlating resistive value and moisture level. Our intent is to take this designed sensor and improve on the electronics and network. Thus, helping a state-of-the-art irrigation system be created and come into fruition.

III. Hardware Components

Considering previous designs, we came up with a list of major components of a node we needed to further investigate. These subparts include the following: A sustainable power system, that does not need constant maintenance; a central controller dedicated to keep the operations and program running; a communication system, to link with the network; a sensor for reading measurements; and an enclosure to house all of this. This single PCB called a node is a design produced in scale. Figure 1 details the individual parts and their links.

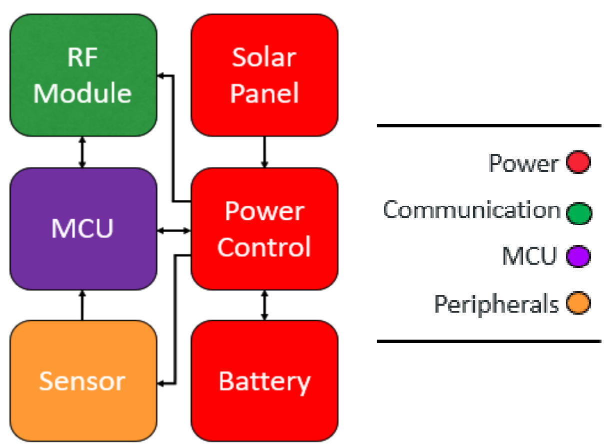


Figure 1 - Node Block Diagram

In order to maintain the functionality of our project, an independent power system is required to supply the main key components of our system such as: solar panel which is comprised of solar cell that convert light energy to electrical energy; charge controller that threshold the output voltage from the solar panel going into the battery; the voltage regulator that regulate to a non-changing voltage; the battery that provide power directly to the components. solar panel is available in three types (monocrystalline, polycrystalline, and amorphous). Between the three-solar panel technology, amorphous is the cheapest but also the least efficient; the polycrystalline combine efficiency and cost which is the best choice for our project because of the high cost of monocrystalline. Because of the portability of the project the parameters for choice of the battery are crucial such as size, voltage, spillage free, and environmentally-friendly.

For our solar panel, the NUZAMAS provides 2W at a 12V output. It is a poly-crystalline solar panel. It is 17% efficient in power retention, and resistant to corrosion and humidity, which is ideal for our sensors out in the field. For our node battery, the Customized 18650 was chosen. This is a Li-Ion rechargeable battery meant to power small devices. Has a nominal capacity of 3.7V with a charge capacity of 2600mAh with a cut-off of 3.0 V. The 18650 will provide the power to the system which require no more than 3.3 volt for the microcontroller. The 46.5 g weight, the 18 mm diameter, and 65 mm height dimension of the battery make it very suitable for the portability of the project. The discharge capacity of the cell with 1.3 amperes down to 3.0 volt is 1 hour which in our case might be more than 3 hours because of our low power system design. In order to mitigate any power conundrum, we have decided to pair the battery with a solar panel which will constitute as a charger for the battery.

For the MCU the Atmel ATmega328P-PU was selected. This chip is the most commonly used microcontroller whether its users realize this or not. It is found on the Arduino Uno hardware. With easy to use interface and pin headers for accessing the MCU pins. This is a candidate to consider. The supported IDE runs a variation of C++ meant for embedded systems. With low cost and adequate power consumption it evenly compares to other MCUs.

Finally, for the radio frequency communication of the nodes the Nordic nRF24L01. Similar to ZigBee, nRF24L01 supports long range mesh network, with low cost and many variations from different vendors makes this choice optimal. Abiding by the IEEE 802.15.4 standards makes this protocol widely available with existing libraries from supporting communities. The ease of integration of this device is very beneficial for what we aim to accomplish with our expertise.

III. SOFTWARE COMPONENTS

The second set of components include an embedded computer used as a server, that hosts a web-application, the database, and all of software needed to start the network. This gateway called a hub has a designed PCB stacked onto it which involves a communication system, and possible solenoid control system for the hose of the Autonomous Irrigation Vehicle. This Gateway is the head link between the nodes so only one per network is needed. Figure 2 represents the hardware and software needed for a Gateway.

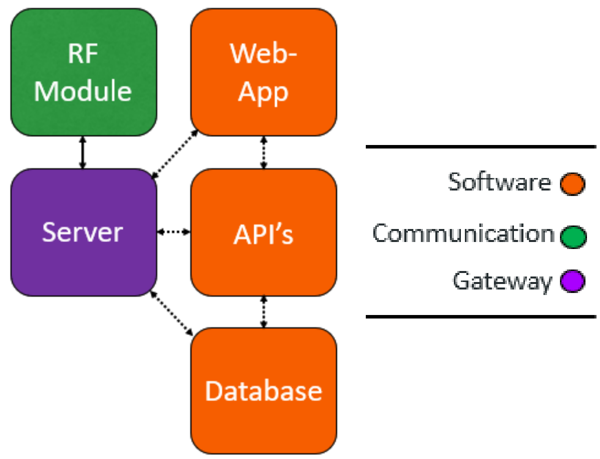


Figure 2 - Gateway Block Diagram

For the gateway server the Raspberry Pi 3 B. At a price of $40.00, Raspberry Pi is the most known single board credit card sized computer today. It supports 512 MB of RAM, USB ports, and a 10/100 Ethernet controller. Raspberry Pi is aimed for hobbyists and supports a resourceful community. With our group having access to one and experience on the Linux OS makes the Raspberry Pi a suitable candidate.

Just as important as the technology used to power the mesh network system is the software integrated inside all of the hardware parts that governs the behavior that the mesh network will take on and how it will respond to user interactions. Various software options exist for the number of operations that the web application will be capable of carrying out. With these various options comes different programming languages, IDE's, and website developers that can be used to produce the most functional software results possible. In order to make our final decisions on the software that will be used in this project, several factors will be considered heavily and equally, such as ease of use by group members, the functionality that it brings into the project, the cost worthiness, and whether the software used will assist our group with potential jobs and schoolwork assignments in the future.

Throughout the undergraduate experience of the members of this group, several programming languages have been used to code different types of functions, games, programs, and applications. The choice made for what programming language will be used in this project largely depends on what languages we can master and implement into our project the most compatibility.

All members of the group working on this project have worked with C in at least a couple classes throughout our undergraduate experiences. C can be used to code the TI MSP430 and other hardware devices that we are interested in using. This programming language has a high probability of being used in this project due to the hardware devices that can be programmed via C code. Working with C to make MOIST functional will prepare the students working on the software side of this project for possible projects requiring the use and knowledge of C in their future careers and endeavors.

The coding for the web application that will be used to monitor and send commands to the different nodes within the mesh network will prove to be the most complicated coding that the students of this group will have to complete. This is due to the lack of experience our group has in web development and in having data be constantly accessed and altered in a database that can have data stored in it via the commands of the web application and changes in moisture level readings of the sensors. The main web development programming languages that we are considering using are JavaScript and PHP incorporated with HTML coding. Both languages offer opportunities to make a presentable web application that can be monitored by all members of this group with some learning and practice put into place. How each web development language will be used will be further discussed in the next section of this report.

Python is considered a more modern programming language than most of the other languages being considered for use in this project. Python would be beneficial to the results of this project in that it possesses the capability to perform powerful functions and data interpretations based on a relatively small amount of code and coding experience. Given the advantages of the MEAN stack, the LAMP stack will still most likely be used to create the web and/or mobile application that governs the mesh network due to the familiarity that this group of students has with web development using the components of the LAMP stack. The operating system used to code the web application for this project will be Linux based, or Windows if too much difficulty is encountered trying to use Linux to code the application. The Apache web server will be connected with and the PHP pages will be coded using the Linux OS. These components will incorporate JavaScript on the front-end to create the user-interface of the web application. The Apache, Linux, and PHP pages are considered the back-end of the application, since they all center around the aspects of a web page that is invisible to the common user. These back-end components will connect the web application to the database, MySQL, whenever a change is authorized or requested by the user through the front-end. This is all displayed in Figure 3 below.

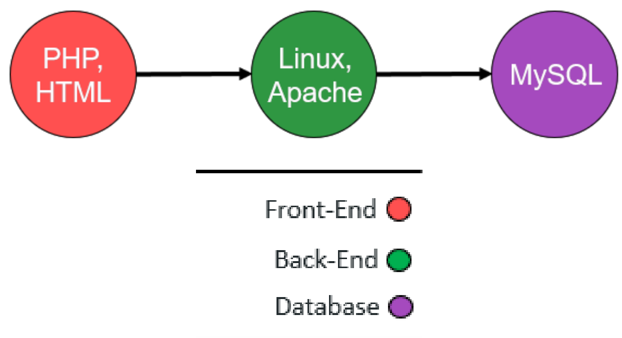


Figure 3 - LAMP Stack Functionality

With all of the strategic components listed, abstract diagrams showing how the components work together and what the flow of operation in this project is helps solidify our design. From the hardware algorithm to the user's front-end web application, these components are all concurrently working together to achieve our objective. Each diagram gives insight into a different aspect of MOIST that, when paired together with each other, will display the inner workings of this project and how it will ultimately function.

IV. Node & Gateway Design

A sensor node is usually equipped with rechargeable batteries, which have limited capacity and pose a challenge to long-term application. These batteries supply power to the sensor nodes by providing the necessary current to maintain each part of the sensor nodes working properly. The total power consumption of a sensor node is the sum of each element in the node (i.e. sensor, microcontroller unit, and RF module) each component may operate at different energy states. Therefore, the lifespan of a sensor node is the time consumed to exhaust its batteries under a sustainable operation threshold. We need to provide sensor nodes with an infinite lifespan.

- Solar panel for charging the batteries; the choice for solar panel
- Charger controller to prevent over charging
- Voltage regulator for regulating a constant 5V DC
- Battery indicator which indicates the remaining voltage left inside the battery
- Cover box to contain all the electronics circuit
- Rechargeable battery that can hold adequate charge

Solar energy has become an alternative source of energy because of its efficiency and affordability, and also because of the arising cost of other sources of energy. The power needed for our project requires independent power source to aliment the sensors and wireless communication module in order to maintain scalable mesh network. The main factors for the choice of power source are cost, efficiency, and portability. The renewable energy such as solar energy which is mainly available will be used as the power source for each node in that project. The polycrystalline solar panel is chosen among the three types of solar panel technology, because it balances cost and efficiency. The size of the solar panel as well as the wattage matter. In this project we selected 2W solar panel manufactured by NUZAMAS. Since we need a solar panel capable of charging a 5 Volt rechargeable battery then the open circuit voltage of the solar panel must be greater than 5 volt which is the reason that we chose a 10 watts solar panel with a Voc (Open Circuit Voltage) of 20.6 volt.

The charge controller will prevent over charging of the battery by automatically reducing the current to a low level when the battery is fully charged. The input is connected to the solar panel and the output is connected to the battery. The state of the switch is determined by two voltage thresholds. When the battery is bulk charging, the load switch is open and the full solar panel output is applied to the battery. When the upper threshold is reached, the load switch closes and the battery discharges through the load until the lower threshold is reached and the switch opens allowing the battery to charge once again. In a series controller design, a relay or solid-state switch either opens the circuit between the array and the battery to discontinuing charging or limits the current in a series-linear manner to hold the battery voltage at a high value. As these on/off charge cycles continue, the 'on' time becoming shorter and shorter as the battery becomes fully charged.in our project we are choosing LTC4079 from linear Technologies.

The voltage regulation (VR) set point is one of the key specifications for charge controllers. The voltage regulation set point is defined as the maximum voltage that the charge controller allows the battery to reach, limiting the overcharge of the battery. Once the controller senses that the battery reaches the voltage regulation set point the input voltage supplied by the battery is suitable for the microcontroller and other peripheral devices of the nodes such as wireless chips and sensors which need a voltage regulator to regulate a constant 3.3 volt to aliment the board.

Solar panels are non-ideal power sources; batteries are often paired with solar panel to provide energy storage. The battery used for this project is the Customized 18650 5 V output with a 2200 mAh. The battery will be paired with the solar panel so that when there is not a proficient amount of energy provided by sunlight to power the Node's functions, then the battery will assist the node with the rest of the power it needs to maintain the node's functionality.

This device has a power consumption of only 26 mW when active and drops to 6 mW when idling. With this low of a power consumption rate, it's a fit for our wireless, solar powered design. The biggest advantage of this chip is the stated familiarity of the device. Arduino provides an IDE that is user friendly to program in and debug. We plan to use existing an Arduino board as a programmer for this chip. This enables us to easily program the chip without adding excess hardware to our project design. And if in the case of a MCU failure, the PDIP socket allows for easy swap-out of chips. ATmega328P-PU is based on the 8-bit AVR family of microcontrollers with a RISC architecture meant for low power consumption and IO peripherals in an embedded environment.

After extensive research our choice for this component came down to E01- ML01IPX. It a surface mount module with nRF24L01 chip embedded, this device is wired to the MCU via SPI, a four-wire interface which out performs the I2C in terms of data speed up to 10 Mbps. This module has a wireless transmission rate of up to 2 Mbps. While active the module consumes 42.9 mW during transmission of data and 37.95 mW during reception of data. And only consumes as low as 3.3 uW during sleep.

The need to setup a coordinate for each node is a need for our database, and for our web-app to accurately map the Nodes with potential location. As an initial install a GPS is only needed when setting or editing a location. Figure 4 is an example of how the whole node and gate way system is set up with the GPS in mind. If the coordinates of the nodes in this position were to be mapped out on the web-app, it would appear as a triangle.

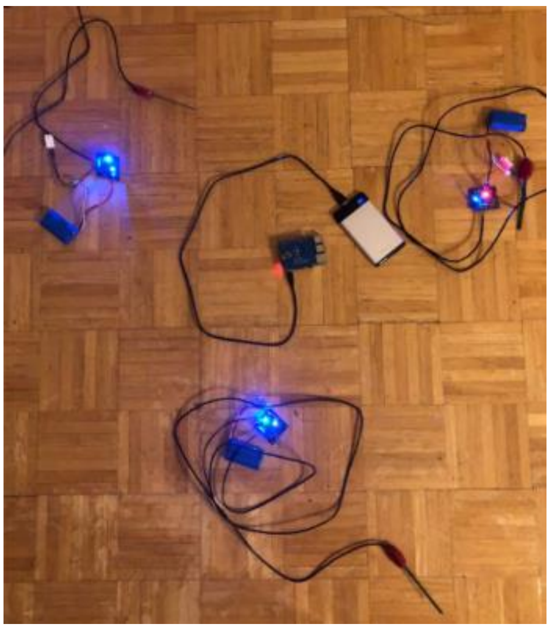


Figure 4 - Node & Gateway Setup

Although personal computers are relatively inexpensive, there are distinct disadvantages to using them in a sensor networks project, especially as sensor nodes. If the sensors are located in areas where mains power is unreliable or unavailable, or where there is a risk of overheating, or where there is simply no room to install a personal computer, we must either transmit the data to another node for processing or store it locally and process it later. However, there is another limitation to using a personal computer as a sensor node: a personal computer has no general input/output (I/O) ports. We can purchase expansion cards for collecting data, but these are often built for use in server or desktop computers. The cost of the computer and the data-collection card, the cost of the sensor node becomes uneconomical. So what do you do in these cases? If only there were a low-cost computer with sufficient processing power and memory, that used standard peripherals, supported programmable I/O ports, and had a small form factor. That's exactly what the Raspberry Pi can do. Figure 5 contains the gateway with a shield attached to it.

Figure 5 - Gateway w/ Shield Attached

After finalizing our test results and calibrating our components, a singular PCB design will be made for actual use. The advantage of creating a PCB is to prevent loose wires and components as well as rid unnecessary material for simplicity, size, and cost. Figure 6 is how the node PCB was envisioned for production release and how the final version of it looks.

Figure 6 - Node PCB

The PCB has three sections of sub circuit, that are grouped based on functionality and common junctions. The power circuit includes a port for a pair of solar panel wires, the NUZAMAS 2W, a port for a pair of battery wires, the LTC4079, and LM1117 surface mount components with their auxiliary capacitors and resistors. The second circuit is the control and communications circuit, this consist of the ATmega328P, surface mount variant, the nRF24l01+ Mini with their auxiliary capacitors and resistor, and a port for programming the microcontroller. This circuit was grouped together based on similarity of the shared SPI communication lines

The second PCB is a shield that stacks on to the Raspberry Pi and offers the RF capability to connect with mesh network. Although much simpler than the Node PCB design, it has constraints of its own. The PCB must fit the dimension of the Raspberry Pi in order for it to properly connect without any obtrusion involved. This creates the Gateway and can then implement the firmware and APIs with the rest of the mesh network. This schematic is designed using Eagle as well.

A board was also created to attach the GPS module, depicted in Figure 7, for our nodes on to each one of them and to then retrieve the current latitude and longitude. A green LED signifies whether the node has loaded the GPS coordinates on to it.

Figure 7 - GPS Board

V. Network Topology

There are many types of potential network setup each with pros and cons. By carefully understanding these networks and how each can be implemented is crucial for our design. Setting up the network is vital in terms of speed and power consumption. If not networked correctly the overuse of a single Node or a fail in data transfer can be a potential issue. Collaborating with our client to figure out the land space and how to set up is necessary. Although many topologies exist, only a few need to be considered for our application, based on number of devices, distance, and frequency of communication.

In the energy-constrained sensor network environments, it is unsuitable in numerous aspects of battery power, processing ability, storage capacity and communication bandwidth, for each node to transmit data to the sink node. This is because in sensor networks with high coverage, the information reported by the neighboring nodes has some degree of redundancy, thus transmitting data separately in each node while consuming bandwidth and energy of the whole sensor network, which shortens lifetime of the network. Data aggregation technology could save energy and improve information accuracy, while sacrificing performance in other areas. On one hand, in the data transfer process, looking for aggregating nodes, data aggregation operations and waiting for the arrival of other data are likely to increase in the average latency of the network. On the other hand, compared to conventional networks, sensor networks have higher data loss rates. Data aggregation could significantly reduce data redundancy but lose more information inadvertently, which reduces the robustness of the sensor network.

The positions of WSN nodes are random, and the nodes can be moved, sheltered and interfered with. The topology of mesh networks has great advantages in flexibility and reliability compared with other network topologies. The self-organizing management approach of network nodes can greatly improve the robustness of the network, resulting in a smart mesh networking technology. In smart mesh ad hoc networking technology, the node first monitors the neighbor nodes and measures the signal strength, and then it selects the appropriate neighbor node for time synchronization and sends a joining request. Then the neighbor node delivers the

request to the gateway. The gateway receives the request and assigns network resources for the node. Based on the mesh network, the sensor network nodes can be assigned with two or more transmission paths to improve the reliability of network. Figure 8 displays an example of the mesh network.

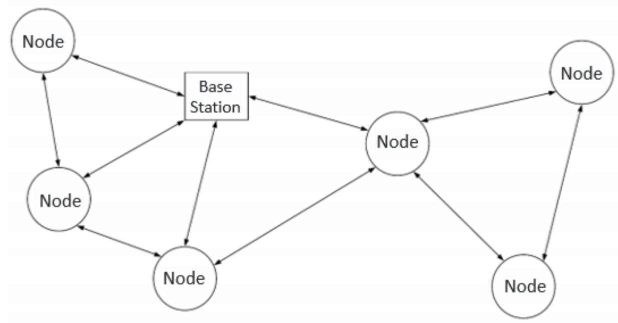


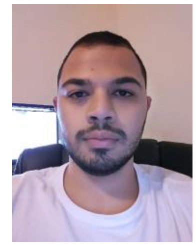
Figure 8 - Mesh Network Topology

V. Conclusion

MOIST has introduced many problems and challenges to the group of students working on this project that were never encountered in other parts of the undergraduate teaching curriculum. In order to persevere through the project and produce a satisfactory result of our diligent research and design, we had to face several issues that a project with such a broad deadline brought upon us, such as time-management, working with a group and with professionals in the engineering field to obtain certain knowledge and materials, and rethinking solutions to the problems we faced that didn't quite work out to our satisfaction the first time around. Through the hardships that this project made us endure, we learned valuable lessons on professionalism that we can take with us for the rest of our lives and put into use immediately upon completing Senior Design and graduating.

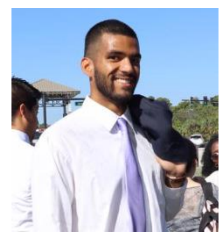
The execution of MOIST flowed smoothly throughout the whole process due to the level-mindedness of all the parties involved in completing tasks for this project. This style of execution could've gone several different ways and other groups of students doing the same project as us could have done it in a completely different fashion based on their experiences with their professor and their own personalities. The methods, products, and purchases to get this project done were completed in the way they were because of the milestones and budgets that our group set and re-established ourselves throughout the course of our research.

Engineers



Ahmed Hamdy is an electrical engineering student with a minor in computer engineering. He enjoys working on his own electronics projects in his spare time. He plans to continue working on MOIST and improve its functionality.

Elange Pierre is an electrical engineering student who also doubles as a logistics manager for UPS. He plans on entering the power industry after graduation.



Gabriel Santos is a computer engineering student that enjoys works on mobile apps and plays sports in his free time. He will be working as a software engineer at the Home Depot headquarters upon graduation.

References

[1] EPA. "When It's Hot." *United States Environmental Protection Agency,* Water Sense , 7 June 2018.

[2] Jawad, Heider. "Energy-Efficient Wireless Sensor Networks for Precision Agriculture." *Sensors.* (2017): *MDPI.* Web. 10 November 2018.