

EEL 4915

Senior Design II

Fall 2013

Eco-Sense

Group Number: 7

Group Members:

Neil Northcutt

Jamie Olheiser   
Daniel Haggerty

Table of Contents

**1.0 Executive Summary1**

**2.0 Project Description2**

2.1 Project Motivation and Goals2

2.2 Objectives2

2.3 Requirements and Specifications3

2.3.1 Central HUB Subsystem4

2.3.2 Wireless Transmission5

2.3.3 User Interface6

2.3.4 Microcontroller7

2.3.5 Power9

2.3.6 Sensors10

**3.0 Research Related to Project Definition13**

3.1 Similar Existing Products and Projects13

3.1.1Autonomous Fixed Irrigation Systems13

3.1.2 PANSY14

3.1.3Autonomous Irrigation Control15

3.2 Lawn Care16

3.3 Weather Prediction18

3.3.1 Pressure18

3.3.2 Humidity20

3.3.3 Temperature21

3.3.4 Combined23

3.4 User Interface24

3.4.1 LCD24

3.4.2 User Input26

3.5 Components27

3.5.1 Sensors27

3.5.1.1 Barometer27

3.5.1.2 Hygrometer29

3.5.1.3 Ground Moisture32

3.5.1.4 Thermometer36

3.5.1.5 Rain Detector38

3.5.2 Microcontroller40

3.5.2.1 I/O41

3.5.2.2 Compiler42

3.5.2.3 Comparisons43

3.5.3 Solenoid43

3.6 Power46

3.6.1 Wall Power46

3.6.2 Batteries47

3.6.3 Solar50

3.6.4 Hydro-Power50

3.6.5 Microcontroller Power51

3.7 Communication and Network Technologies52

3.7.1 Protocols55

3.7.1.1 RF55

3.7.1.2 Wi-Fi55

3.7.1.3 Bluetooth56

3.7.1.4 Xbee/Zigbee56

3.7.2 Zigbee57

3.7.2.1 Star Topology58

3.7.2.2 Tree Topology58

3.7.2.3 Mesh Topology58

3.8 System Housing59

3.8.1 Premade Project Boxes59

3.8.2 Custom Created Housing61

3.9 Optional Additional Features62

3.9.1 Mobile Application62

3.9.2 SD Storage63

**4.0 Project Hardware and Software Overview64**

4.1 Project Block Diagrams64

4.1.1 Hardware Block Diagram64

4.1.2 Software Block Diagram66

4.2 Central HUB Subsystem70

4.3 Sensor Subsystem71

4.4 Solenoid Subsystem72

**5.0 Project Hardware and Software Design73**

5.1 Central HUB Subsystem73

5.2 Sensor Subsystem77

5.3 Solenoid Subsystem82

**6.0 Prototype Construction83**

6.1 System Housing83

6.1.1 Central HUB Subsystem83

6.1.2 Sensor Subsystem85

6.1.3 Solenoid Subsystem86

**7.0 Prototype Testing87**

7.1 Microcontroller87

7.2 Software88

7.3 Sensors89

7.3.1 Thermometer89

7.3.2 Barometer89

7.3.3 Hygrometer90

7.3.4 Ground Moisture90

7.3.5 Rain Detector91

7.4 Solenoid93

7.5 Communications94

7.6 User Interface97

7.7 Full System Test98

7.8 Long Term Test100

**8.0 Administrative Content101**

8.1 Bill of Materials101

8.2 Division of Labor101

8.3 Senior Design II Milestones102

**9.0 Appendices104**

9.1 Appendix A - Copyright Content104

9.2 Appendix B - Bibliography104

**1.0 Executive Summary**

Environmental regulations are becoming much stricter for water consumption and energy efficiency. Eco-Sense is a lightweight and portable home addition to those with existing irrigation systems or for those who would like a new environmental friendly irrigation system. Eco-Sense’s main focus is efficiency with power use and water consumption. It takes advantage of low power mode technology to ensure the device requires minimum power to operate as intended. Eco-Sense comes equipped with autonomous control of water consumption, consisting of a central HUB subsystem (user-friendly device that can be installed inside the home), a solenoid subsystem for water control, and a sensor controller station (to be placed outdoors for monitoring).

The sensor subsystem will come equipped with a barometer, soil hydration sensor, temperature sensor, humidity sensor, and a rain sensor. All the sensors are controlled autonomously, requiring no input from the user. The housing of the sensor subsystem will be waterproof to allow the station to be placed anywhere the user desires. To adhere to the low power standard, the sensor subsystem only becomes active once every period of time. The station will allow reading on varies inputs and relay them back to the central HUB subsystem wirelessly for the user to view.

The solenoid subsystem will contain a solenoid valve to control when water should be distributed. The valve acts as a bridge from an outdoor water spigot to the installed sprinkler system. As with the sensor subsystem, the solenoid subsystem is waterproof to prevent stray water from the outdoor water spigot to damage the circuit.

The central HUB subsystem allows the user to view live feedback throughout the week. The central HUB subsystem also contains a low power mode during idle moments of operation. The user is able to interact with the central HUB subsystem through an LCD display and a set of buttons for overriding actions. The display contains various amounts of information such as previous hydration levels, temperature, humidity level, last time watering took place, and much more. The central HUB subsystem contains an override feature in the event that water should be stopped during a cycle or if the user wants to water more frequently. The central HUB subsystem’s purpose is to offer a simple interface to the Eco-Sense system. Eco-Sense isn’t made to be a strict controller on water consumption, but offer a more environmental friendly approach to efficiency which in return helps save money each month and avoids violating regulatory rules for distribution of water each week.

**2.0 Project Description**

2.1 Project Motivation and Goals

Many neighborhoods have requirements associated with the proper maintenance of lawns especially the aesthetic look of the lawn. If proper watering methods are not adhered to, the lawn will become less appealing. Lawn watering is a hassle that is necessary to maintain a healthy and beautiful lawn.

Hand watering is the process of watering the lawn without the assistance of any watering system. This method is the most time consuming technique for lawn watering provides the lawn with inadequate watering in some locations and overwatering in others. When this is the primary watering method used, it is commonly overlooked because of its time consuming nature. A common solution to hand watering and probably the most used domestic watering technique is the use of a portable sprinkler system. Portable sprinkler systems allow for a more evenly distributed lawn watering and a great decrease in time used for watering. All that’s necessary is the user to turn the system on and off when watering is desired. This often can lead to overwatering due to forgetting to turn off the sprinklers. To solve the issue of remembering to turn the water on and off, typical timer based automated systems have been used. Timer based automated systems have several flaws the most common being the watering of lawn when not necessary, as in when it is raining. It is a common sight to be driving down the street in a rain storm seeing sprinkler systems running. This is not only a waste of water but also can damage lawns with excessive overwatering.

Eco-Sense is a product that is designed to be an alternative to hand watering or traditional automated watering methods. Eco-Sense accomplishes this by using various sensors and algorithms to water the lawn only when it is needed, and not when it is, or going to start raining. By using this system not only will personal lawns receive the water necessary to provide health results, but will also save money by reducing the amount of overwatering by traditional methods.

2.2 Objectives

The Eco-Sense system monitors the weather in an environment area such as a lawn to determine when to water the grass. Each module of the system performs as follows: The sensing parts relay gathered data to the main control. The water valve section turns the water on and off as instructed by the main control hub. And the main control hub makes decisions based on data gathered from the sensor nodes and water the grass if needed.

The communication between the nodes has a large range with low power consumption and the ability to scale the whole system. Data transmitted from one node to another will reach the receiver with all data intact without errors. It is able to transmit data indoors and outdoors going through any obstructions such as walls and buildings. The communication system is be built in such a way that adding or removing sensor nodes would not affect existing sensor nodes. Their functionality is independent of each other. On the event that one node is communicating to another, they will not interfere with other nodes communicating.

The nodes will have a classification system to define their function. The classification scale has simple hierarchy levels. At the bottom are the sensor nodes, which are allowed to go to sleep since they are primarily for transmitting information and collecting data. Next are any nodes such as the water valve control node, which have to be awake for longer period of time listening for a request to turn the water on or off. This level has a pre-set sleep patterns to conserve as much power as possible but able to be on doing nothing and listening for requests. Finally, the highest level is the central HUB subsystem hub. This node is always on listening and transmitting data when needed.

The User Interface (GUI) should is intuitive to use allowing the user to check current readings from the environment, schedule any watering times, and has an override to manually turn on the water valve.

The sensors are able to sense pressure, temperature, humidity, soil moisture, and rain with great accuracy. The temperature sensor determines the air temperature around the sensor node. This senor ranges from 0°F to 120°F. The moisture sensor is placed into the soil of the lawn near the sensor node. It determines the water saturation in the soil to see if it needs watering. The barometric pressure sensor determines the pressure of the air. This is useful in predicting future weather patterns. It uses the data to prevent any lawn watering that is scheduled if Mother Nature decides to water it for free. The rain detector, as its name implies, detects if it's raining. This is used to stop any watering if the system is watering the grass and it starts to rain. This prevents overwatering and wasting water.

2.3 Requirements and Specifications

2.3.1 Central HUB Subsystem

Eco-Sense’s central HUB subsystem consists of a lightweight portable hub that the user can interface with. The central HUB subsystem is simplistic and offers an adequate control over Eco-Sense without having to step foot into the outside environment containing the sensor hub. The central HUB subsystem consists of a human-computer interface design, wireless transmission, data processing, and system control.

The human-computer interface (HCI) is able to be operated with minimal knowledge by the user. The interface contains a display and several pushbuttons for user inputs. The HCI is very reliable and offer a low power state during idle use. It’s made to simplify the interaction between the sensor hub and the user, allowing remote access to all the systems features. The user is able to override the system should the need arise. Inputs override the internal timers of the system and allow the data processing to reoccur at any given time if the user requests so. The system has fast response times with no errors being generated at all times. If malfunctions should occur, error codes are displayed for the user to provide information regarding the problem.

Wireless transmission offer reliable means for communication between the sensor hub and the central HUB subsystem. Frequencies and propagation are chosen to not interfere with current electronics the user may have such as cell phones, Wi-Fi 802.11g, Wi-Fi 802.11n, radio transmission in MHz range, etc. The range for transmission is up to 150 feet without diminishing signals. The central HUB subsystem offers on-site data processing through a low powered microprocessor. The microprocessor offers an adequate frequency but not over exceed the processing requirements to maintain an efficient system with minimal heat dissipation. Data is able to be received, transmitted, and displayed at set intervals. All processing is done by the system with no user interaction.

Power consumption for Eco-Sense’s central HUB subsystem is minimal to reduce cost. Multiple components are powered efficient for this to happen such as wireless transmission, display hardware, and low power modes for a majority of the components. Without having minimal power consumption, Eco-Sense would be efficient in water while creating inefficiency in power use for the user. Eco-Sense is able to operate in an idle mode, active mode, and be able to be turned off completely for a restart.

The central HUB subsystem is mountable on any standard wall without much overhead. The housing for the central HUB subsystem is not waterproof due to being indoors, unlike the sensor subsystem. The mounting is stable and able to withstand various amounts of pressure due to wind gusts and storms in case the unit is mounted within a covered patio or deck. The housing that covers the display potion is not be reflective to avoid a hard to read screen.

2.3.2 Wireless Transmission

Eco-Sense uses wireless transmission to communicate between our central HUB subsystem, solenoid subsystem, and sensor subsystem. The transmission allows us to transmit and receive data without user input and have multiple sensor subsystems if needed. Since data is a critical core to Eco-Sense, the wireless transmission technology is power efficient and reliable. Eco-Sense requires that the transmission module be cheap since there was no sponsor for this project. The transmission module we chose is equipped with digital signal processing capabilities to reduce the workload on a core function. Power consumption is small since this module will most likely be in idle mode most of the time. Transmission will not take place as often as reading sensors, making this a high priority for having a low power mode. The transmissions are generated with a unique identification code to help identify one transmission from another.

Sensor nodes impact this system dramatically, so an easy to use and cheap wireless transmitter was needed. Each sensor node has its own receiver and transmitter, requiring each transmission module to have its own unique code for identification. Sensor nodes use the data transmission at different times to avoid collisions at the receiver on the central HUB subsystem. The data being transmitted is consistent regardless of different identification codes.

Power consumption for the wireless transmission is minimal. This requires the module to have a low power mode during idle use. Each of the transmission modules for the different sensor nodes has the same amount of power and allow them to be live at approximately the same time before replace power must be provided. Wireless transmissions have limited overhead on each of the sensor subsystems, and are able to operate for more than a weeks’ time. The user can forget about the nodes in their outdoor environment and not worry about whether they’re going to fail within the next few days. Low power consumption leads to higher reliability which is a must.

Each wireless transmission module has a microcontroller to do its processing. The transmission module acts as an “end of the line” component, meaning the sensors and microcontroller do all the processing work while the transmission module only relays the results to the respective station. The frequency for the transmission module can vary, allowing Eco-Sense to use a custom channel to run on in the case that the default frequency is being used by another device (most unlikely). The topology for the sensor nodes and transmission modules is efficiency enough so that each node requires the same amount of power to transmit a certain distance. The system topology is the star topology.

2.3.3 User Interface

Eco-Sense is all about efficiency and usability. The user is able to operate and manage the device with little to no trouble. The interface is detailed and accurate for statistics and readability. The user interface (UI) provides a simple human-computer interface that the user can manage with little to no knowledge about the system. The user interface contains an LCD display, has a push button input methods, and has an output to see progress of the system at any given time. The user interface is minimized (display and output) in a low power state.

The display interface is easy to read, manage, and navigate. The display has 16x2 characters and is able to be displayed at any given time. The display module is small to maintain easy portability when attached to the central HUB subsystem. The voltage power requirement are minimum (3.3V to 7V) while offering a low power mode for idle use. The display is able to have several input pins for writing data to the screen and has a moderate refresh speed. The display is able to withstand mild weather conditions such as rain, wind, and dirt.

The interface is not be overcomplicated with inputs and does not give an overwhelming amount of information. Eco-Sense is lightweight and easy to use, requiring minimum knowledge. It’s designed to be an add-on for efficient outdoor watering and care. No programming is required of the user, only simple inputs for actions. All inputs are easy to access while all outputs are easy to read. The interface offers a selection of buttons to navigate the display. There are inputs available for overriding the automation process and allow the user to perform their own actions. Inputs are minimized to make the human-computer interface simplistic. The UI offers accurate and detailed results of the current session. All of the results should be shown through an LCD display and be easy to read. The outputs vary whether the device is in use or in idle mode. Bright lit characters are displayed during operation while a blank screen is shown after an idle time of five minutes.

The UI is smoothly integrated into the central HUB subsystem for the user to offer an enjoyable experience rather than be a chore. The organization of the UI is extremely important for Eco-Sense due to the user should actually want to use this device. The interface is self-sustainable and reliable so no maintenance needs to be done over periods of time for the user to keep using the system. The majority of the UI is incorporated with button inputs, LCD displaying the outputs, circuit and component organization, and a reasonable color scheme to avoid clashing colors. The UI and housing on the central HUB subsystem work together seamlessly. The user will be interacting with this device periodically so the UI is easy to get to and navigate. Buttons are in reach and easy to press, the display is easy to read through the housing, and the electrical side of the device is hidden from view to avoid damage. The UI does not have any unnecessary features due to our power consumption goal. It is basic and does not add overhead to the system such as increase in size, increase in power consumption, or increase in difficulty to use. The UI is the only design that stands between the user and the Eco-Sense system.

2.3.4 Microcontroller

The following requirements and specifications are the guiding constraints that we considered when selecting the microcontroller for every sub-system of the whole wireless network. In the following sections, we decided on the most important parameters for selecting a proper microcontroller that will best fit out needs.

Cost: To reduce costs and time, our group decided to use the same microcontroller on all of our nodes. As this is a self-sponsored project, we cannot afford to spend a large amount of money for each node because there are multiple nodes and the cost would increase greatly. We built a prototype and final design on a custom PCB, so being are able to reuse the same microcontrollers on both, saved us money. Also, the low cost is important due to budget and chance of shorting any pins, which we would then have to buy more. The parts are readily available and unlikely to be discontinued in manufacturing.

Connectivity: Our basic system includes three different nodes, all of which have their own requirements on the needed connectivity. But because we want to reduce cost and keep it simple, we decided to use the same microcontroller on all three nodes. But this would require a microcontroller with all of the connectivity required by the various nodes. Including but not limited to, the sensors, wireless communication service, user interface such as the LCD screen and push buttons, and the solenoid valve. It was also important that the microcontroller have the proper I/O communication protocols such as I2C and serial to communicate to the specific requirements of the sensors, and the serial connection of the transmitter. To ensure that we could connect these components, we aimed for a minimum analog and digital pin count. We needed to have a minimum of 4 analog pins and a minimum of 5 digital pins and able to read data from sensors via IC2 or SPI.

Power: With all of the sensors pulling power from the microcontroller, it is important that the powered required by microcontroller itself is very low. As with any wireless sensor network, it is important that the microcontroller is power efficient and has the power saving schema functionality built in. For our network, the microcontrollers in the sensor nodes are sleeping (very low power) for most of their lives. But occasionally they will need to wake up and doing something, such as read data or transmit data. In order for the microcontroller to wake up, we used timer interrupts. We needed a minimum of two timers that we can use for this. This allowed us to wake it up for different timing schedules. Depending on what the microcontroller offer in terms of power states and what we needed, a minimum number of power states needed was two. One state where it is fully awake, with all of the functionality provided by the microcontroller and other is asleep where only non-core timers are active. Also while awake, it uses less than 50 mW while active or idle, and less than 5 mW while fully powered down. We can calculate an approximation of the battery life using the following equation.

For the central HUB subsystem, the power usage is low enough that the battery would maintain the node for at least a day before recharging. This would be determined using the following formula.

MCC (microcontroller current) would have to be less than 52mA to get 1 day of power with a 1000mAh battery. Because we have a 2 cell lithium polymer battery, the voltage needed does not exceed 7.4 volts.

Functionality and Performance: The microcontroller used also has to perform well enough for the system to make calculations and decisions based on the collected data. The central HUB subsystem is the node requiring the most performance, since it will be making all of the decisions. For this, it was preferable to have as much flash and RAM as possible, but a having 8192 bytes of flash and 128 bytes of RAM is our minimum requirement. Clock speed is not a big requirement, anything around 8 MHz or higher was acceptable. Designing and implementing an embedded system is new for everyone in our group, and adding to the difficulty of designing everything was a bad idea. So to make it simpler, the compiler must be in a familiar language such as C or java. This is so we would spend more time designing the functionality of the microcontroller and not trying to figure out how to program it with a new language. The next aspect in this section would be operating conditions.

The sensor nodes will be out-in-the-elements, it will have to withstand the wrath of Mother Nature. From being in direct sun light, to rain with high humidity, this device needs to last. The main electronic components are sealed in an air tight container to prevent any water from entering it. Being in Florida, the operating temperature range should is at least 0°F to 120°F. But the temperature can still change in the main electronic compartment, which is why that range is important. 120°F is hotter than what is it outside in Florida, but this is not the ambient air temperature; it is the temperature that the sensor node case may get to being in direct sunlight during the summer.

2.3.5 Power

Eco-Sense is a low power system that takes advantage of idle modes to consume minimum power. The system has enough power to operate and avoids having excess amounts of power. Features that needed to be considered when deciding on power is its cost for each unit of power, the lifespan of the power, how reliable it is, not have excess power, and it must be safe to use.

Eco-Sense is a self-funded project; therefore most of our components are within a set price range. Since most power sources are relatively cheap compared to other components, the long term costs were taken into account for the user. This project has acquired power that’s cheap enough but also has its lifespan be average or better. Since Eco-Sense is saving money on water, it should not be relocating those savings into power consumption, which required us to find cheap power.

The lifespan of the power source had to be average or better. Eco-Sense is able to support its sensor hub’s lifespan to over one month to avoid the constant change of downtime. Since Eco-Sense uses low power modes, it is noted that the hubs are able to run for a few months at a time. Because the lifespan of the power source is maximized, it increases the systems reliability. Eco-Sense requires a reliable power source to ensure minimum downtime excluding sleep times. The system has an uptime of at least ninety percent to accurately record data and transmit it to the user.

Eco-Sense has the average amount of power required to operate as intended. To protect the system from too much power, voltage regulators were placed in the system to protect the components and burn the excess power off as heat. Since burning off power is wasted power, Eco-Sense only has the power it’s going to use to minimize wasted energy. Most components require 3.3 to 5V while select components require up to 12V. Because a minimum of 12V is supplied to the system, caution was taken to ensure there are no component failures.

Eco-Sense required that the power source is not only reliable, but safe as well. The power does pose an unsafe hazard to the system or to the user. The power source is able to be sustained in different climates and temperatures along with various weather conditions. The central HUB subsystem power supply is adequately safe due to being indoors but the sensor subsystem’s power supply is developed to withstand nature and also be able to provide power no matter the condition. To ensure the safety of Eco-Sense, the power source is in an enclosed structure and easily accessible incase replacement is required by the user. While every precaution is taken for the safety of the user, Eco-Sense will be a self-sustainable system while power is supplied, therefore requiring the user to only interact with the system during times of failure from power or other critical possibilities by nature.

2.3.6 Sensor Requirements

The thermometer is a very important aspect of the sensor subsystem and has multiple requirements on the functionality of the sensors that was considered. The temperature sensor is required to run on the supply voltage available in the sensor hub, while maintaining a low power usage. The required range of sensing would is able to sense all possible temperatures within the normal temperature range found in the US. The recorded highest temperature in the US is 56.7⁰C and the record low is -62.1⁰C. As -62⁰C is far below the operating temperature of several other components and the accuracy of many of the components falls sharply below 0⁰C , the minimum temperature required to be sensed by the temperature sensor is 0⁰C. To prevent damage and/or less accurate results to skew the data, Eco-Sense notifies the user to remove all outside components from use and location when the temperature falls to or below 5⁰C. For the upper range of temperatures that need to be measured, an increased safety margin of 3% is desired giving an upper bound of 58.5⁰C. Accuracy is another requirement of this sensor. While accuracy is not as critical as other specifications, a better accuracy rating effects which sensor will be used. The accuracy of the sensor has an error of no greater than ±3⁰C. Cost plays an important role in the choice of temperature sensors as the nature of Eco-Sense is of a reduced cost one. The thermometer is placed in a location away from the main circuitry of the sensor central HUB subsystem to insure results are not skewed by heat produced by the other components, because of this, to save money; the temperature sensor is bought and used on a pre-mounted breakout board.

As with all components of the sensor sub-system, the barometer needed to meet the voltage requirements available, 3.3V to 5.0V, while consuming the least amount of power possible in both active and standby mode (shutdown current). Another major requirement is that the sensor chosen has to be able to perform within the range of barometric pressure that is common at various ground levels across the United States. According to weather historian Christopher C. Burt’s 2011 report on barometric pressure records, the lowest recorded barometric pressure in the US was 27.61inHg which converts to 935hPa and the highest recorded was 31.4inHg which converts to 1063hPa. With a margin of 3% on both sides creating greater certainty that the sensor will always be able to accurately sense the barometric pressure. This acceptable range of pressure required for the barometer is 906hPa to 1095hPa. Operating temperature is a requirement that cannot be overlooked. As the range for temperatures needed to be measured is 0⁰C to 58.5⁰C, the operating temperature includes at least this range. Accuracy of the barometric pressure sensor needed to be minimized as much as possible and was preferred to have an error no greater than ±3hPa. The sensor chosen was desired to have as little error as possible; this is a minor requirement as increased error can be compensated through averaging a larger sample of values. Cost was to be reduced as much as possible while maintain the other specifications listed above. Pressure in a sealed container can easily be affected by changes in heat, so to get a proper reading it was required to allow air to move freely, this can expose the other components of the sensor subsystem so the barometer was moved to a location where it can receive the proper air flow needed. With this in mind to save money it was recommended to obtain a barometer on a premade breakout board, as it would save money on the fabrication of a custom PCB.

Humidity sensors, also known as hygrometers, are a common addition to weather monitoring systems, and are important to the goals of Eco-Sense. Because of this the specifications needed had to be followed. The humidity sensor needed to be able to run off the voltage provided by the sensor subsystem, 3.3V or 5.0VDC. Hygrometers are known to be more costly on power consumption so minimizing the power consumed was an important requirement to fulfill. The sensor chosen was desired to have an average current as low as possible to reduce the amount of power used. The range at which the humidity can be detected is an easy specification, as the lowest possible relative humidity is 0% and the max is 100% the requirement does not need to have any extra safety margin because relative humidity ratings above 100% and below 0% do not exist. The operating temperature of the humidity sensor needed to be at least the range of the sensing range of the temperature sensor due to the fact that the sensor subsystem is to be pulled from operation if ever under the temperature of 5⁰C. The hygrometer’s accuracy needs to be high enough to provide accuracy within 5% relative humidity. As with the other sensor specifications, the hygrometer that was selected needed its price reduced as much as possible while still adhering to these other requirements. Hygrometers need to be exposed to the surroundings to get a proper atmosphere, and like the barometer this exposure could damage or destroy other parts of the sensor subsystem and because of this the hygrometer needed to be placed in a separate location where the exposed atmosphere can be obtained for measurement. With this in mind to reduce the cost of mounting this sensor on a new PCB board, it was preferred to purchase a hygrometer that is mounted on a breakout board.

The rain detection sensor is an interesting component because its function is to be exposed to rain water as a part of its function. The sensor chosen has a special specification because of this and that is the ability to be able to completely seal off the circuit from rain water while leaving the necessary parts exposed. Like all the other sensors, the rain detection sensor needed to be able to run off of either the 3.3V or 5.0V provided in the sensor subsystem while providing as little power usage as possible. It was preferred that the rain detector gives an output proportional to the amount of rain detected and not just if there is rain. Another specification is reaction time of the sensor. When a rain storm starts the system needs to know as soon as possible to stop any watering activity. The rain detection sensor also needed to be as cost effective as possible, including the possibility to create one from scratch if necessary.

The soil moisture sensor is a unique sensor because this sensor will be built from scratch; because of this the specifications are not based on products, but methods of creating the moisture sensor. One of the biggest concerns for creating the moisture sensor was the ability to be automated. If the sensor cannot be automated there is no way it can be used in the project. Response time of the moisture sensor was also a big concern. Many techniques can take hours or days to complete the analysis; any moisture sensor that was to be considered for Eco-Sense must have a response time less than 5 minutes time and is preferred to be as fast as possible. Techniques for monitoring ground moisture can be very large and bulky; the selected moisture sensor needed to be small enough as to not create a bulky sensor subsystem station and fit into the sensor substation housing. Some moisture sensors only provide accurate measurements in certain depths of soil; the moisture sensor chosen needed to provide accurate measurements in the level of soil needed. Accuracy is another big factor for the decision of moisture sensors. While accuracy of the measurements is important, because calibration of the sensor will be needed to be performed, the accuracy on the values only needed to show an accurate trend and not any official units of moisture content. Like the other sensor, the ground moisture sensor was required to be able to run off of 3.3V or 5.0V while pulling a low current to keep power consumption down. The ground moisture sensor has to be exposed to the soil to obtain its readings, with this in mind it was required that the moisture sensor had to be able to resist these harsh conditions. The last requirement for the ground moisture sensor was cost. Creating a moisture sensor from scratch helped to keep the cost down, but as the project is self-funded cost needs to be minimized as much as possible.

**3.0 Research Related to Project Definition**

3.1 Existing Similar Projects and Products

Eco-Sense is not be the first of its kind in the field of autonomous irrigation systems. There are various designs for this approach while all the systems are trying to conserve water and use low power architecture to operate the device. A few of the existing projects and products will be discussed to get an insight into current technologies in the area of autonomous irrigation systems.

3.1.1 Autonomous Site-Specific Management of Fixed Irrigation Systems

Autonomous Controller for Site-Specific Management of Fixed Irrigation Systems is a project and design by the University of Nebraska, University of Tennessee, and Embrapa Tropical Agro-Industry. Their project is to offer autonomous control for water consumption in different types of terrain and soil. Their design will include a low power system including solar, low cost, and reduced wiring and piping requirements.

The goals for the project are to develop a system to have multiple stations monitor soil moisture and soil water potential to increase the yield of crops. The stations will be powered by solar energy with battery backups and will offer a specific management in each area depending on the type of soil and environment. A major advantage they want to strive for is to reduce wiring in their system to make it simpler while reducing the amount of piping within the irrigation to reduce the water flow distance.

The equipment the project used for their controller included a microcontroller with a clock, memory, analog interfacing, and battery charging. Other components included in the project was a solar panel, battery, latching solenoid valve, alarm, soil water potential sensor, soil temperature sensor, and a system pressure sensor. The most important component of the controller was their soil water potential sensor. The main features of the sensor was its accuracy, reliability, and low maintenance requirement. The controller was setup to be powered by a solar panel with a battery backup. The controller acted as a bridge between the solenoid valve, pressure sensor, and the field in which they were monitoring the soil. The placement of their soil water potential sensors was a critical choice to their design. They decided to place the sensors right near the roots of the crops to get the most accurate details on the crops current condition. Along with monitoring the moisture content for the crops, the water pressure in the irrigation piping was also monitored by a pressure transducer. This allowed the design team to monitor pressure values within the pipeline while converting the output signal to the microcontroller using analog to digital converters. The software to control the microcontroller was written to measure the soil temperature, the soil water potential, and the pressure of the pipelined system. They decided to store the date and time of each read in the controllers EEPROM. A custom feature they added in was to allow a user based schedule to be programmed into the system to allow watering when the user desired. The user was able to manage and limit the battery charging process also due to varying conditions in weather for the solar panel.

To test their design, the team used four controllers and simulated different types of environments by placing their soil and crops into containers. The experiment took place at the University of Tennessee in the summer of 2002, so adequate solar power was available during the day. The main statistics they were testing for included reasonable soil water potential measurements, temperature variations, and uptime of the system during a one month period. The system proved to be a success in producing a higher yield in crops for the user. Water was used the most during the period of noon and 8:00pm on each day due to the heat and direct sunlight exposure on the soil. The costs for each unit if it were to be produced amounted to about $0.21 in components in 2003. The design proved to be efficient, cheap, and reliable during the one month period.

3.1.2 PANSY: a Portable Autonomous Irrigation System

The PANSY project was started to make greenhouse techniques more efficient. The project was developed by Benjamin Beckmann and Ajay Gupta from the Department of Computer Science, Western Michigan University. It incorporates a watering efficiency state and also a fertilizer efficiency state. The project was made to monitor each plant individually by proving an autonomous controlled irrigation system within the greenhouse. This control would reduce human interaction and follow patterns in treatment for the crops to make the growing process more efficient.

The goals of the project were to offer an autonomous solution to environment control for crops within a greenhouse. The project would be equipped with sensors such as a temperature sensor, humidity sensor, and a soil moisture sensor. The design team decided to make each sensor be configurable to each crop to help increase the usefulness of the project. The system was made to increase efficiency in two main areas: crop yield and human interaction times. If the system would automate what the user does on a typical day, the user could spend their time elsewhere and leave the PANSY system to take care of the irrigation within a greenhouse.

The PANSY system was equipped with a temperature sensor to monitor the greenhouse temperature, a humidity sensor, and a soil moisture sensor for each designated spot for the crops. For the project, the team decided to use a greenhouse that was divided into sections. This allowed the option to see if one section was surpassing another section according to the data being recorded. The sections were to be connected via wireless communication with each node interlinked with the main controller. The nodes are able to reconfigure themselves to respond to changing environments within the greenhouse. For the nodes that were too far away, they used a repeater node to bridge the controller and sensor nodes. The moisture sensor that is used on the sensor nodes measures the dielectric constant and can be converted to a water constant for use in their software algorithms on the controller.

The project was setup in the late summer of 2004. During this time, the greenhouse would typically be very humid and allowed a vast amount of data to be read during the month. The team decided to section off a portion of the greenhouse for a grower to periodically check his crops and water as he sees fit, and have the PANSY system monitor another section to see how efficient the project was. The controller was hooked up to a display, also hooked up to the switching circuit and solenoid valve. They used one hundred stock pots for testing the PANSY system. Since the timing of the experiment was during the late summer months, the team decided to test their design on pansies due to availability. The data that would be collected for this design would be reliability, amount of maintenance needed, the amount of water used, the amount of fertilizer used, the foliage of the plants, and the root structure of the plants.

The experiment and design concluded to about one third less water used for the plants that were controlled by PANSY. The plants also needed less fertilizer during the testing phase in comparison to the growers’ attempt of his crops. By using less water and fertilizer, the cost of the maintenance was also reduced. The root structured from selected plants in each section was reviewed and found that the PANSY plants had a more uniform structure while the growers’ plants had a sparse unorganized structure. The end result for this design was that PANSY was capable of keeping up with human judgment in determining the watering amounts and times. High hopes were given to this project since the algorithms would be further improved and make the irrigation system much more efficient.

3.1.3 Autonomous Irrigation Control

Autonomous Irrigation Control is a product developed by Acromag Inc. The purpose of this product is to offer irrigation systems that run autonomously in larger fields that produce mass crops. The sprinklers that are used in larger crop fields are industrial size, meaning less sprinklers will be needed in the field compared to traditional little circuit nodes. The sprinklers will be controlled from a central server that can take advantage of local energy sources such as solar, water, geothermal, wind, and a backup battery.

The motivation and goals for this product are to offer efficiency and accurate amounts of water to the field for farming. The larger sprinklers can cover much more area than a traditional sprinkler for home use, meaning larger amounts of land can be watered simultaneously. The product is designed to control field water consumption and delivery to a minimum, to reduce human interaction with the field system, and provide alerts when the irrigation system requires maintenance.

The product comes equipped with several irrigation components such as a moisture sensor, pressure sensor, and a flow meter, flow control valve, shut off valve, drip tubes, and a solar panel. The system will be connected to the main water supply of the specified land and over cellular network capability to monitor the status of the product. The valves and sensors are connected to an Ethernet bus on a TCP network that communicates to the autonomous controller. The controller is able to determine whether the field needs to be watered depending on the inputs it’s provided through the sensors and valves. Solar power is used to power the shut off valve, flow control valve, flow meter, moisture sensors, and the server.

Various amounts of information can be collected off this product during operation. The Autonomous controller is able to relay information such as the current status of the field and whether it’s in service or not at any point in time, the power available and how much is being consumed, and the battery status to determine whether the batteries are in use. The main water supply can relay information such as water pressure of the system, flow rate of water, and the shut off valve status. The irrigation pipes can relay information such as the emergency shut off valve status, moisture sensors, flow rate, flow valve position, the power being used, and the condition of the batteries. The product still needs a minor amount of supervision and maintenance (usually the customer) for human intervention and status reports for the system. Key features for the product describes is its ability to operate from -30 to 75 ◦C, low current draw, Ethernet support, and choice of operating system.

3.2 Lawn Care

The Eco-Sense projects main focus is on the proper hydration of home laws while reduce the waste of excess water used by traditional automated watering systems. To accomplish this several factors have to be considered. These factors are the effects of overwatering, the differences between shallow watering and deep watering, soil field capacity, and conditions for proper watering.

Overwatering is a big issue with the maintenance of a healthy lawn. Contrary to popular belief, watering lawns every day or every other day does not promote healthy growth but in fact has the opposite effect. The first effect of over watering is shallower root systems. This is caused by many factors most prevalent factor being that there is no reason for the plant roots to go deeper if water is always present. This problem doesn’t seem bad at first, but when the constant water source stops for any reason, the grass can easily wilt and die. Other problems can arise due to shallow root systems including increased susceptibility to disease and damage by insects and lacking the ability to recover fully from these conditions. Overwatering also leads to increases in yard weeds and mushrooms. Weeds and mushrooms will fight over the resources available causing grass to die off. A well water lawn will not die as easily in drought seasons as an overwatered lawn while being more resistant to other issues like invasive plants and insects. While not a direct effect to the well-being of the lawn, overwater also has the effect of wasting large amounts of money.

Shallow watering is another problem similar to overwater. While shallow water is a different issue it can still be a form of overwatering and has similar effects. Shallow water is the process of water lawns only enough to get the top couple inches moist with water. This leads to shallow root systems creating the same issues as overwater. This method of shallow watering is very attractive because it seems like water is being conserved by only watering a little at a time. This shallow watering method is often performed on a daily basis leading to overwatering even if the water applied is less. Deep watering is considered ideal for watering lawns. Health lawns need a deep root system to insure the survival of the grass, herbs, and shrubberies. Deep watering is defined as watering to where about 12 inches (for lawns) of soil are moist. Use deep watering methods, however, can be very damaging to the soil, lawn, and environment if done constantly with high frequency. If used daily it will result in shallow root systems, more disease and insect prone lawn, waist of water and money, and the erosion of the soil supporting the lawn. It is important to water the lawn deeply but to avoid overwatering.

How long to wait until re-watering is a difficult decision to make. As there are many factors that affect the decrease in soil moisture over time it is hard to give an exact time. Some of the factors that affect loss of soil moisture are plant type, soil type, and weather conditions. Different plants absorb different amounts of water. Even plants of a similar species absorb water differently from each other. There are many types of soil types including dirt, sandy, and clay soil types. Sandy soils absorb water quicker and drain at a faster rate; this would require more frequent shorter watering sessions. Clay soil types absorb water very slowly but can retain it for a long period of time. Water applied to fast to clay soil types will erode easier. A soils field capacity is a way to quantify how much water a particular soil can retain. Like a sponge pulled from a bucket of water, the soil when fully saturated will drain water due to gravity until the point that it can hold onto all the water present. This point is the soil field capacity. This point is the ideal amount of water for any given soil. When the soil gets bellow a given point it is recommended to water to the field capacity level. This point is usually about 75% of the field capacity point, if a field can hold 100 gallons of water when it reaches 75 gallons it is time to re-water. Like the air conditioner in a home, this method will keep the water content in the soil relatively constant.

Not all conditions for watering the lawns are created equal. There are many factors that affect the proper times to water most prevalent being the evaporation rate of water. Watering the lawn serves no purpose other than wasting money if the water evaporates before it can be of use. The evaporation rate of water used for watering the lawn is contributed by three main factors heat, humidity, and wind. It is well known that water evaporates when heated, so it is not a big surprise to learn that the higher the temperature is the more water evaporates. Relative humidity plays an important role in determining watering conditions. Relative humidity is the percentage of water present in the surrounding atmosphere. When the relative humidity is low the air has a higher capacity to absorb moisture. A higher relative humidity has less room for moisture as it is already in high quantities in the air. Even with high temperatures if the moisture has nowhere to go, the evaporation will be less. Wind may play a role in evaporation especially if the wind is a dry wind. Dry winds have a low relative humidity and therefore have a large capacity for absorbing moisture. With these conditions in mind the most suitable times to water would be at lower temperatures, and higher humidity levels. Night time is a great time to water as the temperature is at its lowest allowing for a longer time to saturate before evaporation starts.

3.3 Weather Prediction

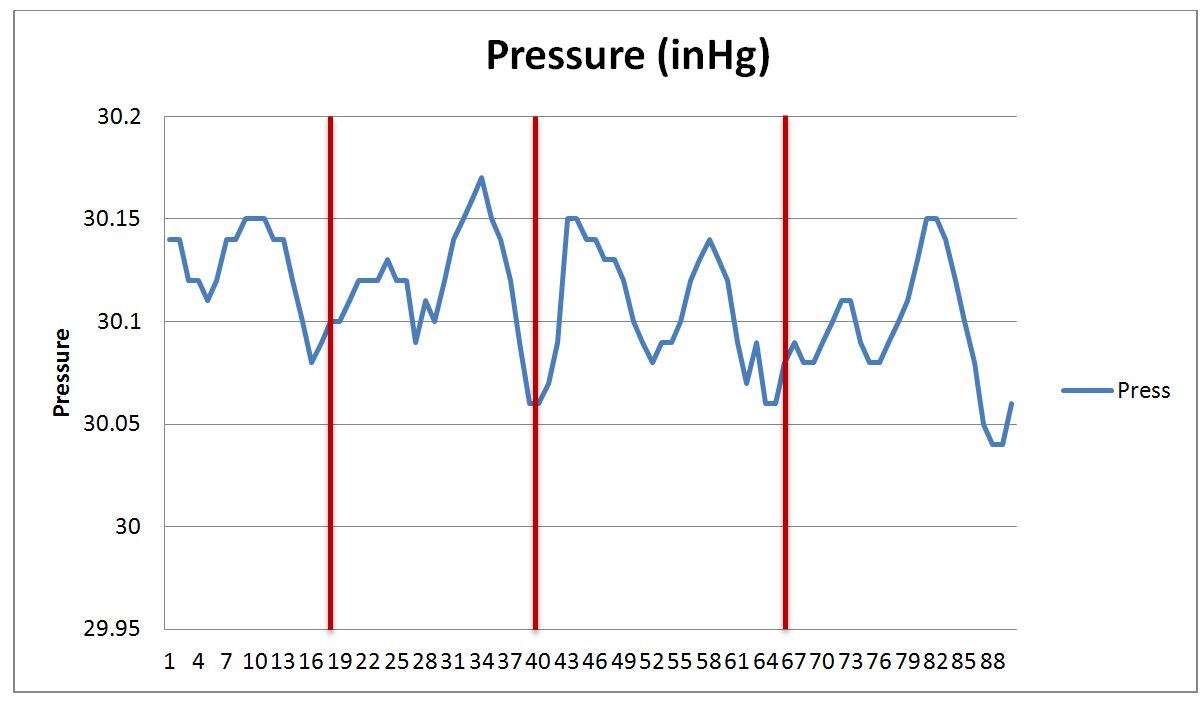
Eco-Sense is designed to be an efficient and cost effective automated watering management system. To be as efficient and cost free as possible Eco-Sense allows the elements of nature, namely rain storms, to take as much of the watering duty as possible, this allows for the greatest reduction in water wastage and saves more money than watering on a timed schedule regardless of weather conditions. To do this Eco-Sense needed to be able to not only know there is inclement weather, but be able to predict its onset. Eco-Sense also needed to determine there is nice weather, or will be soon. If stormy weather is on the way and the lawn needs watering, Eco-Sense will not water due to the prediction of rain storms on their way. Weather forecasting is an interesting and difficult science that has to look into the interactions of many physical processes at the same time, it is not a science that can be minimized into an automated system however, there are methods to predict a general forecast several hours out and this is perfect for the needs of Eco-Sense. For this weather prediction each of the main predictors for weather prediction that are used by Eco-Sense, and these are pressure, humidity, temperature, and there effects when combined with each other. With these weather indicators and a general understanding of meteorology it was possible to create an algorithm used by Eco-Sense to predict the onset of stormy weather conditions to minimize watering in or before rain.

3.3.1 Pressure

Weather forecasting has been done since the earliest of times, looking up at the sky and attempting to predict weather based on cloud color, movement, or other phenomena. It wasn’t until 1644 that Italian physicist and mathematician Evangelista Torricelli created the first mercury barometer. The barometer measures atmospheric pressure which has become known also as barometric pressure. Barometric pressure measures the weight of air being pulled down by gravity. Soon after the invention of the barometer, it was found that barometric pressure naturally fluctuates in air pressure while the barometer is kept at the same elevation. It is important that the elevation remains constant because pressure changes with elevation, as the elevation raises pressure decreases and as elevation lowers the pressure decreases. If the barometer is not in a fixed location the measurements can easily be misleading. Barometric pressure has to main subjects to be discussed to understand its application to weather prediction; those subjects are the reasons for pressure change, and the effects of pressure change on weather.

Barometric pressure fluctuates constantly throughout the day and there are many reasons for this to happen, these are air rising, change of air density, and the change of the mass of air. Air rising counteracts some of the downward force of gravity on the atmosphere lowering the pressure. Pressure troughs are defined as an elongated region of relatively low pressure. They are commonly associated with weather fronts with cold fronts having a more defined pressure trough then warm fronts. These pressure troughs of fronts force the air to quickly rise lowering the barometric pressure. The density of air is related to the amount of moisture present in the atmosphere. As more moisture air is less dense then dry air causing the air to rise and lowering the barometric pressure. Moisture advection is the process of less dense, moist air is moving into an area decreasing barometric pressure. Warm air advection is a similar phenomenon of warm dry air moving into a region causing a rise in barometric pressure. Changes in the mass of air can also cause pressure changes. The most common causes of mass changes are a reduction in mass caused by upper level divergences. These upper level divergences take place near the top of the troposphere, the level of earth atmosphere that is livable, and is caused when a mass of air is pulled away from a region faster than that mass can be replaced creating a reduction in the mass. These upper level divergences are caused by fast winds in the upper troposphere by weather events such as vortices and jet streams.

Barometric pressure does not affect weather at all; it is, in fact, the weather that affects the barometric pressure. There is a very clear correlation between pressure and weather. In general pressure changes indicate different types of weather, not just lower or higher pressure. Figure 3.3.1.1 shows data taken from a barometric pressure sensor for a 90 hour period. The red marks located on the graph are times when rain storms started. As the graph shows, when weather that causes rain storms begin to move in, the barometric pressure declines.



Pressure measurements for 90 hours  
Figure 3.3.1.1

This decline in pressure is present before any visual sign are present and can happen hours before the onset of a storm. Decent, rain free weather can also be determined by the trend of barometric pressure. As barometric pressure raises the chance for rain storms drop considerably and it can be inferred that decent weather was around at that time. Constant low pressures with no rain storms can mean that rain storms can in sue if pressure drops a little more, while a constant high pressure means that pressure has to drop considerably more before rain occurs.

3.3.2 Humidity

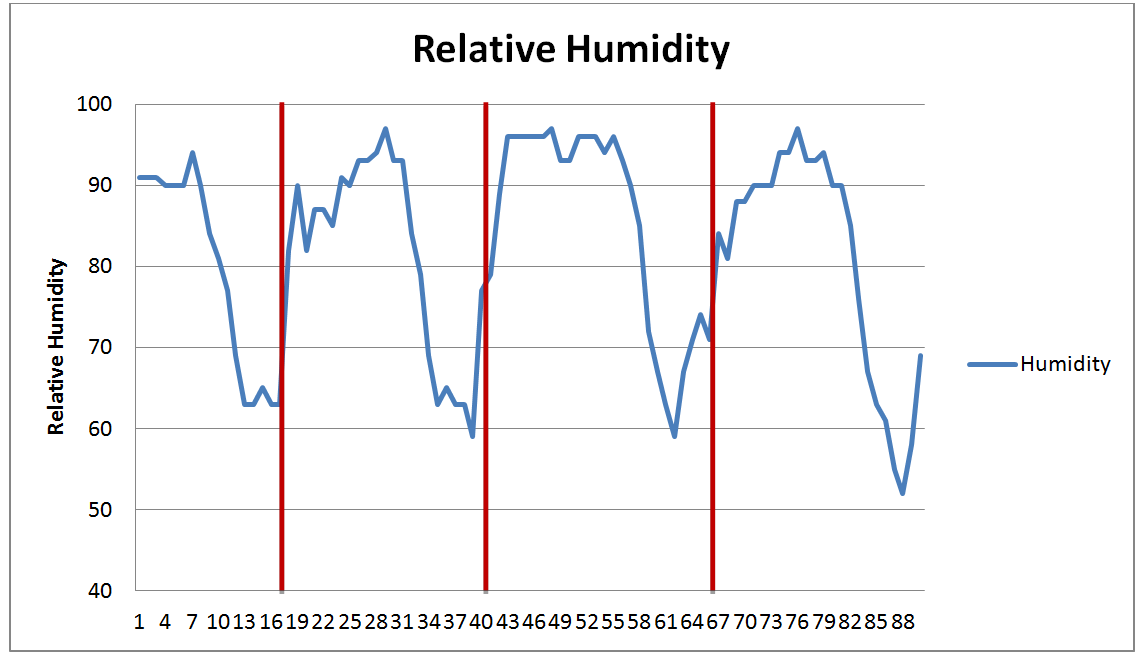
Humidity is the amount vaporous water present in the lower troposphere of the earth’s atmosphere. When water is in a vapor from molecules like to be as far apart as possible because of this the water vapor is invisible and suspended evenly through the atmosphere. Relative humidity is the ratio of the amount of water present compared to the capacity of moisture the air can hold at the moment. If the relative humidity is at 50% that means the surrounding atmosphere is holding 50% the moisture that it can hold. At 100% humidity the atmosphere can no longer hold any more moisture then what is present. If a humidity sensor was to be put into a glass of water the humidity would read 100%. The capacity at which the air can absorb moisture is directly related to the ambient temperature of the air, the formula to obtain the relative humidity is displayed bellow.

Where E is equal to the dew point temperature in Celsius, and as is the saturation vapor pressure given with the Clausius-Clapeyron equation bellow.

Where L is the latent heat of vaporization which equals 2.453x10^6 J/Kg, Rv is the gas constant for moist air which is equal to 461 J/Kg, and T is the temperature in Celsius.

There are several reasons in which humidity can change, knowing these reasons are essential in understanding how to use humidity to predict weather. These common ways of humidity increases are rising air, decreasing temperature, and moisture advection. Rising air is one way humidity changes. As air rises through the atmosphere, it cools down causing the dew point depression to lower and the relative humidity increases. The dew point depression is the difference between the temperature and the dew point temperature at a certain height in the atmosphere, the smaller the difference, the more moisture is present resulting in a higher relative humidity. As explained with rising air pressure, the decrease in temperature, even without a rise in the air, will cause an increase in relative humidity. As discussed in section 3.2.1 moisture advection is the process of winds being blown from locations of large bodies of water carrying along additional moisture raising the relative humidity at the location.

Relative humidity plays an important role in the prediction of weather. Figure 3.3.2.1 shows a line chart of relative humidity values over a 90 hour period. The red lines are approximations to when the moisture in the atmosphere began to condense and create rain. In general precipitation occurs when the relative humidity increases, thus increasing the amount of moisture present in the atmosphere. When the relative humidity lowers it is a signal that less moisture is present in the air and therefore the chance to rain is decreased. As it can be seen on figure 3.3.2.1 the relative humidity always rises sharply before measured precipitation. These rises are seen to be



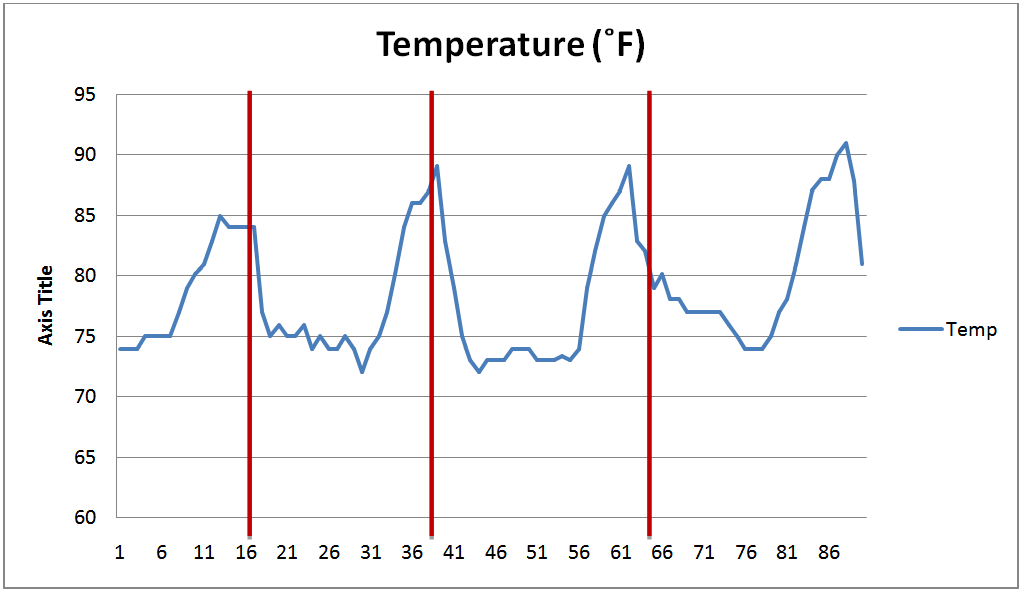
Pressure measurements for 90 hours   
Figure 3.3.2.1

an indicator for precipitation but tend to be a very short term prediction method with the data shown. Even though it is displayed in the figure, it is not necessary for the humidity to drop sharply before the sharp rise and is only shown in this figure due to otherfactors affecting the relative humidity. It is also important to note that precipitation does not need a specific level of humidity before rain occurs nor does it always rain at 100% humidity. There are occasions where 100% relative humidity exists and there is no precipitation, these occasions include weather conditions like fog.

3.3.3 Temperature

The degree or intensity of heat present in a substance or object is the last of the big three conditions which Eco-Sense will use to predict the onset of precipitation, Temperature. Out of the three major phenomena Eco-Sense will use to determine weather conditions, temperature is the one most easily detectable and understood, it is because of this that temperature was the first of these to have a measuring device created. The thermometer was created several times before by several investors over the year, but the first thermometer that could be used to measure variations in temperature is accredited to Galileo Galilee in 1593. Since then the causes of temperature changes and the effects that temperatures have on weather have been studied.

There exists variety of different reasons that would cause temperature to change. While some are complex in nature, some are easier to understand. The first of these factors to the cause of changing temperatures is probably the biggest and most known factor, sunlight. Sunlight has a direct impact of the temperature at a given location. With the rotation of the earth, sunlight’s presence gives way to rising temperatures. While the presence of sunlight is enough to cause a temperature the intensity of the sunlight is also a contributing condition. The next major contributing factor is the presence of a very thick and gassy atmosphere. This gassy atmosphere can amplify the temperature and is known as the greenhouse effect. The last major cause of temperature change is wind. There are many different ways that winds can affect temperature with the first being horizontal winds bringing in air with different temperature. These winds can either be unpredictable or they could be what are known as prevailing winds. Prevailing winds are winds that blow commonly in one direction on a given location on the earth. As winds carry temperatures from different regions, colder or warmer temperature winds can affect the temperature at a given region. The second way air movement can affect temperature is with vertical winds. As air travels upward clouds begin to form causing a shading effect from the sun, lower the temperature due to direct sunlight.

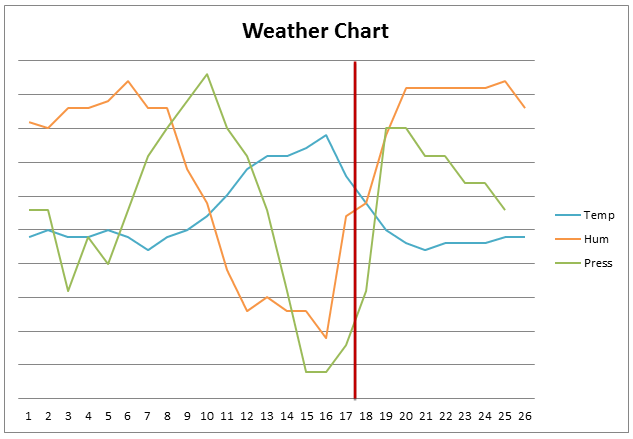


Temperature Measurements for 90 hours  
Figure 3.3.3.1

As the temperature changes there are distinct reactions that affect weather conditions, the first of these is the evaporation rate of ground water. As temperature increases the evaporation rate of water also increases. As more moisture moves into the air, there is a higher chance for precipitation due to the increase in moisture present. Evaporation is not the only requirement for rain, when temperature lowers the moisture present in the air condenses. When water molecules in the air condense there is a point that the water can no longer be suspended in the atmosphere and causes precipitation. Figure 3.3.3.1 shows temperature data taken over a 90 hour period where the red marks indicate when precipitation commenced. In this figure you can easily see the temperature cycle due to the sun rising and falling. One thing that is common in every indication of rain is that the rain is always preceded by a general rise in temperature followed by a sharper fall in temperature. With this it can be shown that precipitation is preceded by a relative decrease in temperature.

3.3.4 Combined

The general interactions of pressure, humidity, and temperature with precipitation have been previously discussed, in sections 3.3.1 through 3.3.3, to be able to predict weather with some degree of accuracy. As these factors by themselves are good way for weather prediction, when used together the accuracy of weather prediction increases.



Pressure, Temperature, and Humidity over 25 hours  
Figure 3.3.4.1

Relative humidity, the proportional amount of water temperature found in the air at a given point, is directly related to temperature. As the temperature increases, the amount of moisture that can be held in the air increases. With this increase the relative humidity decreases when heat rises. With the relative humidity lower because of a temperature increase, the air has more room for moisture and with the higher temperature causes more water to evaporate increasing the relative temperature and repeating the process until another factor is taken into effect. With a barometer, barometric pressure can be used to measure the amount of air is traveling vertically because barometric pressure is less with more upward motion. The lower the barometric pressure the faster air moves upward. As air travels upward it is cooled off, this cooling will cause any moisture to begin to condense. If the relative humidity was high enough, there would be enough moisture to condense into rain. Hot, moist air is less dense then colder, dryer air causing the barometric pressure to drop.

It is now clear how to fully predict the onset of precipitation with barometric pressure, relative humidity, and temperature. Figure 3.3.4.1 shows a 25 hour sampling of humidity, pressure, and temperature taken from the same data as figures 3.3.1-1, 3.3.2-1, and 3.3.3.1. The ride line marks the start of precipitation. This figure confirms that a barometric pressure drop is the first signal that a rain storm maybe on its way. If the temperature decrease and or the relative humidity increases while the barometric pressure decreases, the possibility that precipitation is imminent is very high.

Figure 3.3.4.2 bellow shows the created weather prediction algorithm used in Eco-sense. This algorithm has a 70% successful prediction of the onset of weather, 1 hour away, and a 15% false positive rating. When the value of ‘rain’ is over 0.4, weather is predicted in the next hour.



Figure 3.3.4.2  
 Rain prediction algorithm

3.4 User Interface

3.4.1 LCD Display

Seven Segments versus Dot Matrix: The display for Eco-Sense needed to be able to portray at least 16 characters in a single line using minimum spacing. Seven segment displays offer easy readability and can be multiplexed together to create the desired length needed. Segmented displays require 2 bytes of data per character. If it were to be used to display Eco-Senses statistics, it would require 16 x 2 bytes per line to be stored in EEPROM as overhead data. Segmented displays require low power to operate and seem like a good candidate for the project.

Dot matrix displays require 1 bit of data to be stored per pixel, requiring a maximum of 1 x 128 x 16 bits of data to be stored in EEPROM as overhead data for Eco-Sense. While the amount of memory needed increases, dot matrix displays can also emulate segmented displays. Compared to segmented displays, dot matrix displays offer a sharper output to the user. They can also be operated in a low power mode, offering backlight solutions for different environments depending on where the central HUB subsystem is placed.

After reviewing the comparison of segment and dot matrix displays, it was decided to use a dot-matrix approach. The dot matrix display uses slightly more memory but the availability and overall display are more beneficial to Eco-Sense.

Passive Matrix versus Active Matrix: Passive Matrix LCD’s offer the lowest power consumption and offer the lowest cost. On the downside, they are nearly impossible to read in dark environments without the use of a backlight and offer no customizability to the user. Eco-Sense’s central HUB subsystem will be placed indoors, making the LCD require a backlighting solution. The display is organized by rows and columns through multiplexing. As more pixels need to be refreshed, response times tend to decrease for passive matrix displays. Since Eco-Sense only needed a 16x2 display, this problem becomes less important.

Active Matrix LCD’s use slightly more power due to being a thin film transistor based display. Refresh rates are much faster and accurate but seem to be too good for the desired use of the project. Addressing is done through each pixel separately and requires a little more overhead for managing different states of the pixels.

After reviewing the comparison of passive matrix and active matrix displays, it was decided to use a passive matrix LCD. With Eco-Sense being an efficient device, minimum power consumption is a must. While the display won’t be as sharp as an active matrix display, passive matrix will offer enough resolution for the intended use of displaying statistics from Eco-Sense.

LCD Display Comparison: Table 3.4.1 below compares a few of the LCD displays being considered and the features they provide. These LCD’s are dot matrix formatted and will use passive matrix addressing unless a cheap active matrix addressing LCD is available. A common driver for dot matrix displays is the Hitachi HD44780 driver.

|  |  |  |  |
| --- | --- | --- | --- |
| **LCD Module** | **GDM1602K** | **LCM-S01602DSR/A** | **LCM-S01602DTR/M** |
| **Character Display** | 16 x 2 | 16 x 2 | 16 x 2 |
| **Character Size** | 5 x 8 dots | 5 x 8 dots | 5 x 8 dots |
| **Dimensions** | 3.15” x 1.425” | 3.15” x 1.421” | 2.60” x 1.02” |
| **Bit Mode** | 4 or 8 | 4 or 8 | 4 or 8 |
| **Backlight** | LED | None | None |
| **Voltage Supply** | 0 to 7 V | 0 to 5V | 0 to 5V |
| **Low Power Mode** | Yes | Yes | No |
| **Interface Pins** | 16 | 16 | 16 |
| **Operation Temperature** | -20 to +60 ◦C | -20 to +70 ◦C | 0 to +50 ◦C |

Table 3.4.1: LCD module comparison

The GDM1602K LCD module offers an LED backlight while the other two displays do not. The operating temperature is average between the other two displays but Eco-Sense will be well within the operation temperature making this a less important decision feature. Since this LCD module offers a backlight, it has a slightly higher voltage requirement if the backlight is used. The nice option about the backlight is its option to toggle on to offer increased vision in dark areas or toggle off to enter a lower power mode.

The LCM-S01602DSR/A LCD module offers very similar features as the GDM1602K but the LCM does not offer a backlight. The operating temperature is higher than the GDM but as previously discussed, this was not a determining feature for Eco-Sense. The attractive feature for this display is the 0 to 5V limit versus the 0 to 7V limit on the GDM. This display will always run within that voltage range since there is no backlight.

The LCM-S01602DTR/M LCD module belongs to same family as the LCD-S01602DSR/A module but seems to be limited in its features. The device offers a much stricter range for its operating temperature, being the minimal range of all three displays. This display offers a smaller dimensional area compared to the other two displays while the LCD screen itself remains the same size. The nice feature for this is the LCD module would have less overhead on the overall central HUB subsystem, but compared to its disadvantages, this display would most likely not succeed into our decision process.

After comparing the three choices for the LCD module, it was decided to go with the GDM1602K LCD module. This module, compared to the others, had the best of both with features and offered a lot of options for customizability for the user. The backlight mode and low power mode were very attractive in this display since this module will be mounted indoors. The backlight will offer easy readability no matter the time of day and offer low power consumption when not in use.

3.4.2 User Input

Eco-Sense is an autonomous system for sensor data, transmission, algorithm computation, and feedback. User input will be available at the central HUB subsystem for the user to override the watering routine and cycle through the display if needed. Common information that the user can cycle through will be cycling times, sensor data, and various timers. The user input is a part of the user interface, the part of the system in which the user is freely able to interact with Eco-Sense. An option for user input will be touchscreen or button inputs.

Touchscreen input devices are nice to look at and use for a wall mounted device. They typically have more functionality than standard buttons and can be easily adapted to any configuration the user wants. A major downside to touchscreen devices is their high cost and high power consumption. A touchscreen input device is usually made from a resistive coating on the upper layer and a separate resistive coating on the bottom glass. When the two coating come in contact to create a point, the circuit completes and a signal is sent. The problem with this type of display for Eco-Sense is the user won’t be inputting anything for the system to technically read. The inputs will be used for cycling through screens on an LCD just to view the information in which Eco-Sense has collected. For overriding the system, the user will have an available button that will allow them to being a water cycle in the next n-minutes. After viewing the possible inputs from the users, touchscreen just doesn’t seem to fit the criteria that Eco-Sense was looking for.

Pushbuttons are small fingertip sized buttons that the user can press to send input into the system. The buttons can be programmed to do what the user wants them to but must be preset before compile time. The buttons are low powered devices and are very cheap to implement. Eco-Sense can take advantage of the low power since it’s a strong focus for this project. The pushbuttons only have two states: pressed and released. This saves some power due to the fact that unlike touchscreen, the controller will only have to monitor for two states and not two states plus vector positioning. A couple buttons will be compared to choose the best fit for Eco-Sense in table 3.4.2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Button** | **Mini Push Button** | **Button Pad 2x2** | **LED Tactile Button** |
| **Number of Terminals** | 4 | 4 | 4 |
| **Number of Buttons** | 1 | 4 | 1 |
| **Dimensions** | 6mm x 6mm | 47.5mm x 47.5mm | 12mm x 12mm |
| **LED Compatible** | No | Yes | Yes |
| **Shape** | Square | Square | Square |
| **Cost** | Very low | Average | Average |

Table 3.4.2: Button comparison chart

After reviewing the button possibilities, it was decided that Eco-Sense would benefit more from the mini push button due to the no power consumption on the LED compatible feature and also its smaller size. The button is large enough for a user to press comfortably and not too large to fit multiple buttons near each another for an easy access menu. The very low cost compared to other buttons is attractive for us since we would be able to buy about five buttons for the price of one. The shape and number of terminals is standard compared to the others but the familiarity of the mini push button was also a deciding factor since every member in our group has had experience is using them on PCB’s.

3.5 Components

3.5.1 Sensors

3.5.1.1 Barometer

Barometric pressure is the most important sensor on Eco-Sense for predicting rain because of this it is important that the sensor chosen for Eco-Sense was, at least, able to follow the requirements listed in section 2.3.6. There are three pressure sensors that are considered for this project, they are the BMP085, MPL115A1, and the HP03M.

The BMP085 is a Bosch Sensortec digital pressure sensor. This sensor comes fully calibrated and an I2C interface for data transmission. The supply voltage needed for operation is 1.8 V to 3.6 V with a ripple max of 50mV peak to peak. The BMP085 comes with four different sensor modes shown in table 3.5.1.2.1. These sensor modes have different functions to optimize either accuracy or power usage. At its lowest power consumption this pressure sensor draws an average of 3μA and at its highest resolution pulls an average of 12μA. Being that the BMP085 is a digital pressure sensor, it has its own on board analog to digital conversions. These conversions take up extra power, which averages a current draw around 500μA while conversions take place. The mode for the sensor effects the max time for analog to digital conversions shown in table 3.5.1.2.1. The standby current for this sensor is only 0.1μA at 25 ⁰C, consuming really low amounts of power.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mode** | **Internal Number of Samples** | **Current** | **Noise** | **Analog to Digital Conversion time** |
| **Ultra Low Power Mode** | 1 | 3μA | 0.06hPa | 4.5ms |
| **Standard Mode** | 2 | 5μA | 0.05hPa | 7.5ms |
| **High Resolution Mode** | 4 | 7μA | 0.04hPa | 13.5ms |
| **Ultra High Resolution Mode** | 8 | 12μA | 0.03hPa | 25.5ms |

BMP085 Sensor Modes  
Table 3.5.1.2.1

The sensing range of the BMP085 is from 300hPa to 1100hPa while retaining an average accuracy of ±1.0hPa and a max rating of ±2.0hPa over the sensing range of 700hPa to 1100hPa. This accuracy rating is also based on the temperature the sensor is operating in being 0⁰C to 65 ⁰C. The temperature range for the sensor is -20⁰C to 65 ⁰C, but with the requirements for sensor operating temperature being specified as 0⁰C to 58.5⁰C and the sensors not being used bellow 0⁰C the accuracy bellow 0⁰C is not necessary for Eco-Sense. This sensor comes with a built in temperature sensor that could be used for the ambient temperature sensor of Eco-Sense. With the use of the barometric pressure sensor to be used in a separate location from the rest of the sensor subsystem circuitry, it is recommended that the pressure sensor be pre-mounted of a PCB board to save money on self-mounting. The BMP085 can come pre-mounted on a PCB breakout board for around $10.

The MPL11A1 is a Freescale Semiconductor miniature digital barometer. This senor uses a serial peripheral interface (SPI) for communications with the microprocessor. MPL11A1 is fully operational with a supply voltage of 2.375 V to 5.5 V, where 3.3 V is the typical voltage used with the MPL11A1 and is used as the baseline for all other parameters. The current draw of this sensor averages at 5μA during sensor measurements (at one measurement per second), has a standby current draw of 3.5μA to 10μA, and a shutdown current of 1μA. MPL115A1 has a built in analog to digital conversion that takes on average 1.6ms to 5.0ms to complete. The measurable pressure range for the MPL115A1 is 50kPa to 115kPa (500hPa to 1150hPa) with an accuracy of 1kPa (±10hPa) over its operating temperature range of -20⁰C to 85⁰C. The pressure sensor also has a built in temperature sensor. The average price for a MPL115A1 is $10 to $20 with a pre-mounted sensor from Sparkfun at $25.

The HP03M is a Hope RF Electronics calibrated integrated pressure sensor. The pressure sensor includes a piezo-resitive pressure sensor and an analog to digital interface that sends pressure and temperature related voltage over an I2C connection. HP03M comes with 2 different packaging options, DIP or SMD (with metal cap). The supply voltage needed ranges from 2.2V to 3.6V with a typical value of 3.0V. The current supply needed is 500μΑ during conversion and a 1μΑ standby current. The operating pressure is within the requirements with a range of 300hPa to 1100hPa. The accuracy of pressure measurements read by the HP03M is accurate to ±1.5hPa for temperatures from 0⁰C to 50⁰C and ±3.0hPa for ambient temperatures outside this range. The full operation temperature range of this sensor is -40⁰C to 125⁰C. The average cost of the HP03M sensor is $35 when mounted on a breakout board.

For use in Eco-Sense, the barometric pressure sensor that was selected was the Bosch Sensortec digital pressure sensor, BMP085. Table 3.5.1.2.2 shows a comparison of the different sensors researched for this project need, form the table it is easy to see why BMP085 was chosen over the others. With its low power usage, excellent accuracy, and affordable cost, this sensor is exactly what is needed for Eco-Sense. Referring back to table 3.5.1.2.1 the mode that is to be used for the project is the ultra-low power mode, this is for the decrease of current use and analog to digital conversion time.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part** | **Supply Voltage** | **Average Current** | **Pressure Range** | **Accuracy** | **Shutdown Current** | **Cost** |
| **BMP085** | 1.8 to 3.6 V | 3 to 12μA | 300 to 1100hPa | ±1hPa | 0.1μA | $20 |
| **MPL115A1** | 2.4 to 5.5 V | 5μA | 500 to 1150hPa | ±10hPa | 1μA | $25 |
| **HP03M** | 2.2 to 3.6 V | - | 300 to 1100hPa | ±3hPa | 1μA | $30 |

Barometric Pressure Sensors  
Table 3.5.1.2.2

3.5.1.2 Hygrometer

The Hygrometer, used to measure absolute humidity, is the last of the sensors needed for the weather predicting capabilities of Eco-Sense. To be considered for use on the sensor subsystem, the hygrometer needed to be compliant with the requirements and specifications defined for hygrometers in section 2.3.6. The humidity sensors researched for this project are the DHT11/DHT22, the SHT1x family, HH10D, HIH-4000, and the HIH4030.

The first of the humidity sensors are the DHT11 and DHT22 humidity and temperature sensors. These sensors are digital signal output sensors that typically come pre-mounted on a PCB board. The supply requirements used for these sensors are both 3.0V to 5.5V supply voltage and supply current of 0.5mA to 2.5mA. The relative humidity sensing range of the DHT11 is 20% to 90% with an accuracy of ± 5% while the DHT22 has a better sensing range of 0% to 100% with a higher accuracy if ± 2%. The general cost of the DHT11 hygrometer is $3 for the circuit and around $5 for the circuit mounted on a breakout board and the cost for the DHT22 is $5 for the circuit and $15 mounted.

Then next set of humidity sensors are the Sensirion SHT1x family of humidity and temperature sensors. SHT1x contains three different sensors; the SHT10, SHT11, and SHT15. This family of sensors have a serial data output and can be connected to the I2C bus however, it cannot be addressed by the I2C bus. The supply voltages for all three of these sensors are 2.4V to 5.5V with a typical voltage supply of 3.3V. The measuring current of these sensors is 0.55mA to 1.0mA and sleeping (shutdown) current of 0.3μA to 1.5μA, with the minimum value of one, 8bit resolution measurement per second the average current draw of 28μA if using the highest resolution measuring at one measurement per second. The operating range of all three humidity sensors contains the maximum range of 0% to 100% while the accuracy of the measurements changes from sensor to sensor. The SHT10 has an accuracy rating of ±4.5% relative humidity over the range of 20% to 80% relative humidity, while anything outside this range will vary by ±8%. The SHT11 has a better accuracy rating of ±3.0% relative humidity over the range of 20% to 80% and will vary by ±5.0 at relative humidity outside this range. The best accuracy rating for this family of sensors is the SHT15 which has an accuracy rating of ±2.0 over the range of 10% to 90% and ±4% accuracy outside this range. The relative costs for the SHT10, SHT11, and SHT15 humidity sensors are $15, $30, and $35 respectively. All these sensors can come pre-mounted on a PCB board but are not included in the price given.

HH10D is a Hope RF humidity sensor module created using a capacitive type humidity sensor. This sensor uses two separate, and accurate, humidity chambers and transfers the data using an I2C interface. Unlike other sensors discussed, the humidity measured by the sensors can be detected a second way, an analog frequency output related to the relative humidity, this output ranges from 5.0 KHz to 10 KHz. The operating voltage required for this module is between 2.7V and 3.3V with a nominal input voltage of 3.0V. The typical current draw used for the HH10D sensor is 120μA to 180μA. The relative humidity range for this sensor is near the maximum with a range of 1% humidity to 99% humidity. This humidity range comes with an accuracy of ±3.0%. The HH10D humidity sensing module comes standard on a premade board containing all the parts needed to run the module, including all capacitors and resistors and costs only $10.

The last two hygrometers to be considered for Eco-Sense are the Honeywell HIH-4000 and the HIH-4030 humidity sensors. While similar in design specifications, these two humidity sensors are of different housing designs. The HIH-4000 has a housing design similar to that of a TO-92 transistor housing, while the HIH-4030 is designed to be implemented on a PCB board as a surface mount device. It is noticed that the HIH-4030 is a surface mountable version of the HIH-4000 and only small differences, other than physical appearance, exist. The supply specifications for these sensors are 4.0V to 5.8V with a typical current draw of 200μA and a maximum of 500μA. The relative humidity range which can be measured for both hygrometers is 0% to 100% with accuracy ratings of ±3.5%. The cost of the HIH-4000 sensor is around $20 while the cost of the HIH-4030 sensor is around $15.

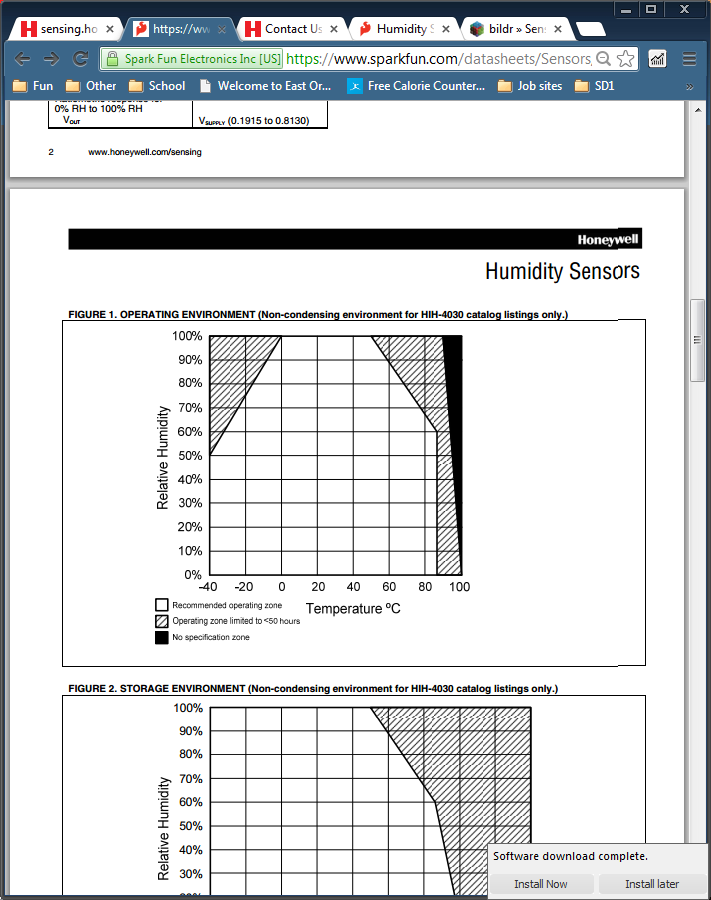
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part** | **Supply Voltage** | **Current** | **Humidity Range** | **Accuracy** | **Shutdown Current** | **Cost** |
| **DHT11** | 3 to 5.5 V | 0.5 to 2.5mA | 20 - 90% | ± 5% | 100μA | $5 |
| **DHT22** | 3 to 5 V | 0.5 to 2.5mA | 0 to 100% | ± 2% | 40-50μA | $15 |
| **SHT10** | 2.4 to 5.5 V | 28μA | 0 to 100% | ± 4.5% | 1 to 2μA | $15 |
| **SHT11** | 2.4 to 5.5 V | 28μA | 0 to 100% | ± 3% | 1 to 2μA | $30 |
| **SHT15** | 2.4 to 5.5 V | 28μA | 0 to 100% | ± 2% | 1 to 2μA | $35 |
| **HH10D** | 2.7 to 3.3 V | 120 to 180μA | 0 to 99% | ± 3% | - | $10 |
| **HIH-4000** | 4 to 5.8 V | 200 to 500μA | 0 to 100% | ± 3.5% | - | $20 |
| **HIH-4030** | 4 to 5.8 V | 200 to 500μA | 0 to 100% | ± 3.5% | - | $18 |

Table of available Hygrometers  
Table 3.5.1.2.1

Table 3.5.1.2.1 shows all the sensors discussed and the primary specifications looked observed. The sensor chosen as Eco-Sense’s humidity sensor was the HIH-4030. The HIH-4030 was chosen over the SHT10 because; even though the average current given for the SHT10 is 28μA the current needed while measuring is between 0.55mA to 1.0mA. The HIH-4030’s current draw averages at 200μA when sensing. The HIH-4030 is an analog voltage output device where its voltage output is linearly related to the absolute humidity, the formula for the voltage out related to the sensors humidity is below.

RHS is the humidity sensed by the HIH-4030 hygrometer. It should be noted that even though the datasheet has its range confined to the range of 4.0V to 5.8V, the input voltage has been confirmed operating properly at 3.3V. This humidity measure, even though the variable is RH, is not the relative humidity. Due to the relative humidity being varied by the temperature of the air containing the humidity, the humidity rating given by the hygrometer will need to be modified with the current temperature. The formula to determine the true relative humidity, RHT, is

Where TC is the temperature measured by the system thermometer in Celsius. The HIH-4030 humidity sensor has a range of operating in temperature/humidity conditions; figure 3.5.1.2.2 shows the range of temperature and humidity that the HIH4030 can operate in. This range for temperature measurements to have a 100% relative humidity is from 0⁰C to around 50⁰C. With the requirements and specifications telling us that the upper bound 58.5⁰C, this range is inadequate. Fortunately the range is still within the zone that can operate at those max conditions having to be limited to 50 hours of use in those conditions. Since the upper bound is higher than the highest recorded temperature in the U.S. there is no concern of sustaining this max condition for 50 hours. For use in Eco-Sense, the HIH-4030 is preferred to be pre-mounted on a PCB board as it has to be in the designated area for exposure to the elements. Sparkfun has created a HIH-4030 humidity sensor with breakout board for $16.95 that has been used.



Operating Conditions for the HIH-4030 Hygrometer  
Figure 3.5.1.2.2 (permission and pic pending)

3.5.1.3 Ground Moisture

Ground moisture is the measurement of the amount of water currently available in the soil and is the most important sensor reading for Eco-Sense. With ground moisture readings, Eco-Sense will have a baseline for when watering is needed and a general timeline for how long until the next watering session. For Eco-Sense the moisture sensor was created from scratch. To create a moisture sensor it is necessary to investigate common methods of determining soil moisture sensors. Several common techniques are the Oven Drying technique, Nuclear Techniques, Electromagnetic Techniques, and other Miscellaneous Techniques. Once these techniques are researched a proper technique can be determined from following the requirements and specifications listed in section 2.3.6.

The Oven Drying technique is the technique with which the standard calibration of all other soil moisture sensing techniques is determined. Being the standard for which all soil moisture content is calibrated to, this technique is the most accurate measurement of absolute moisture content. This technique measures moisture by comparing the weight of a soil sample to the same sample weight when completely dry. Although the most accurate and being a relatively cheap method, the oven drying technique takes a minimum of 24 hours to procure results, this technique also cannot be automated.

There are three soil moisture detection techniques that can be classified as nuclear sensing techniques; these are neutron scattering, gamma attenuation, and nuclear magnetic resonance. Neutron Scattering is a technique that determines the water content by emitting fast neutrons from a radioactive source and measuring the proportion of neutrons that are slowed down or “thermalized”. This method can determine the moisture content very fast, around 2 minutes, but due to the cost and hazard of radiation is not recommended for this project. Gamma Attenuation is a method that measures the soil water content by assuming that any scattering and absorption of gamma rays, applied to a soil sample, is caused by the density of the sample it passes through. Assuming the gravity on the soil sample remains relatively constant any changes to the density of the sample is caused by moisture changes. Nuclear Magnetic Resonance is the last of the nuclear moisture sensing techniques. This method of soil moisture detection is a technique similar to that of an MRI. When a soil sample is subjected to a strong magnetic field, hydrogen atoms, abundant in water, aligns there rotating axis with the magnetic field. When a radio frequency (RF) signal is introduced at a 90 degree angle to the magnetic field, the axis of the hydrogen’s no longer align and change their axis. When the RF signal is turned off, the axis of the hydrogen’s realign and emit a readable RF signal that can be used to determine the concentrations of water in the soil sample. This method requires very bulky and costly equipment making it less desirable.

As the name suggests, electromagnetic sensing techniques use electromagnetic properties to measure ground moisture. There are four electromagnetic techniques that are to be considered for Eco-Sense, these are resistive sensors, gypsum resistive sensors, capacitive sensors, and time domain reflectometry. A Resistive Sensor can be created based on the properties of water and resistance. Pure distilled water has no resistivity, but when adding contaminants, such as soil, resistivity becomes measurable. Soil resistivity is proportional to its soil moisture content. The resistivity between two probes can be used to find a general measurement of the soil moisture content. This technique provides an instantaneous approximation of the moisture content. Gypsum Resistive Sensors are similar to regular resistive sensors except for the probes are incased in a porous block, usually of gypsum. When the device is buried in soil it will, eventually, have the same matric potential as the soil. This allows for measurement of the average moisture content as it takes up to 3 hours for new moisture to affect the readings. Capacitive sensors are used to determine the soil water content by measuring the dielectric constant of the soil. The dielectric constant of the soil is directly proportional to the water content of the soil. By charging and discharging two plates, forming a capacitor in the soil, the capacitance can be determined by the charge and discharge time (time constant), which can then be used to determine the dielectric constant. This provides instantaneous soil moisture measurements with a high degree of precision. The last of the electromagnetic moisture sensing methods is time domain reflectometry. Time Domain Reflectometry determines the soil water content by measuring the propagation of electromagnetic waves. The propagation of these signals depends on soil properties, primarily soil water content. While usually used for finding the location of a brake in fiber optic cables, the moisture content is measured by the time it takes an impulse to be reflected back to the source.

There are many other moisture sensor techniques, but for Eco-Sense three more are to be discussed for use. These extra miscellaneous techniques are the tensiometric technique, hygrometers, and remote sensing techniques. The Tensiometric technique is a method of using a tensiometer to determine the capillary tension in the soil. Although an extremely cheap method, this method does not measure moisture content in the range specified by the requirements, and will not work for lawn moisture sensing. Hygrometers measure moisture content, also known as humidity. Special hygrometers can be used to measure moisture content in the soil by using the relationship between moisture content, in a porous material, and the relative humidity in the immediate surroundings. They utilize chemicals and thermal principles to measure the relative humidity. This is a destructive technique as it slowly erodes the soil with use and each material to be tested (soil samples) need to be specifically calibrated beforehand. The last of the miscellaneous techniques to be discussed is remote sensing techniques. Remote sensing techniques are non-contact techniques that are used to measure water content in soil. Moisture content is measured by either the amount of electromagnetic energy being reflected or emitted from the soil surface. The intensity of this radiation is proportional to soil moisture, dielectric properties, and soil temperature. Most common devices to implement this type of sensing are satellites, radars, or planes and are good for monitoring soil moisture over a large area. This method is far too bulky and costly to be used in Eco-Sense.

To decide which sensing method should be used several factors need to be considered. The requirements and specifications for the moisture sensors described in section 2.3.6 are defined as; response time of the sensors, the possibility for automation, measurement in desired location, small enough to fit the design, low cost, and sensing accuracy. Table 3.5.1.3.1 shows the decision matrix in deciding which technique to use. The only method that meets all requirements for this design is the resistive (general) circuit.

The resistive sensor was chosen for Eco-Sense because it provides the cheapest means of moisture sensing in the soil while providing the necessary specifications. It works on the principle of conductivity. Dry soil does not conduct electricity very well, but when water is added the conductivity of the soil increases. By placing two probes into the soil a determined distance away, they create a complete circuit through the soil. When passing an electric current through the soil, the soil will provide resistance proportional to the moisture content. The more water in the soil the less resistive the soil becomes. There are two issues in particular that will affect the design of the moisture sensor for Eco-Sense; these are the effects of temperature on the conductivity of the soil, and electrolysis.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sensor type** | **Response time** | **Can be Automated?** | **Measure in needed location?** |  | **Relatively Low Cost?** | **Accuracy (1-10)** |
| **Oven Drying** | 24 Hours | No | Yes |  | Yes | 10 |
| **Neutron Scattering** | 2 Minutes | Yes | No |  | No | 6 |
| **Gamma Attenuation** | >1 Minute | Yes | Yes |  | No | 8 |
| **Nuclear Magnetic Resonance** | <1 Minute | Yes | Yes |  | No | 8 |
| **Resistive (General)** | Instant | Yes | Yes |  | Yes | 7 |
| **Resistive (Gypsum)** | 3 Hours | Yes | Yes |  | Yes | 7 |
| **Capacitive** | Instant | Yes | Yes |  | No | 9 |
| **Time Domain Reflectometer** | 30 seconds | Yes | Yes |  | No | 9 |
| **Tensiometer** | 3 Hours | No | Yes |  | Yes | 5 |
| **Hygrometric Technique** | 3 Minutes | No | Yes |  | No | 7 |
| **Remote Sensing** | Instant | Yes | No |  | No | 8 |

Moisture Sensor Decision Matrix  
 Table 3.5.1.3.1

Soil temperature dramatically affects the conductivity of the soil. As the heat rises the resistance seen by the sensor decreases. Soil temperature rarely have the same temperature as the ambient temperature of the air so to compensate for the variation the soil heat creates, a soil thermometer needs to be placed near the sensor to provide soil heat measurements to the central HUB for processing. This change in resistivity seen by the sensor is not linear so to obtain the information on how temperature affects the resistivity measurements, several experiments with a prototype sensor will be necessary to calculate the percentage change of the resistance to heat formula.

Electrolysis takes place when a DC current constantly flows between two pieces of metal, such as the sensor probes. This causes etching on the sensor probes which will eventually cause changes in the resistance seen by the sensors, or degrading the sensors to the point they can no longer be used. To prolong the use of the probes the moisture sensor will only be used right before transmission of data to the central HUB. With this process the sensors will be used much less, reducing the effects of electrolysis greatly.

The resistance seen by the sensor is directly related to not only the soil moisture content, but also the distance between the probes and soil composition. With these factors to take into consideration certain calibrations are needed. The distance between the sensor probes needs to be consistent to simplify the calibration calculations. Individual soil composition affects the resistance seen by the moisture sensors. To compensate for this change, the moisture sensor needs to be calibrated to the soil Field Capacity. As discussed in section 3.2, this point is the ideal amount of water to be present for plant growth. To obtain this Field Capacity measurement we will use the buck method, which is to use a bucket with 5 gallons of water and poor it on a small portion of the soil to be tested. After 24 hours the moisture level in that location is approximately at the Field Capacity levels. Inputting this measurement into the controller as the Field Capacity point completes the calibration for soil composition.

3.5.1.4 Thermometer

Temperature sensors are important for Eco-Sense as there are multiple uses, relative humidity calculations, weather prediction, and ground temperature compensation for moisture sensor. Both the relative humidity calculations and weather prediction aspects of Eco-Sense can be used with the same sensor but because of ground temperature differing from the ambient temperature of the surrounding air, a different thermometer will have to be implemented for ground temperature. These two different temperature systems will be referred to as the system thermometer and the ground probe thermometer.

The system thermometer used for Eco-Sense is the built in thermometer of the BMP085 barometric pressure sensor. Without any additional circuitry or power usage the BMP085 can measure temperatures from 0⁰C to 65⁰C with an average accuracy of ±0.5⁰C near room temperature and up to ±1.0⁰C over the entire sensing range. These specifications are within the range set by the requirements and specifications for temperature sensors in section 2.3.6. Using the BMP085 pressure sensor reduces the total cost of Eco-Sense by reusing existing parts instead of buying a new sensor.

The ground probe temperature sensor cannot be the same sensor used for the system thermometer because the ground temperature sensor needs to be buried, and exposed, under the soil. There are four different temperature sensors that are considered for the ground probe thermometer. These temperature sensors are the AD22103, TMP36, LM45, and the AD22100K temperature sensor. It is essential that the temperature sensors follow the requirements put forth in section 2.3.6 while being able to be small enough to make into a sensor probe. The first sensor is the AD22103, Analog Devices series of voltage output temperature sensors. This sensor is an analog voltage out sensor whose output is linearly proportional to the voltage in. The operating voltage of this sensor is 2.7V to 3.6V with an operating current of 350μΑ to 600μΑ with a typical current draw of 500μΑ. The temperature sensing range of the AD22103 is 0⁰C to 100⁰C with a typical accuracy of ±0.5⁰C. AD22103 comes in 2 different packages TO-92 and SOIC which cost about $3.50 each. The TMP36 is an Analog Devices series low voltage temperature sensor; it has an input voltage requirement of 2.7V to 5.5V. The supply current for this sensor averages at 50μΑ. The shutdown current is typically 0.01μΑ with a max shutdown current of 0.5μΑ. The sensing capabilities of the TMP36 are impressive with a temperature sensing range of -40⁰C to 125⁰C with a typical accuracy rating of ±1.0⁰C. The TMP36 comes in three packages, TO-92, SOIC\_N, and SOT-23 with their costs being between $0.80 and $1.50. The next temperature sensor is National Semiconductors LM45. The LM45 is a precision temperature sensor available SOT-23 package. The supply voltage of the LM45 temperature sensor is 4.0V to 10.0V with a maximum current draw of 10μΑ. The temperature sensing range of this thermometer is -20⁰C to 100⁰C with an accuracy rating of ±3.0⁰C. The average cost for the LM45 temperature sensor is $2.5. The last sensor researched for Eco-Sense is the Analog Devices AD22100K voltage output temperature sensor. The AD22100K uses an input voltage of 4.0V to 6.5V and pulls a current of 500μΑ to 650μΑ during normal operations. The AD22100K has the largest temperature range of all the sensors with a range of -50⁰C to 150⁰C and accuracy of ±2.0⁰C. Contained in a TO-92 package, the cost of this sensor averages $1.90.

The sensor chosen to be the temperature sensor, for Eco-Sense’s ground temperature probe, was the Analog Devices TMP36. Table 3.5.1.4.1 shows the comparison between the temperature sensors discussed for the ground moisture sensor. The TMP36 was chosen due to its low current usage, low cost, and good accuracy rating. This sensor will be bought in its TO-92 package form and used to create a ground sensor probe.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Supply Voltage** | **Average Current** | **Temperature Range** | **Accuracy** | **Cost** |
| **AD22103** | 2.7 to 3.6V | 350 to 600μA | 0 to 100⁰C | ±0.5⁰C | $3.50 |
| **TMP36** | 2.7 to 5.5V | 50μA | -40 to 125⁰C | ±1.0⁰C | $0.80 to $1.50 |
| **LM45** | 4.0 to 10.0V | 10μA | -20 to 100⁰C | ±3.0⁰C | $2.50 |
| **AD22100K** | -50 to 150 | 500μA to 650μA | -50 to 150⁰C | ±2.0⁰C | $1.90 |

Ground Temperature Sensors  
Table 3.5.1.4.1

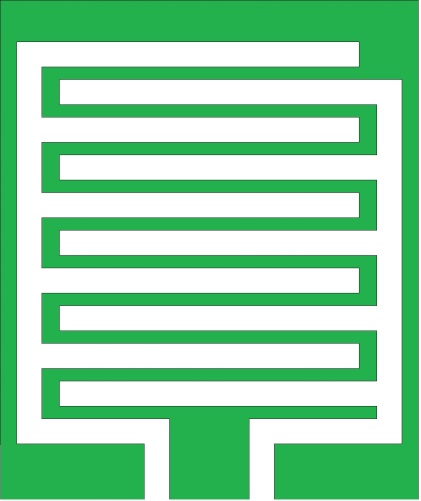
The TMP36 temperature sensor is an analog temperature sensor which reports the temperature it measures through an output voltage that is linearly proportional to its input voltage. The datasheet for the TMP36 gives that at 25⁰C and an input voltage of 3.3V, the output voltage should be 700mV and vary at a rate of 1mV/⁰C. Using this information, the formula for converting the output voltage given by the TMP36 temperature sensor to degrees Celsius (Tc) is given bellow.

With this linearly varying voltage the micro-processor can accurately read the temperature form the ground temperature sensor.

3.5.1.5 Rain Detector

The primary focus of the Eco-Sense system is to automate the proper watering maintenance of a household lawn while saving money. One of the major ways to save money is to not be watering when it is raining out. The weather detection system built into Eco-Sense should be able to sense the oncoming presence of stormy conditions, but as a backup precaution a way to detect if it is raining is preferred. Any rain detection system implemented with Eco-Sense needed to follow the specifications defined in section 2.3.6 for rain detectors. The rain detector system will not be a prepackaged part, but a system created for the sole use of Eco-Sense.

The first component of the rain detection system is the component that is exposed to the rain and tells the microcontroller that there is rain. The most common ways to tell if there is rain are infrared detectors and resistivity sensors. Infrared rain detectors are very common due to the fact that no components have to be exposed to the weather. The infrared sensors detect rain by bouncing infrared light off of plastic or glass, as water accumulates on the outside of the plastic or glass the amount of infrared light reflected back to the receiver changes. The price for using this method varies with the application of the technology, rain sensing for lawn watering porpoises can cost anywhere from $20 to $200 depending on how quickly the sensor can detect and react to rain. The other common method for rain detection is using the principles of resistivity. Similar to the moisture sensor, the resistance seen by using this method is proportional to the amount of water on the sensing element. The resistance of the sensor decreases as more water is added. The sensing element is, usually, an etched PCB board that acts like an open circuit until rain water connects the circuits together. Figure 3.5.1.5.1 shows a typical PCB layout used for rain detection. It is our intention of using the second method, resistive rain sensor, as the sensing mechanism for the Eco-Sense’s rain detection feature. A small coated premade board would prove to be a cheaper option due to the creation of these boards in high volume, and can be purchased around the cost of $5.



Typical PCB layout for a Resistive Rain Sensor  
Figure 3.5.1.5.1

For the use of this sensor there are to methods two possible methods for delivering the signal to the microcontroller, analog signal or a digital signal. An analog signal would allow for Eco-Sense to know the exact amount of moisture present on the sensor and a digital signal would only tell the Eco-Sense if there is water on the sensor or not, depending on the threshold. With the addition of so many analog sensors it is necessary to use a digital port to free up more space and Eco-Sense does not need to know how bad it is raining only that it is raining. To create the digital signal needed a comparator was used to compare the voltage of the resistive sensor, in a voltage divider, and an adjustable reference voltage. The LM339N voltage comparator was used for this purpose as it allows for an input and supply voltage up to 36V and is less than a dollar per chip, of four.

The dew point is the temperature which water vapor in the air will, given a constant barometric pressure, condense into water and evaporate at the same rate. This usually occurring in the morning and can possibly cause an issue with the rain detection system. If the dew becomes heavy enough it could give a false positive to the rain detector. To compensate for this the central HUB microcontroller will have to measure the dew point and determine if there is dew present, if the system wants to water. A common formula used to approximate the dew point with the temperature and relative humidity (M*agnus formula)* is shown below.

Tdp is the dew point calculated by this formula, RH is the relative humidity, and T is the temperature in Celsius. The constant b = 17.67 and the constant c = 243.5˚C. If the temperature is less than or equal to this dew point temperature Eco-Sense should not water due to the inability to tell if the rain detector is being triggered by dew or rain.

The mounting of the rain sensor contains two sections. The circuitry containing the comparator is to be mounted on the main circuit board of the sensor subsystem. The second section of the rain sensor is the resistive rain sensor. This rain sensor is mounted on the outside of the sensor housing, being exposed to the rain. While being exposed to the rain the rain sensor needed to be protected from the watering system to avoid a false positive for rain.

3.5.2 Microcontroller

Since the cost developing microcontrollers has dropped dramatically in the last few decades, there is a large selection of microcontrollers available -- all of them with similar features. The microcontroller is the controller for all other systems. It is responsible for everything from data gathering, relaying gathered information, and power management of the sensor node system. Choosing an appropriate microcontroller is important as this will use the majority of the power while the node is awake. Any extraneous functionality of the microcontroller board will use more power than is needed by the system and should be avoided. An example would be if the wireless sensor network does not need any extensive processing and calculations run at the sensor nodes, then a low power, low performance microcontroller would be much more appropriate then a processor that you would find in a computer. Power consumption is an important specification to consider as the system will have a limited power supply and a longer life is desirable. The choice of a microcontroller should be chosen by the application of the wireless sensor network. And to best match the microcontroller performance to what is needed by the network.

The sensor node communicates processes and gathers all sensor data from each sensor. The sensor nodes needed processing units in order to do this. The microcontroller of the sensor node determines both the energy consumption as well as the functionality of a sensor node. A sensor node lifetime is determined by what the battery can deliver; power conservation is the primary metric for selecting an appropriate microcontroller. To save power, some microcontrollers offer different power modes with varying amounts of functionality and power usage for each level. Such as very lower power but only running an asynchronous timer to wake the system up at scheduled times (sleep mode). It is also important to know that switching (transitioning) between the different modes have a time overhead (latency) which is taken into account when determining the total power consumption of the system over a period of time. A good goal is to have the wireless sensor node have a low duty cycle of around one percent. This is done by scheduling events for the future then sleeping as much as possible. Microcontroller can do this with an asynchronous timer using a clock and leave the other peripheral components and core turned off. When the timer gets to the scheduled time, it will use an interrupt to wake the rest of the system up.

FPGAs (field programmable gate arrays) were not considered for the following reasons. For the low powered FPGAs on the market (such as the BAYSIS board), they use more power in both sleep and active modes of operation when compared to a normal microcontroller. Also unlike microcontrollers, it is not possible to turn off separate blocks in the FPGA to conserve power like a microcontroller can do. The compilers for FPGAs do not use traditional programming languages such as C to program with which was not a good idea for this project for reasons that will be discussed later.

Nowadays, microcontrollers include not only memory and processor, but non-volatile memory and interfaces such as analog to digital converters, SPI, I2C, UART, counters and timers. The microcontrollers can interact with sensors and communication devices such as a short range radio. There are many types of microcontrollers, ranging from 4 to 32 bits, varying the number of timers, bits of ADC, power consumption, etc. but for this project, we will compare several popular microcontrollers including the MSP430, ATMega128, ATMega328, PIC16, and PIC18.

3.5.2.1 I/O

I2C or Inter-Integrated Circuit is multi-master bus that allows multiple devices to send data on the same line. In this protocol, there are four connections: a 5 or 3.3 volt VCC, a 0 volt ground, a Serial Clock (SCL) and a Serial Data (SDA). This protocol can use the same lines for all devices because it uses an acknowledge scheme to request permission from the receiver (master device) to send its data. This addressing scheme allows a device to determine if it is being used by the master or if the (master device) receiver is busy.  A master sees that a slave device is present when the slave responds to its address by sending an acknowledgement bit (putting SDA low). However, if no such addressed device exists, the SDA line is put high, and this is interpreted by the master device as a not acknowledgement and will not transmit its data. For our project, using this protocol would be very beneficial. It would allow multiple sensors to communicate with microcontroller without the extra wires needed for each sensor. As everyone knows, electronics and water tend not to get along well, so as this project is exposed to Mother Nature, we are waterproofing the container containing the electronics. But some of the sensors require direct access to the environment to work properly. These sensors will have to be connected through wires from the waterproof section to the exposed part of the container. Having as few wires as possible by using the I2C protocol will reduce the risk of water entering the section that can easily short circuit. An example of using this would be a barometric pressure and temperature sensor all in one package. The data created from each of the respective sensors will be sent on the SDA line to the microcontroller. There it will be interpreted and split into the temperature and pressure data points. These four wires would replace eight wires, plus all of the soldering and waterproofing for the additional wires. This would make designing and then building the system easier.

To allow multiple sensors to be connected to out sensor nodes it’s important to have the needed connectivity. Different sensors can have different requirements for giving their information to the microcontroller. Some have an analog interface, others a digital or I2C connection. So a microcontroller that supports this is important. All of the microcontrollers feature a number of digital I/O ports. The MSP430 has 40 digital pins, the ATMega128 has 53, the PIC18 has 36, the PIC16 has 33, and the ATMega328L has 28. All of the devices contain hardware UART, which stands for Universal Asynchronous Receiver / Transmitter, and it is a piece of computer hardware that translates data between parallel and serial forms. The ATMega128 has hardware supporting I2C and SPI, 4 timers and a 10 bit, 8 channels ADC. The ATMega328 has hardware supporting I2C and SPI, two 8bit timers, one 16 bit timer, six PWM channels, and eight channels ADC. The MSP family has an extensive set of I/O options ranging from low-power devices with little I/O to devices with multiple ADCs, serial I/O support and even DMA controllers, although the particular device considered here does not have any of these. The PIC18 has hardware which can either carry out I2C or SPI, 4 timers and a 13 channel, 10 bit ADC. The PIC16 is the same, but only 3 timers and 8 ADC channels.

3.5.2.2 Compiler

There are many languages used in programming of a microcontroller including assembly, C, C++, Verilog etc. But choosing a language that is both familiar to what our group knows and easy to use is important. By using a known language, we can spend more time implementing functionality into the system rather than spending the time learning a new language and debugging it. The MSP430, ATMega128L, and ATMega328L are supported by a free port of gcc. The PIC16 and PIC18 have their own commercial compilers available, although the limitations of the load/store architecture make these relatively difficult to use. The existing compilers for the PIC architectures use a compiled stack; all others use a dynamic stack.

Comparing AVR and PIC microcontrollers: In the memory architecture for a PIC microcontroller, it uses bank registers to access 256 bytes of memory at a time. For AVR microcontrollers, there no need to access the data memory because unlike PIC microcontrollers that have only one general purpose register, AVR microcontrollers have 32 general purpose registers. AVR has SRAM that the stack is contained within the SRAM in order to build the hardware stack. PIC does not have it. The PIC18F and Atmega have hardware multipliers whereas the PIC16F don’t having hardware multipliers. PICs have high clock speed but it is divided by four to give the actual instruction rate. Many PIC microcontrollers take multiple (4) clock cycles to run an instruction whereas the AVR can run instructions in 1 clock cycle. By comparing ATMega128L and MSP430 have rich instruction set and also it has a wide range of arithmetic instructions sets, many addressing modes. But in PIC18 and 8051 microcontrollers are limited. But PIC16 has added with carry instructions. The MSP430 has 40 digital pins, ATMega128L has 53 pins, ATMega328L has 28, PIC18 has 36, and PIC16 has 33.

3.5.2.3 Microcontroller Comparison Table

Looking at this table shows that many microprocessors have a very low powered down power usage and others do not. Some controllers such as the PICs have only 1 byte of memory where others have 128 kilobytes. For our project, having enough flash memory to write our program to, and store any data collected is important. Having the same or similar microcontroller for both the sensor nodes and central HUB subsystem would simplify things so that we would be able to use the same development boards for prototypes and keep costs down. So not only does the microcontroller have to be low powered for the sensor nodes, but have enough pin-outs for including peripherals such as an LCD screen, momentary switches (push buttons), LED indicators, and an SD storage to keep a log of the data. It will have to also be able to run all the processing of the data from the sensor nodes. This would require a large enough RAM space to run the calculations in. Also, since ideally we use the same microcontroller for both the central HUB subsystem and sensor nodes, having enough RAM to run data processing is important. Finally, the powered down and active power usage should be low. Many have .1 μA for powered down, but only a few have less than 3 mA active current usages. The MSP430 F149 and ATMega328 have really low power usage, but the cost of the MSP430 F149 was around 2-3 times as much as the ATMega328.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Characteristic** | **AT90L**  **S8535** | **Atmega**  **103L** | **PIC16F8X** | **MSP430**  **F149** | **At91M**  **42800A** | **MC68HC05PV8A** | **80C51RD** | **EM6603** | **MC9328MX1** | **ATMega328** |
| **Bits** | 8 | 8 | 8 | 16 | 16 | 8 | 8 | 4 | 16 | 8 |
| **Flash kB** | 8 | 128 | 68 | 60 |  |  | 64 |  |  | 32 |
| **RAM B** | 512 | 4k | 1 | 2k | 8k | 192 | 1k | 96x4 | 128k | 2k |
| **ADC bits** | 10 | 10 |  | 12 | 08 |  |  |  | 13 | 2x8 |
| **Timers** | 3 | 3 | 1 | 3 | 6 | 1 | 1 | 1 | 2 | 2+1 |
| **Operating Voltage** | 4-6 | 2.7-3.6 | 2-6 | 1.8-3.6 | 2.7-3.6 | 3.3-5 | 2.7-5.5 | 1.2-3.6 | 1.62-3.3 | 1.8-5.5 |
| **Power Active** | 6.4 mA | 5.5 mA | 2 mA | .4 mA |  | 4.4 mA | 16 mA | 1.8 μA | 90 mA | .2 mA |
| **Power Idle Mode** | 1.9 mA | 1.6 mA |  | 1.3 μA |  | 1.95 mA | 4 mA | .35 μA | .16 mA | .75 μA |
| **Powered Down** | 1 μA | 1 μA | 1 μA | .1 μA |  | 485 μA | 50 μA | .1 μA |  | .1 μA |

Table 3.5.2: microcontroller comparison chart

3.5.3 Solenoid

Solenoid valves are used in Eco-Senses’ design for controlling when water should be distributed to the rest of the system. Solenoid valves that operate on electricity are typically closed by default and open when a certain threshold voltage is applied to the device. Currently, solenoid valves for water distribution cannot limit the rate of water flow when open which results in constant water pressure throughout the system. This major impact will determine the time the valve should stay open, whether it’s safe to keep the valve open for different time periods, and how often the valve can be toggled.

The solenoid valve used for this project requires 12V to turn on, which is more voltage than our system will have for the majority of the components. The solenoids can successfully run off of 7.4V. The voltage range that Eco-Sense can take advantage of is fifteen percent for this part. The solenoid opens and closes at different speeds depending in the supplied voltage, but since this design isn’t directed towards speed, the system can slowly open the valve and still achieve the same result. The average lifespan for a solenoid valve can range from 200,000 to 1,000,000 uses. Assuming lowest lifespan and high frequency of watering (once per day), one valve will last the lifetime of the product.

Solenoid valves come in different orientations and designs depending on the use. Eco-Sense will being using the valve for water control, so no specific orientation is needed for correct operation. Most valves require about 3 PSI to work as intended. According to PSIFactor, a typical garden hose will release a pressure of about 40 PSI, well above the minimum operating PSI of the valve. Threading is an important feature to the compatibility between solenoid valves and garden hoses. Most of the valves being researched have a British Standard Pipe (BSP) threading. Garden hoses have their own threading known commonly as Garden Hose Thread (GHT). BSP threads are usually 1/2” to 3/4” while GHT can be 5/8” to 1”. If this design were to connect a BSP outlet with a GHT inlet, water may leak through the threading and cause inefficiency in water consumption. Since Eco-Sense is about water management, it was decided to also look into BSP to GHT adapters in case they are needed. A comparison of water solenoid valves can found in table 3.5.3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Valve Model** | **AQT15SCB** | **AQT15SP** | **Orbit Yard Water Valve** | **Water Valve ID997** |
| **Thread Size** | 1/2” BSP | 3/4” NPT | 3/4” GHT | 1/2” NPT |
| **Material** | Brass | Plastic | Plastic | Plastic |
| **Voltage Required** | 12V | 12V | 24V | 12V |
| **Voltage Range** | 15% | 15% | 15% | 15% |
| **Lifespan** | >5 years | >5 years | >5 years | >5 years |
| **Purpose** | Water, Oil | Water | Water | Water |

Table 3.5.3.1: Water valve comparison

The water valves shown above each have different threading, making the choice more difficult. There seems to be no standard to threading for water control devices. British Standard Piping (BSP), National Pipe Thread (NPT), and Garden Hose Thread (GHT) are the three types listed above. GHT and NPT are both United States threads, so the AQT15SCB will be ruled out for various reasons. The threading is not compatible with any US regulated spigots, and the purpose of the valve includes oil which results in more expense to be paid. The other three valves the requirement for Eco-Sense, so adapters are shown below in table 3.5.3.2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Adapter** | **Brass Coupler 196-E** | **Dixon BA776 Adapter** | **Brass Connector 196-D** |
| **Thread Input** | 3/4” GHT | 3/4” NPT | 3/4” GHT |
| **Thread Output** | 3/4” NPT | 3/4” GHT | 1/2” NPT |
| **Material** | Brass | Brass | Brass |

Table 3.5.3.2: Thread adapters

Finding adapters for the solenoid valves is a chore due to limited availability. It was thought that with such a variation in thread sizes that adapters would be plentiful, unfortunately they were not. Another major problem with the water valves are the datasheets supplied with them. The datasheets have no useful information such as current limits, safe durations of operation, and the correct thread type. Due to misleading datasheets, it proved to be more work to research the correct thread type of each part than to research which part to use. Since it’s too difficult to make out the correct threading type of the listed adapters, this project will stick with the AQT15SP Water Valve.

The AQT15SP Water Valve is part of an irrigation kit for homes. It contains the normal features of the other solenoid valves such as voltage activation and water control. The valve already contains garden hose threading so no adapter should be needed for this project. The valve runs at 12V rated, but is confirmed to work at 7.4V.

Since the water valve is GHT compatible, this project will take advantage of a standard home spigot and also a garden hose which most homes will have. By using available resources from the user, it reduces the cost in which the user must provide to implement this system. This project also uses a spike based sprinkler for demonstration purposes. The spike based sprinklers will provide an easy connection to any garden hose and allow us to test data throughout the yard with a single board The cost for spike based sprinklers A standard sprinkler will be used, as shown in table 3.5.3.3.

|  |  |  |
| --- | --- | --- |
| **Sprinkler** | **Melnor Pulsating Sprinkler** | **Orbit Spike Sprinkler** |
| **Fitting Size** | 3/4” | 3/4” |
| **Maximum Pressure** | 100 psi | 100 psi |
| **Position** | Above Ground | Above Ground |
| **Rotation** | 180 Degrees | 360 Degrees |

Table 3.5.3.3: Sprinkler comparison

After reviewing a couple common sprinklers, the project decided to use the Orbit Spike Sprinkler. This sprinkler is a higher quality sprinkler for mounting and slightly cheaper. The major benefits of the Orbit Spike Sprinkler includes high area coverage (5000 sq. ft.), adjustable patterns, adjustable distance, and can change the degree of rotation. These features are attractive due to customizability in watering needs.

3.6 Power

3.6.1 Wall Power

Our central HUB subsystem needed to remain on continuously. In order to not require a large battery to keep it on for a long period of time, we needed external power. A wall power adaptor allowed the central HUB subsystem to remain on. For our central HUB subsystem, we needed the adaptor to be between 6-12 volts. Also, it needed to provide enough current to charge the small internal battery and run the hardware including the LCD. The current needed is around 250-300mA. This current allows for multiple components to run continuously.

Due to the fact that there are many cheaply made AC wall adaptors, the DC output voltage may not be stable enough to run the microcontroller. The voltage of the adaptor should be near what is required, but it doesn't matter too much, because we will used a voltage a regulator, and it has to stay at the same level all of the time. The microcontroller has many transistors firing inside and run at voltages proportional to the power supply voltage, so if it changes, then the transistor's voltage does too. To prevent noise from the AC adaptor, the use of bypass capacitors on the power supply and ADC reference pins. This also prevents the microcontroller from jumping to some odd address, or resets with no warning, or just plain locks up.

Power needed: The microcontroller itself only uses a small amount of power. But why would we build a circuit for a microcontroller with no connection to the outside world? It would serve no purpose. Adding any components to interact with the outside world increases the power significantly. For example, adding 10mA LED at 5 volts then the power goes up by 50mW. If the AC adaptor or the battery's voltage is has too high than what is set on a linear voltage regulator, then the energy is wasted as heat. And for the sensor nodes, every drop of power is precious. A switching regulator was implemented, since it is be much more efficient than a linear regulator. The switching regulator may be more complex, and more expensive, but it allows the system to remain running for longer period of time.

3.6.2 Batteries

For our system, having a power outlet in a lawn is highly unlikely; therefore using a battery in the sensor nodes is how we provided power. We do not want to have to continuously recharge or replace the battery so having a high energy density and capacity was important. There are several types batteries, for the rechargeable there are Nickel Cadmium, Nickel-Metal hydride, Lead-Acid, Lithium Ion, and Lithium iron phosphate. For the non-rechargeable batteries, the most popular ones are the alkaline and lithium ion ones, but for our system, needed to be using rechargeable ones.

The factors what where focused on were capacity, cost, size, and weight. The battery needed to be light and a small size around 2x2x4 inches since the modules are portable and having a large heavy battery would limit the portability of the system nodes. All chemistries have advantages and disadvantages that must be weighed before choosing battery pack chemistry.

Types: Nickel Cadmium (NiCad)

The NiCad rechargeable batteries were one of the first rechargeable batteries. The battery is made from a nickel oxide as the cathode and cadmium as the anode, both submersed in a while potassium hydroxide as the electrolyte. For storing, it may be left charged or uncharged. Their energy capacity is about 10-20% of the capacity of alkaline of a similar size. The typical run time of a NiCad would be around 10-20% of what the alkaline battery could do. This type of battery contains toxic metals that require 36 hazardous waste disposals. However, the biggest disadvantage of this battery is that it suffers from memory effect. The memory effect is when the batteries come weaker with continued use (from charging and discharging), this tends to happen when the actual battery has seen some use and they do not hold has much of a charge on the next charge.

Ni-MH: Over the past few decades, Ni-MH batteries have become more popular. These are commonly seen in handheld GPS units, flashlights, and AAA to D rechargeable batteries and many more devices. The nickel-metal battery is very similar to the NiCad battery but with a higher capacity and no super toxic components. Some advantages are that it has a high energy density with a flat discharge curve, a wide operating temperature range, and able to do a rapid charge, which would charge the battery in around an hour or so. The battery is a much safer battery than the lithium ion battery in case of any accident or abuse due to active chemicals. Some disadvantages to this battery are the fact that it has lower current then the NiCad battery.

Li-ion / Li-Poly: Most portable devices today run off of lithium ion batteries. These are one of the most popular batteries because they have one of the highest energy densities for rechargeable batteries available. The advantages of the lithium ion batteries are why they are used in almost the common electronic devices. They include the following: They are lighter than many of the other rechargeable batteries of similar size. Even the electrodes of the battery attribute to the light weight, since they are made of carbon and lithium. These batteries can store a lot of energy in its atomic bonds, since lithium is a highly reactive element. The lithium ion batteries also hold their charge while only losing about 5% of it charge per month. This type of battery has no memory effect, they can handle hundreds of charge and discharge cycles. The lithium ion battery also has disadvantages. Some disadvantages include: The lithium ion battery will start to degrade as soon as they are sent out of the warehouse. In just recent news, they have discovered that the lithium ions batteries have the ability to burst into flames. This is not a common issue where it would happen every time you purchased a pack, but around two or three packs out of every million sold. The Li-Poly or lithium polymer battery is the same as the Li-ion for all respects in power usage. However, it is able to be produced in different shapes and not just rectangular blocks.

Rechargeable Alkaline: Alkaline batteries have very long shelf lives. These batteries do not suffer from the memory effects that other batteries may suffer from. The rechargeable alkaline battery has a higher standard voltage that the standard alkaline battery. The lifetime of alkaline rechargeable battery is almost up to 30 times greater than the non-rechargeable kind. Non-rechargeable batteries could usually cost around 3 to 6 dollars for a pack of six, while the rechargeable cost around 20 dollars with the charging kit included. The alkaline battery that is made to be recharged is made out of alkaline manganese.

Lead acid: The lead-acid battery was invented in 1859 by French physicist Gaston Plant, being one of the oldest rechargeable batteries. They use a chemical reaction of lead, lead oxide and sulfuric acid to produce voltage. It has the second lowest energy to weight ratio, the nickel-iron battery being on top. One of the most important characteristic of a battery is the lifetime, the maintenance required, and the depth of the discharge. The Lead acid batteries typically have a energy efficiency of 70% and a columbic efficiency of a 85%. The battery is almost 2 to 4 times less expensive than the NiCad batteries. The lead-acid battery can also retain nearly their full charge for about two months when they are unattached to a charger. The advantages of a lead-acid battery are numerous. The battery is tolerant to abuse and tolerant to overcharging. It can deliver very high voltage and can be left on trickle or float charge for prolonged periods. This product is also the world's most recycled product, with a wide range of sizes and capacities available in the market. The battery has disadvantages, one of them being that they are typically heavy and bulky. The battery could also overheat during charging, and it is not really suitable for fast charging. The battery will usually last longer with partial discharges. Because our sensor node will be draining the batteries to a low level, lead acid would be a bad choice. Also they are heavy and low energy density.

Batteries supply power to the sensor node. It was important to choose the battery type since it can affect the design of sensor node. Since we decided to use a rechargeable battery, having a battery protection circuit will avoid overcharge and over discharge problems. A power voltage regulator and others components were also added to the sensor nodes to get the voltages that are needed. There are many types of batteries that can be used. Batteries can be divided into non-rechargeable and rechargeable categories. They also can be classified according to electrochemical material used for electrode such as NiCd, NiMh, Lithium-Ion, and Lithium-Iron-Phosphate. For this project, for the sensor nodes, since space is limited and the battery life is very important, a battery with a high energy density is needed. The Lithium-Ion and Lithium-Iron-Phosphate have high energy densities, also they are light weight. To extend the lifetime of the sensor node, we could implement ways of recharging the battery by extracting energy from the environment such as by solar power or RF. For the RF, we could use a Radiant energy circuit made by Tesla to run anything that only needs a volt. Or just include a small solar panel on the module to charge whenever there is light. This would prolong the li-po (lithium polymer) as long as possible.

The table below, (table 3.6.2) shows a comparison between different battery types and their respective specifications.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Specification** | **Ni-Cad** | **Ni-MH** | **Li-ion** | **Rechargeable Alkaline** | **Lead Acid** |
| **Charge/Discharge life cycles** | 500-700 | 300-400 | 300-1000 | 100+ | 500-800 |
| **Memory Effect** | High effect | Little effect | No effect |  |  |
| **Operating Temperature** | -22 to +140°F | -4 to +122°F | +14 to +140°F | -20º F to 130º F | -85º F to 149º F |
| **Charging Temperature** | +32 to 113°F | +32 to +113°F | +32 to +104°F |  |  |
| **Storage Temperature** | +32 to +86°F | +32 to +86°F | +32 to +86°F |  |  |
| **High-Temperature Susceptibility** | Some permanent loss of capacity above 140°F | Greater permanent loss  of capacity above  140°F | Greater permanent loss  of capacity above  140°F |  |  |
| **Self-Discharge** | 20% loss of charge/month at 77°F | 30% loss of  charge/month at 77°F | Self-discharge  3% loss of  charge/month at 77°F | (5/12)% loss per month |  |
| **Chemical Reaction (Charged = Discharged)** | 2NiOOH + Cd + 2H2O = 2Ni(OH)2 +  Cd(OH)2 | NiOOH + (M)H = Ni(OH)2 + (M)  Where (M) is the specific metal alloy | Varies, depends on chemistry. | 2Zn + 3MnO2  KOH  --> 2ZnO + MN3O4 | PbO2 + SO4 + 4H = PbSO4 + 2H2O |
| **Initial Voltage** | 1.2 volts | 1.2 volts | 3.6 volts (3.7 li-poly) | 1.5 volts or 9.0 volts. | 2.1 volts |
| **Discharge Rate** | curve is flat; it will  deliver 1.25 volts throughout most of the discharge time | Flat | Flat | Sloping. Products with a high current drain  are particularly hard on batteries; the higher  the current drain is, the steeper the discharge  slope will be. | Flat |
| **Energy density(Wh/kg)** | 40-60 | 30-80 | 150-250  (Lithium polymer 130-200) |  | 30-40 |
| **Power** | 150 | 250-1000 | 1800  (Lithium polymer 3000+) |  | 180 |

Table 3.6.2 A comparison between battery types.

3.6.3 Solar

If time permitted, we would have liked to extend the sensor node's life time with alternative power sources. Being in the sunshine state, solar power is an obvious root to go. Solar cells capture an average around 18% of the suns energy and convert it into electricity that can be used to do various things such as recharge a battery. There are two main types of solar cells, monocrystalline and polycrystalline both having their respective advantages and disadvantages. Monocrystalline solar cells are grown from single crystal ingot, which is what gives them their clean appearance. The polycrystalline solar cells are sliced from multiple crystals which is why they have many different color flakes in them. They are cheaper to produce but they are around 2% less efficient than the monocrystalline cells. For our project, for a 20% cost reduction, getting a polycrystalline cell rather than monocrystalline cell would be good. The 2% difference is not that important since the sensor node is designed to be super power efficient anyways. We were not able to incorporate this into Eco-Sense.

3.6.4 Hydro-power

Likewise, if time permitted, we would have like to not ever recharge the battery on the node with the solenoid. If this area is shaded, then solar is not a good option. However, next to the water valve is a wonderful source of energy, flowing water. When the system is watering the lawn, it would be possible to capture some of the energy from the flowing water and use it to charge the battery or capacitors in the node. This would greatly extend the life of the unit without having a human recharge the battery. And unlike solar, it would not be as dependent on the weather conditions. The most simple and applicable design would be having an elbow join on the PVC pipe. Then at the bend, have a shaft connecting to a propeller blade inside the elbow join. The water will flow past the propeller spinning the shaft which connects to a small motor to generate electricity. We would also have to design a battery charger circuit to convert this power into something usable for the system. We were not able to incorporate this into Eco-Sense.

Saving Power: To maximize the sensor network life time (without human interaction), it was most desirable to minimize power consumption (maximize battery life) and still have an appropriate coverage area based on our needed requirements. Using various power management and scheduling algorithms, it was possible to have the sensor nodes only awake for transmitting the sensor data and sleep for the rest of the time. For example, a sensor node is active 100% of the time and on battery power lasts for only one day, but now it was set to be active 1% of the time and sleep the rest of the time. Now we get months of run time. Maximizing battery life time requires a well-designed structure to allow for low power operation in all aspects of the network, from hardware to software and network protocols. An example for hardware selection would be using the bare minimum required to obtain sensor data, transmit it when needed, and sleep. Having components such as LEDs or USB components just use up power that isn't needed for everyday operation and should be removed. (Or be modular so it can be added when needed.)

3.6.5 Microcontroller Power

Having low power consumption is of great importance to maintain the system over a long period of time. For the microcontrollers, to save power they have different power modes. The MSP430 has six diﬀerent power modes, ranging from fully active, to not clocking the core, to keeping the digital oscillator running to generate the clock but disabling the loop control to save power to fully powered down (with peripherals separately enabled or disabled). It also has a digital oscillator which allows the wakeup time to be around 6µs. The ATMega128 also has a variety of power down modes. The CPU clock can be stopped, leaving the peripheral clocks running. The CPU oscillator can be kept running, which allows it to restart in under one microsecond, or to be powered off completely. There is a powered down mode that stops all peripherals but keeps the asynchronous timer and oscillator, which runs oﬀ an external 32kHz crystal. The asynchronous timer allows for the controller to be woken up (into active mode). The ATMega328 has 6 different sleep modes. Each with different clocks (core and peripherals) disabled and some with disabled memory access. The PIC18 has two power-saving mode. One that runes the peripherals but not the core, the other mode powers down both (except the asynchronous timer to run). The PIC16 has one power-saving mode. This allows the asynchronous timer to run with everything else powered oﬀ.

Other components: Different components including the sensors, microcontroller, and transmitter require different voltages to run properly. Most sensors run at 3.3 volts or 5 volts, because we used lithium polymer battery packs, then each cell is 3.7 volts which wouldn't be enough voltage for the 5 volt components. 2 Cells had to be used providing 7.2 volts, more than what is needed for both but still too high to use. It’s also important to note that because a rechargeable battery is used, the rated voltage is the nominal voltage. When it is freshly charged, the voltage will be higher. For example, a Li-ion will be 4.2volts per cell and not 3.7 volts. This is where DC-DC converters come in. Using potentiometers they can lower the voltage to meet the requirements of each component.

3.7 Communication and Network Technologies

Wireless sensors arranged in a small radio network allow us to remotely monitor the physical world for any set of given parameters of interest. As a whole, these sensor networks act in a similar manner in how they obtain information and send it to a source for interpretation to make use of the data. For our project, we have a small example of a wireless sensor network to monitor plants/lawn and make decisions on when to water or add nutrients based on the sensor data to ensure proper growth and plant maintenance.

A sensor node can be thought of as four basic components.

1.) Microcontroller or microprocessor.

2.) Short range RF communication for the network communication.

3.) A system of sensors to gather data from the physical word.

4.) Power subsystem including the battery, DC-DC convertors, and any charging modules such as solar power.

For each sensor node, it is important to consider the following requirements:

The nodes must be energy efficient. They have a limited power source that may not have a way to recharge itself. So their battery life determines how long the node will last without human interaction. For a large scale sensor network system, it would be impractical to have to recharge all the nodes over a short period of time (weekly). Therefore being energy efficient is the primary metric in this analysis.

As this is self-funded, the next important metric is for this to be a low-cost module for the project. If this project was scaled larger, there may be hundreds of these nodes and the cost would grow rapidly. As there are multiple sensor nodes that can be a significant distance from the central HUB subsystem node, having wireless capabilities is a must. These sensor nodes can be placed in a variety of environments and having wires running everywhere just to connect all of the nodes would be very costly (and unsightly).

As a whole, the system must be distributed processing and sensing. Each sensor node must be able to collect data from multiple sensors, process the data as needed, then relay the information to the central HUB subsystem node for further processing. But for larger wireless sensor network systems, the sensor nodes may be out of range of the central HUB subsystem node, but in range of other nodes. So the system should be able to do a multi-hop transmissions. Where the information is sent a short distance multiple times to reach the central HUB subsystem node. Otherwise each sensor node would require a higher cost and a higher powered transmitter and as stated before, this must be and energy and cost efficient system.

For our project, we need to be able to relay information to and from our sensor nodes to the central HUB subsystem. Each sensor node must be able to communicate to other devices and to a central HUB subsystem using a wireless communication device. The only feasible communication method is by using radio frequency to transmit the information. Line of sight methods such as IR are impracticable as buildings and walls would prevent proper transmission of the data from sensor node to central HUB subsystem. Radio frequency communication provides a moderate transmission range and high data rates, and an acceptable error rates at a reasonable power usage. And additionally, it does not require line of sight between sender and receiver like some wireless devices. The industrial, scientific and medical (ISM) radio bands are used for industrial, scientific and medical purposes other than communications. The 915 MHz, 2.4 GHz, and 5 GHz bands are widely used for sensor networks because they do not require an expensive license to legally broadcast on those frequencies. Therefore, whichever wireless device we use; it must be in the ISM band frequencies.

In designing our own wireless sensor network, the important factors in choosing a proper wireless technology can be broken down into: considering the range (of transmission), the transmission power, which frequencies/wavelengths are used, how sensitive to noise the receiver is, and the type of antenna used. Strictly from a physics standpoint, lower frequencies extend further than higher ones can. Wireless signal's power decays exponentially, similar to a graph of 1/d2where d is the distance from the transmitter to receiver. For example, a 915 MHz will travel farther than a 2.4 GHz signal. Using lower data rates also extend the range and signal quality (which is why your WIFI and cellular signal changes speeds when the signal strength is poor.) Also, the slower data rates are less prone to interference from other devices. If possible, using a lower data rate will give the best results. Although this depends on the application, and for us, we do not need high definition video streaming quality data rates. To calculate the loss for a typical signal, you first assume the furthest distance the signal will need to travel. For this example, assume the distance "d" is range of one mile, "f" is the frequency in MHz.

For a 915 MHz signal, the loss would be. 37 + 20 log(915) + 20 log(1) = 96.23 dB.

For a 2.4 GHz (2400 MHz) signal, the loss would be. 37 + 20 log(2400) + 20 log(1) = 104.6 dB.

This formula assumes there are no obstacles obstructing the signal such as trees, walls, buildings from transmitter to receiver that would interfere with the signal. The next calculation is to ensure a good quality link connection.

Receive sensitivity (minimum) = transmit power (dBmW) + transmit antenna gain (dB) + receive antenna gain (dB) – path loss (dB) – fade margin (dB).

Fade margin is a best guess on how much you think the signal will fade. It should not be less than 5 dB, but could be as high as 40 dB to ensure a 100% link reliability. Any other losses like the transmission line loss or obstructions should be subtracted as well. The number that results from this should be larger than what the receiver's sensitivity is. Usually, receiver sensitivities range from a low -70 dBmW to -130 dBmW+. Assume the transmission power is 2 dBmW, 0 dB antenna gain, and a 96.23 dB loss for the 915 MHz signal, and a 104.6 dB dB loss for the 2.4 GHz signal. Also, set the fade margin to around 10 dB. The resulting link characteristic should be:

For the 915 MHz: dBmW

For the 2.4 GHz: dBmW

To obtain a reliable link, the receiver sensitivity must be greater than -104.23 dBmW for the 915 MHz signal and more than -112.6 dBmW for the 2.4 GHz signal.

Our system uses a wireless communication device such as those used in wireless devices including cellular phones and laptop. However, unlike your computer which are transmitting data frequently, the (sensor) nodes will only be on for very short durations to send a small amount of information then go back to sleep. Also important is the power usage, and (when active) the transmitter radio is one of the highest power consumers in a sensor node. Other factors that can affect the power usage for the radio include the data rate, transmit power, and modulation of the signal used. Most transmitter radios have four separate modes of operation: transmit, receive, idle, and sleep. It is important that the radio is put to sleep when it is not needed because many wireless radios operating in idle mode results in high power consumption, sometimes equal to receive mode. Energy efficiency is an important metric to consider, however the device power cannot be so low that we save power but have barely any coverage range. So a compromise on power usage and transmission range is needed. Also, it would be nice to have both a transmitter and a receiver, which would allow small code changes to be sent to the sensor node such as changing the interval time for logging data or the interval time for transmitting data. But in general, for the majority of the sensor node's life time, they are transmitting information.

For scaling our project, if the sensor nodes are out of range of the central HUB subsystem node, it would be desirable to have the functionality to have intermediate nodes to relay the information from several sensor nodes to a relay station then to the central HUB subsystem. These intermediate nodes would relay the information from the sensor nodes to the central HUB subsystem. This means that they would always have to be on, listening for any transmissions like the central HUB subsystem would be. Using these nodes in a mesh like network would allow us to obtain a larger range from central HUB subsystem to sensor nodes. Otherwise we would have to increase the transmission power on the sensor nodes which would consume our limited power that is there.

In summary, whichever wireless technology we choose must have the following attributes: It must be very low power, have a long range of transmission, capable of transmitting and receiving data, and be able to go to sleep (low power mode), and be woken up with a microcontroller. Specific technologies for radio frequency transmission that we considered include WIFI, Bluetooth, pure RF, and Zigbee.

3.7.1 Protocols

3.7.1.1 RF

It is an old technology, being used for around 60 years. There are unlimited frequency bands this device could use, but for the majority of them, you would have to acquire a license, which is expensive, or just use the free unlicensed frequency bands. As data is being transmitted, it is not changed in any way. A stack protocol is not used like in other wireless communications. This makes the latency low and the need of fewer electronic components on the breakout board. However this means that only one node can talk to another at a time. Otherwise the hidden node problem can occur. For example, node B is in range of A and C, but A and C are out of range of each other. If A transmits to B, then C wants to transmit, it does not know that A is transmitting. C will transmit its data causing interference to B, then A's transmission will fail as well. Another problem that can occur is the exposed node problem. If for example, nodes A,B, C, and D are in range of each other and A is transmitting to B. Then C wants to transmit to D but thinks D may be getting a transmission since it can listen for any transmissions in the area, which would be A sending data to B. This would make C not transmit anything do D. Both of these would be large problems with our system. Because this system cannot take into account all of the obstructions that the environment it is in. If there were several sensor nodes trying to communicate to the central HUB subsystem or each other, data would be lost, or delayed in being sent. This would waste time in retransmitting the data and the more transmissions and time the node is on, the more power is waited.

Also for our project, radio frequency uses low power and has a good range of 150m, which is what we need, however, they have slow data rates means the node would have to be on for a longer period of time, and only one node can transmit at a time. Which would be hard to coordinate between the nodes when to send and when not to. The cost is very low around $10 for the module. Overall, it is a bad choice.

3.7.1.2 WIFI

WIFI is one of the most popular wireless communication protocols. There are little magical boxes in many modern homes known as a Wi-Fi router to transmit data to all devices connected to it. WIFI operates on the unlicensed frequency bands of 2.4, 5, and now 60 GHz. It uses the IEEE 802.11 protocol, with the 802.11b being the slowest at 11 Mbit/s and the fastest (although not publically available yet) at 802.11ad which gives around 7 Gbit/s. It has very high data rates and can have multiple nodes transmitting/receiving at the same time, and a good range. However it requires a large amount of power, and for the amount of data we need to send, 10mbps is much more than what is needed. The cost for just the WIFI module is moderately expensive at around $50. But if we have several nodes using this, it would be expensive quickly.

3.7.1.3 Bluetooth

Bluetooth operates at a public frequency band. It uses the ISM 2.4🡪2.48 GHz unlicensed band. It was designed for short range communication up to 10 meters. Bluetooth uses a personal area network, but its security is weak only using a 4 digit encryption. The maximum theoretical date rate is 3 Mbit/s using 8DPSK modulation, but in practice the data rate is around 1 Mbit/s. It uses a master to slave communication design which would not be good for this project since we need to have multiple nodes connect to each other. Bluetooth for our system has a reasonable data rate but the range is not that far at 10m. For the power, although less than WIFI, it still uses quite a bit of power to run. But it is a popular technology to interface to computers and cell phones so this is still viable for the controller hub to communicate to a phone or computer. But as for the nodes, they would never need to communicate to a computer or phone on a regular basis. The cost for just the WIFI module is moderately expensive at around $30.

3.7.1.4 Xbee/Zigbee

Xbee is the module using Zigbee protocol. It is relatively new on the market. It was designed for low power digital radio wireless communication. It allows many devices to cross transmit data in the network at the same time. The whole network has a large amount of data, but the throughput of a single device transmitting is low, around 250 Kbit/s. They are able to transmit outdoors 100m to 1600m (for the pro version) which is great. And the transmit power requirement is 1mW and 63mW (for the pro). The bandwidth is around 250kbps which is enough to transmit/receive our data from the nodes to the controller hub (main node). Also, Xbee gives us 255 nodes to use which is more than needed and will allow for scalability of the project if needed. Xbee modules cost around $25, which isn't as cheap as the RF ones, but can do significantly more things than the RF modules can. Also since they use such low power, a 3.6V 600mA Lithium battery may last 6 months to power them up.

The table below, (table 3.7.1.4) shows that the Xbee (which uses Zigbee) and Zigbee devices have a very low power usage, a long range, but a low data rate. This is fine since we are not transmitting large amounts of data such as a data stream. All of the data that is transmitted are in short bursts, spread out over long periods of time. Xbee/Zigbee gives both the benefit of low power and a large range with an acceptable bandwidth making it a good choice among the choices. Bluetooth could be used in our central HUB subsystem as to provide an external user interface on an Android device or computer. This would show the collected data (graphically) and allow the user to make changes to the scheduling rather than using the interface of the central HUB subsystem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Xbee | Xbee pro | Zigbee  802.15.4 | Bluetooth  802.15.1 | WIFI b/g  802.11b/g |
| Battery Days |  |  | 100-1000 | 1-7 | .1-5 |
| Power transmitting | 1mW | 63mW | 30 mA | 40 mA | 400 mA |
| Power sleeping |  |  | .3 mA | .2 mA | 20 mA |
| Nodes/Network |  |  | 255/65K+ | 7 | 30 |
| Bandwidth | 250 kbps | 250 kbps | 20-250 kbps | 720 kbps | 11-54 mbps |
| Range (m) | 30(indoor)  90(outdoor) | 90(indoor)  1600(outdoor) | 75+ | 10+ | 100 |
| Key attribute | Low cost, Low power | Low power | Low power, reliable | convenience | speed |

Table 3.7.1.4: wireless module comparison

Because we have a limited time to develop the project, we will try to the easiest and cheapest solution in order to satisfy the needs of the system. By using a Xbee/Zigbee device will fit out needs and after comparing the prices online, each module was less expensive than the WIFI for Bluetooth modules. Pure RF was the cheapest, but it would not be useful in our project.

3.7.2 Zigbee

The Zigbee device such as Xbee covers the Physical Layer and Data Link Layer, if we need it, the upper layers would be developed by us. According to the Zigbee website, it is able to implement the following: star, mesh, and tree topologies. All Zigbee networks need a coordinator (which is a central HUB subsystem that everything is sent to) and it routes the data if needed. The coordinator node (which is our base node) has different names depending on the topology used in the network. Using Zigbee will also give us the flexibility to use different wireless topologies depending on how large we intend to make our sensor network.

Figure 3.7.2.1: star topology

3.7.2.1 Star topology

There are two kinds of devices, the **central node** (central HUB subsystem) and the **sensor nodes**. All the nodes are connected to the central node and all the communication was through it. This kind of network represents a risky scenario and the problem is that if the central node goes down, the network will stop working. All the data will go to the central node.

Figure 3.7.2.1: Tree topology

3.7.2.2 Tree topology

This topology has a main node called **top node** (central HUB subsystem) and below it there are branches which connect the different nodes. The **sensor nodes** are at the farthest points. When a node wants to send a packet to another node, the packet will go up until the root node that connects both end nodes. This approach could be a good idea if we want to connect many devices in a moderately large area all to the **relay nodes** (they just relay the information) and then the relay nodes to the top node (central HUB subsystem). But ithas the same problem that the star topology, if the central node fails, some areas of the houses are going to be in troubles.

Figure 3.7.2.3: Mesh topology

3.7.2.3 Mesh topology

This is the most flexible topology that our group can implement using Zigbee technology. In this kind of network all the devices can communicate with each other’s. There is a **central node** (central HUB subsystem), and **relay nodes** around it connecting to **the sensor nodes**. The network will and the route automatically to deliver the message correctly and the main advantage of this kind of network is that they are decentralized, if some node has problems, it doesn't mean that the whole network will go down like it happens when we use the other topologies. The only problem with this network topology is the cost. It would require many different stations, but if we were monitoring all of UCF, then it would be the most practical one.

Our project is a small scale wireless sensor network. A network topology stitch as a star network would best fit because we have one central HUB subsystem node as the controller. We can easily add any number of end nodes as needed without having to worry about packet forwarding like the other topologies have. Also this simple layout would be the least expensive while still having the flexibility to move, add, or remove nodes as we see fit.

3.8 System Housing

Eco-Sense will contain three different boards and designs that need to be housed accordingly. The three boards that are referred to are the central hub, sensor hub, and the solenoid subsystem. The main difference between the housing of the three boards will include water resistant materials and design (sensor hub), aesthetic design (central hub), and efficient design (solenoid subsystem). Due to wide availability, our project will research prebuilt housing and also custom built housing.

3.8.1 Premade Housing

Premade housing benefits this project by saving effort and time for construction. There are plenty of different designs to be looked into but there’s a factor that does make this decision difficult which is the fact that these designs are prebuilt, meaning the system would have to fit into what’s given and offer a possible chance of wasted space or not enough room. Besides the spacing problem, our group decided it was worth looking into to see what was readily available for housing. Premade housing for the central hub is shown in table 3.8.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Housing | PACTEC Box | RP6472 Box | MFG Box | SK4094 Box |
| External Dimensions (in.) | 5.5 x 3.6 x 1.1 | 11.12 x 7.2 x 3.4 | 5.07 x 2.53 x 1.75 | 4.125 x 2.75 x 1.562 |
| Internal Dimensions (in.) | 5 x 3.1 x 1.1 | 10.8 x 7 x 3.2 | 4.76 x 2.25 x 1.57 | 4 x 2.5 x 1.51 |
| Material | Plastic | Metal | Plastic | Plastic w/ Aluminum Lid |
| Shape | Rectangle | Rectangle | Rectangle | Rectangle |
| Price | <$5 | <$10 | <$10 | <$5 |
| Color | Off-White | Off-White | Black | Black |

Table 3.8.1: Central hub premade housing options

All the premade housing that was researched turned out to be rectangular in shape and less than $10. After viewing these options, it would be very cheap to buy a premade housing box for a foundation then modify the inside to the system’s needs. Most of these housing models were found at Sky Craft. The RP6472 box was ruled out almost immediately due to its large size and clunky design. The PACTEC box is considered the best fit for this board due to its small and thin design. This board is mounted a wall or even made mobile and this box offers good mobility and an easy to use material. The only precautions that need to be made are the PCB and component sizes so they will fit inside the box.

The sensor hub will require more spacing than the central hub. The sensor hub must be modifiable to manage an air sealed area for the electronics and an exposed area for the sensors. The box should be a lighter color to reflect light and should be made of an environmentally friendly material. Drilling is required inside the box so the material of choice for this board is plastic. The premade housing for the sensor hub is shown in table 3.8.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Housing | RP0861 Box | Reglocard Enclosure | Wisher Project Box |
| External Dimensions (in.) | 8.85 x 4.44 x 2.48 | 8.375 x 7 x 4 | 7.9 x 6.3 x 2.6 |
| Internal Dimensions (in.) | 8 x 4 x 2.1 | 8 x 6.75 x 3.8 | 7.5 x 6 x 2.25 |
| Material | Plastic | Plastic | Plastic |
| Shape | Rectangle | Rectangle (2) | Rectangle |
| Price | <$10 | <$10 | <$20 |
| Color | Black | Gray | Black |
| Compartment Included? | No | Yes | No |

Table 3.8.2: Sensor hub premade housing options

The sensor hub housing options were more diverse than the central hub options. The Wisher project box will most likely be dismissed first due to high cost compared to the other two and smaller dimensions. The Reglocard enclosure box is considered the best fit for this board due its large dimensions, lighter color, and a small compartment included. The idea of the sensor hub housing was the have a sealed portion for circuitry to protect against water buildup, and have wires through the sealed area that lead outside for the respective sensors to read data and transmit through the wires. It is unknown if the compartment is sealed from the rest of the box, but in the event that it is not, we would be able to make a makeshift barrier and drill holes for the wires to then be patched up and sealed.

The solenoid subsystem housing should be compact and lightweight so it may be stored near the spigot of the home. It should be made of an easy to drill material so the solenoid threads can feed through the sides of the box. The spigot will connect to one end while the hose will connect to the other side of the box. The box should be lightweight to avoid drag on the spigot. The premade housing for the solenoid subsystem is shown in table 3.8.3.

|  |  |  |  |
| --- | --- | --- | --- |
| Housing | RP0860 Box | 270-1805 Box | 270-1806 Box |
| External Dimensions (in.) | 6.14 x 2.64 x 1.72 | 6 x 3 x 2 | 6 x 4 x 2 |
| Internal Dimensions (in.) | 4.82 x 2.36 x 1.55 | 5.75 x 2.9 x 1.9 | 5.75 x 3.9 x 1.9 |
| Material | Plastic | Plastic | Plastic |
| Shape | Rectangle | Rectangle | Rectangle |
| Price | <$5 | <$5 | <$5 |
| Color | Black | Black | Black |
| Hole Cutouts? | No | No | No |

Table 3.8.3: Solenoid subsystem premade housing options

The solenoid subsystem housing options were pretty standard compared to the options for the sensor hub. The boxes were pretty much the same except for slight dimension differences. Due to this, our group decided to go with the 270-1805 box since it was in-between the other two in dimensions. If problems occur for sizing, we can always downgrade or upgrade to a larger box.

3.8.2 Custom Created Housing

Another option for Eco-Sense is to create our own housing box for each board. This approach will require much more labor and time but will offer the benefits of shape and sizing limitations. Each box could have its own compartment layout and offer an efficient design when compared to buying a premade housing. The important features for custom housing that needs to be looked at include the dimensional cost, the material the housing is made out of, the shape, whether or not the box is sealable, and if it can withstand the outdoor elements.

The material for the custom houses will vary depending on which board is being used. The central hub should have an aesthetic design since the user interacts with it. It should be thin and lightweight, have mountable points, and also the option to carry it if needed. The outer sides of the box should not be completely transparent so the user won’t have to look at the electrical work behind the scenes. A plastic material is ideal for the box to keep it lightweight. The sensor hub housing should be well shaped due its sealed compartment requirement. The material should be easy to drill to ensure easy construction of the sensor inputs. The color of the material should be a natural or light color to reflect light which will keep the system cooler. A plastic material is ideal for the box to avoid electrical conduction from nature and to avoid rust or corrosion to the outside portion. The solenoid subsystem housing should be small and compact, just wide enough to fit the solenoid valve in the cutouts. This housing will most likely be near the spigot, so waterproof material should be used in case of a sudden burst or leak from the spigot.

Tap Plastics offers polycarbonate sheets that are abrasion resistant. This material offers high temperature resistance, high impact resistance, resistance to weather, and easy construction. The material is quite costly in the long run at about $0.14 per square inch. Assuming an average box is 8” x 5” x 2”, just the outside of the box is $11.67, not including dividers to section off areas. A decision will need to be made whether this extra cost is worth the customizability or if a premade box will work just fine. Tap Plastics also offers plastic canister boxes that could fit in a module design in each box. Prices range from $3.00 to $6.50 per canister of different sizes. This might be a better choice to make the system modular, it would make testing and maintenance much easier. They also offer molded boxes that have a snap shut lid. This feature can add on to the modular design and make components extremely easy to get at instead of sealing up parts and having to tear them down for maintenance. The boxes cost around $1.50 - $3.00 each, making this an attractive choice for the sealable compartment of our housing. With all the available resources for housing, our group may use various sources and combinations to produce a housing structure suited for Eco-Sense.

After reviewing all the available options for housing, this project will benefit the most with a combination of premade outer boxes with a module design inside. The cost for a premade box is much lower and it guarantees a water sealed structure and uniform dimensions. Instead of buying plastic sheets and cutting them ourselves, it would be more beneficial to buy modular boxes and remove lids if necessary and have them mounted inside the box. This approach offers some components to be exposed to the elements while some circuitry is sealed off to protect against water and dirt. Having modules inside the box will help with organization and maintenance during testing and allow us to organize incase more components are added to the final design.

3.9 Optional Additional Features

3.9.1 Mobile App

Why Android?: Nearly everyone has a smart phone these days. If time allows for it, we would like to have the central HUB subsystem and mobile devices interact. To interact with our system, the application would have to be constructed from graphical user interfaces (GUIs), and be able to interact with the hardware on the central HUB subsystem. Using an Android based app rather than an IOS application would make more sense because all members in the group own an Android smart phone. Also one member already has experience in developing Android applications and two members are already familiar with traditional Java programming. This would also allow everyone in our group to have access to the system and be able to test it independently.

Android uses a java based programming language with many libraries to interact with the operating system. The operating system is Linux based, running on an ARM CPU. The Android development interface is fully integrated into the Eclipse suite making the transition from java programming to the Android's java plug-ins. Android was also chosen due to the availability of reference material and the open source development platform environment. Google, unlike Apple has provided numerous pages and tutorials on how to develop Android applications. The Android developers' website has reference documents on the separate classes, types, and structures that make learning easier. The community on android websites also have many other resources for developers as well as tips for designing the user interface. They provide an editor in Eclipse to create your own layouts. The Layout Editor will allow us to build a friendly Android interface with all the underlying functionality of the application. Using this editor, you have the ability to visually create the graphical interface which that you wish to use. This layout editor is tightly associated with the XML that it generates within your code. The layout can also be modified through other third party layout generators and inserted into the XML layout files. This will give us more flexibility in designing the user interface to fit out needs. The Eclipse compiler also has an Android Emulator that simulates the phone, which can be used for testing. Google has provided this development software, along with any of the emulated operating systems at no change to anyone. This will prevent wasting money on this development portion of the project, since it is just an optional feature if time permits to develop and test it.

Android Application: The smart phone would be able to request data from the central HUB subsystem which stores all of the data acquired from the sensor nodes. This would be stored in the Android database and graphically shown on the screen the changes in readings from the sensors. Graphs for each sensor would be available to the user on a simple graphical user interface. For the functionality part of the Android mobile app, settings on the central HUB subsystem would be able to be modified. The modifiable settings would include changing the schedule of when to water and any parameters that we define for calibrating the sensors. The user would select when to water their lawn and plants, and based off of the parameters set on the central HUB subsystem, it would signal the solenoid to water the lawn and plants. The parameters would include things to predict imminent rain or if it is currently raining. Sensors such as the temperature, barometric pressure, and the humidity sensors obtaining a certain range would be used to predict the near future weather. For example, it is known that when the barometric pressure drops, it tends to be come cloudy and possibly rain. When this happens, the temperature usually drops, and the humidity skyrockets up. These together are a good indication that mother nature will water the grass for us, for free. If there are any scheduled lawn watering, and then it is delayed. As with the central HUB subsystem, the smart phone application would have the ability to override everything, and manually water the grass. This would be useful to a user who just laid fresh sod and needs to water the grass continuously.

3.9.2 SD Storage

Unlike your computer hard drive, our chosen microcontroller has less internal memory than those 3.5 inch floppies that no one uses anymore. Storing large amounts of data collected from the sensor nodes plus the program, the internal flash memory is used up quickly. If we wanted to data log everything, then external storage device is the solution. Using an external SD card will allow us to have gigabytes of storage. And because we are using an ATMega328 microcontroller which is popular in the Arduino development boards, and the beauty of this is that in the Arduino libraries, they have a FAT16/FAT32 library. So, we can use a large SD Card without any problem provided that the AVR you use has enough RAM &programming memory. To save space, we can use a micro SD rather than the regular SD.SD Cards are also using SPI protocol for the communication, so the SD pin connections are the VCC (3.3v), GND, MOSI, MISO, SCK, and CS. When connecting the two devices, we can use the ATMEGA328’s built in 10bit ADC with the Vref=3.3V. The figure below shows the pin connections to the ATMega328. The SD card will connect to the microcontroller through the digital and clock pins.

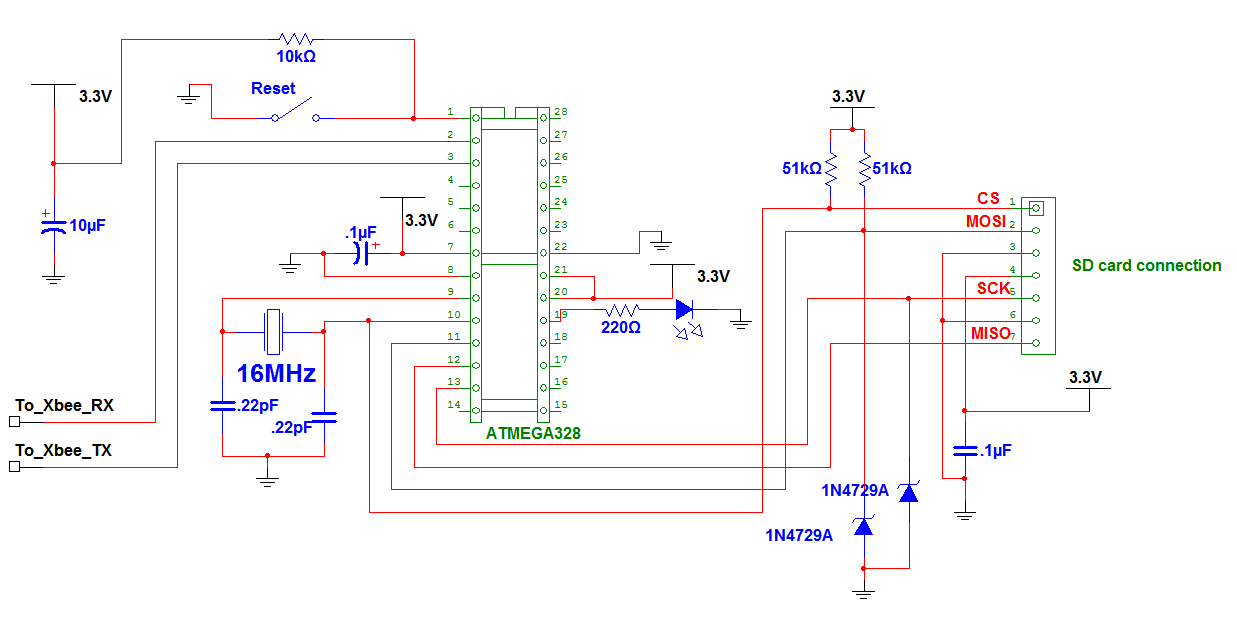


Figure 3.9.2 :SD schematic

The benefits of having the ability to store more data is so that we can use it to make better weather predictions. We can determine what time of day it is more likely to rain based off of the stored sensor data. Also, if we use an Android application, the external data can be read by the smart phone and shown as a graph to the user. The graph would show a change of the data for each sensor over a set period of time. This extra storage can be used for storing a program log of any error messages that the microcontroller has. For example, if the central HUB subsystem tells the solenoid to turn the water on, but it doesn't; this would be an error. Errors such as these would be stored on the SD card for the user (and programmer) to view them at a later time, to fix the problems. Other messages such as actions (a system log) that the microcontroller does that are not errors can be stored. Such actions as a "Scheduled to water grass, but sensors predict rain soon", "While watering the grass at (timestamp), it started to rain. Turned off sprinkler.", or "Finished watering the grass at (timestamp), but it rained (# of) minutes after watering the lawn". Such messages would be useful in fine tuning the data ranges that are used to predict rain in the near future. This is so the system will not even attempt to water the grass and stop prematurely (due to rain).

4.0 Project Hardware and Software Overview

4.1 Project Block Diagrams

4.1.1 Hardware Block Diagrams

Due to its size and separation distance, Eco-Sense’s hardware block diagrams are split into its three subsystems, sensor, central HUB, and Solenoid. The hardware block diagrams are used to represent the physical connections between each component and how the information, or power, flows between them. The blue lines represent the flow of data between components while the think black lines represent where and how power is distributed. Note that these diagrams are subject to change in the final design.

The sensor subsystem has the most individual components of any other subsystem in Eco-Sense. Figure 4.1.1.1 shows the hardware block diagram for the sensor subsystem. The main focus of this subsystem is to take data obtained by the various sensors and deliver it to the central HUB, when given the command to detect for rain, and send any detection of rain while watering immediately to the central HUB.

Barometric Pressure Sensor

Ground Moisture  
Sensor

Transmitter  
Receiver

Microcontroller

Power Source

Ground Temp.  
Sensor

Ambient Temp.  
Sensor

Rain Detection  
Sensor

Humidity  
Sensor

Sensor Subsystem Hardware Block Diagram  
Figure 4.1.1.1

The central HUB subsystem is the brains of Eco-Sense; the process for the central HUB is shown in figure 4.1.1.2. It is here that all the data collected from the sensor is processed and stored. All the calculations, on weather prediction and if to water, and all the commands are made and sent from here. The central HUB subsystem contains a LCD display system to show data to the user and allows for different data to be called with the user interface.

Power Source

Microcontroller

LCD Display

User Interface

Transmitter  
Receiver

Central HUB subsystem hardware block diagram  
Figure 4.1.1.2

The last of the subsystems is the solenoid subsystem. The solenoid subsystems purpose is to control the open and close feature of the solenoid attached to the water supply for Eco-Sense. Figure 4.1.1.3 shows the general layout of this system. When the microcontroller receives the signal from the central HUB, it opens the solenoid allowing water to flow. When the signal to stop watering is received, or the time-out timer runs out, the solenoid will close the valve and the microcontroller will go into a sleep mode awaiting the next signal to water.

Solenoid

Transmitter  
Receiver

Microcontroller

Power Source

Solenoid subsystem hardware block diagram  
Figure 4.1.1.3

4.1.2 Software Block Diagrams

Eco-Sense will require each board to have its own software diagram to show the processes that will occur on normal operation. The parameters and fine details will not be shown, but the overall process hierarchy and patterns is described for each system below. Note that this diagram is subject to change in the final design.

Sensor Subsystem Software Diagram: The sensor subsystem is responsible for reading sensor data at select intervals and storing it until the central hub requests the information. The sensor subsystem will contain its own timer that is separate from the other systems. The main components this software diagram will take advantage of are the sensors, memory, and the wireless transmission module.

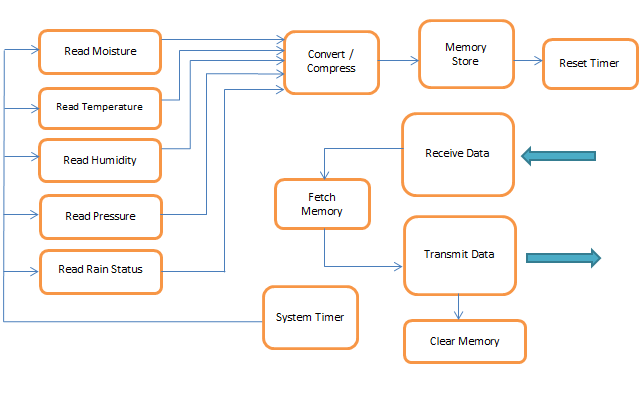


Figure 4.1.2.1: Sensor subsystem software diagram

The sensor subsystem process in Figure 4.1.2.1 starts with the system timer when the device powers on. The system timer is a preset value for the user. The system might be in sleep mode during this time until the system timer reaches its value or a receive data request is found. The system will then process to read the current data from the various sensors. The microcontroller will convert any analog signals to digital and then compress the data. Once the data is compressed, it is stored in memory and the system timer is reset for the next cycle. Once the central hub requests data, the sensor subsystem will fetch its current data in memory and transmit it as a packet to the central hub system. It is worth noting that the sensor timer will expire more frequently than the central hub timer to reduce transmission frequency.

Central HUB Subsystem Software Diagram: The central hub subsystem is responsible for providing the user with current statistics of the Eco-Sense system and offer access control for overriding the system. The main components this software diagram will take advantage of are the input buttons, LCD module, and the wireless transmission module.

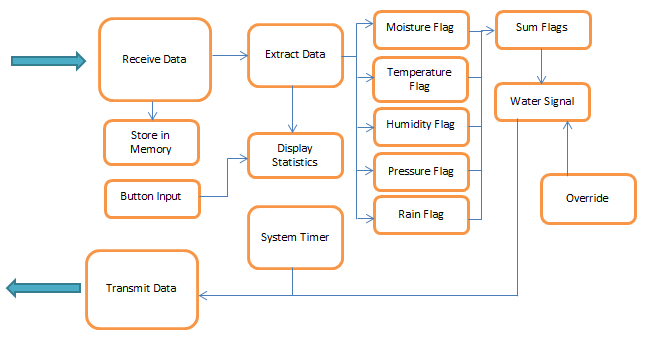


Figure 4.1.2.2: Central HUB Subsystem Software Diagram

The central HUB subsystem process in Figure 4.1.2.2 starts with the system timer when the device powers on. The system timer is a preset value for the user. The system is in sleep mode during the idle times if not transmitting or displaying statistics on the LCD. The system timer will expire and then transmit a request to the sensor subsystem for its data. Once the sensor subsystem sends its data, the central HUB subsystem will receive the data and store it in memory. The data is extracted from compression and is displayed on the LCD module for the user to view at any time until the next cycle, which it will then be refreshed. The extracted data will also be sent to several check conditions, each having a flag. If the condition sets its flag to 1, then that will indicate the specific check was computed and water should be requested. Each flag carries the same weight except the rain flag, which will have a high priority. The rain flag is to check if it is currently raining outside and if the rain threshold is passed, the system should not water unless all flags are set to 1. The flag values are sent to a process called sum flags. Sum flags adds the flag values at determine if the water signal should be set. The water signal is sent as data through the wireless module to the solenoid subsystem. While the flags will determine the water signal, the user is able to take advantage of the override process. This process will allow the user to request that the system should water. A delay is set during this time to allow the user to cancel the request. The user is also able to request the LCD module to turn on at the press of a button. The LCD module will go into sleep mode after a certain idle duration and will awake when either the system receives data or the user requests that the statistics should be shown.

Solenoid Subsystem: The solenoid subsystem is responsible for turning on the solenoid valve for the system. The software diagram for this subsystem is very simple compared to the other systems due to its one function for the project: to turn the valve on and off.

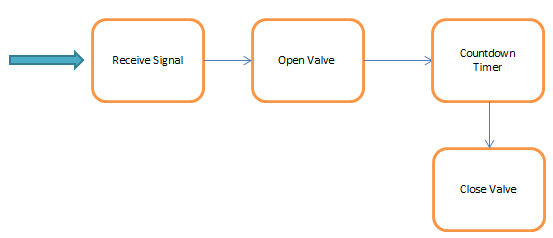


Figure 4.1.2.3: Solenoid subsystem software diagram

The solenoid subsystem process in Figure 4.1.2.3 starts with receiving a signal from the central HUB subsystem. The signal will contain a data bit that is 0 or 1. If the bit is 1, the valve will open and allow water to flow to the sprinkler. If the bit is 0, the system will do nothing and wait for the next signal to be received. When the valve opens, a countdown timer will start. The time is to set the duration the valve remains open. Once the timer expires, the valve will close and the system will wait until another signal is received.

There are two main algorithms that are employed in Eco-Sense. These algorithms are the when to water algorithm and the weather prediction algorithm. These algorithms play an important role in the primary goal of Eco-Sense, to minimize wasted and overwatering. The first algorithm to activate is the when to water algorithm (W2W), this happens right after information from the sensor subsystem is stored at the Central HUB. This algorithm checks to see if watering is needed by using the following process. The first trigger in the process is if it is an allowed watering time, this is done with the user specified watering time (USWT) inputted into the system by the user during setup. If the time is not within the days/times set by the USWT then the algorithm says it is not time for watering. The second trigger is the soil field capacity. The lower the soil field capacity the more needed watering is. The next trigger is temperature and humidity. There are several not allowed temperature and humidity combinations. These are any temperatures over 32˚C, any temperatures under 5˚C, and any temperatures above 25˚C with a humidity level of 50% or less. The last trigger is the dew point. As mentioned in section 3.5.1.5 the dew point can play havoc on the rain detector. So if the current temperature is below the dew point no watering is to be done as Eco-Sense cannot tell it is raining or just dew. Figure 4.1.2.4 shows a process flow diagram of this algorithm.

Within USWT limits?

Temperature above 32˚C

Temperature below dew point?

Soil Field Capacity below 75%?

Temperature above 25˚C?

Humidity below 50%?

Temperature below 5˚C?

Do Not Water

Water

Sensor  
Data

Yes

No

No

Yes

Yes

Yes

Yes

Yes

Yes

No

No

No

No

No

When to Water flow diagram  
Figure 4.1.2.4

The second major algorithm to be used is the weather prediction algorithm. This algorithm is triggered after a positive result from the W2W algorithm is generated. The purpose of this algorithm is to predict the onset of weather storms that can produce rain. To save water, watering is not to be started when a rain storm is eminent. This algorithm is still under construction and is not available at this time.

4.2 Central HUB Subsystem

Eco-Sense is using three different boards to make up the irrigation system. The Central HUB subsystem’s responsibilities are to provide a custom built user interface and to bridge the sensor subsystem and solenoid subsystem. The user interface will include an LCD module and button inputs. Designing the central HUB subsystem will take into account the design of the sensor subsystem and the solenoid subsystem since they will need to communicate to each other. Hardware and software are used for the central HUB subsystem in a way that makes it easy to use for the user, unlike the other two subsystems. Software is written with the arduino boot loader burned onto an Atmega 328 DIP chip. The 328 offers 32kB of flash memory while the boot loader will take up 2kB of that memory, offering 30kb of flash per board for our system.

The hardware design will include an Atmega328 DIP chip, several electrical components, a wireless Xbee module, power supply, LCD module, and a group of push buttons. As stated before, the 328 chip will have a boot loader burned onto it so allow access to arduino’s libraries which make it easy to configure input / output digital and analog pins. The 328 will require 3.3V to 5V to operate as intended. The wireless Xbee module will allow us to communicate with the other boards during programming and operation. The Xbee will have its own addressing modes through Unicode to determine which board to communicate to. The Xbee will connect to the 328 chip for transmitting and receiving and will require 3.3V to operate as intended. The LCD module is available to offer a display of statistics from the latest sensor reading. The display is 16x2 characters and is programmed to be a scrolling display. The display is wired to the 328 chip to receive data from memory and will require 5V to operate normally. The exception for the LCD module is pin 15 which will control the LED backlight and will require 4.2V instead of 5V. The push buttons will offer the user to turn the system on from being in sleep mode and to override the watering schedule. The push buttons can be pressed and the LCD module should wake up to display the latest sensor data for a period of time before being idle again. A button will also be reserved for an override action, allowing the user to tell the system to water now regardless of the sensor data. The push buttons are connected to the 328 chip and can operate off of a 5V signal. Typical wiring of pushbuttons allows the option to determine which state is on and which state is off. 5V is applied to the button, so while the button is HIGH, the status is actually LOW. The button is pressed to complete a circuit that will make the button status LOW, and the pin status is actually HIGH. For this board, the maximum voltage needed is between 5V to 7V to operate as intended.

The software design is programmed onto the 328 chip for the use of the libraries that arduino has to offer. The software will take advantage of the pin status for the receiver pin, transmitter pin, digital IO pins, and the analog IO pins. The receiver and transmitter pins will most likely be used by the Xbee module while the digital IO pins are used by the LCD module. The software is written as a continuous loop that will offer a sleep mode during the timer delay. Once the delay expires, the system will wake up and perform its new cycle. The software is used to interface to the sensor and solenoid subsystems, be used to interface the LCD module to the user, and to control the power consumption of the board during idle and active times. The code should fit easily into the flash memory since we’re planning on 32kB to be plenty of space. The software is written once to get it operating as intended, and then some parts are rewritten to make sure the code is efficient.

4.3 Sensors Subsystem

The sensor subsystem acts as the eyes and ears for Eco-Sense providing all the data needed to make the decisions as to when to water and when not to water. The sensor subsystem hardware design is designed to reduce power and increase its cost effectiveness. The software is written onto the Atmega 328 microprocessor chip.

The hardware design will include an Atmega 328 microprocessor chip, a humidity sensor, a temperature sensor, a barometric/temperature sensor, a moisture sensor, a rain detector, an Xbee transmitter/receiver, and a power supply system. With the sensors there are 3 different kinds of connections, I2C for the barometer/temperature, digital for the rain detector, and analog voltage signals for the rest. The barometer/temperature sensor and hygrometer are required to be in a separate section of the sensor subsystem housing, allowing for necessary exposure to the elements. This separation is to prevent exposure to other components. All of these components run off either 3.3V to 5.0V which is to be provided by the power supply.

The software of the sensor subsystem runs the system in cycles to save power and follow the following process. The process for the sensor system starts with the sensor subsystem in sleep mode with an internal timer of about 30 minutes is running. When the internal timer of the sensor subsystem comes to an end, the microcontroller will initiate the all sensing elements (excluding rain detector), and send the data to the microcontroller. At the microcontroller the data is merged into a single data stream to be sent to the central HUB subsystem via the Xbee transmitter/receiver. The sensor subsystem then goes into a sleep mode for 30 minutes until the next sensor measurements is needed. Apart from this signal the software needs to be able to interrupt the sleep cycle when the central HUB subsystem sends the signal to begin the watering cycle. The watering cycle starts with the activation of the rain sensor for 5 minutes before the water begins. After 5 minutes the sensor subsystem is to send a signal to the central HUB informing it of the rain status. If the signal for rain is positive, the watering cycle is over and the sensor subsystem returns to its sleep mode. If the signal was negative the rain sensor stays on until a signal from the central HUB signals the end of the watering cycle, a 60 minute time out timer ends, or the rain sensor detects rain. If the rain sensor detects rain while watering, a signal is to be sent to the central HUB indicating rain. The sensor subsystem then returns to its sleeping state for 30 minutes.

4.4 Solenoid Subsystem

The solenoid subsystem’s responsibilities is to provide a way to control the water distribution to the rest of the system. The water distribution will include a water solenoid valve that is connected to an Atmega 328 DIP chip to act as a controller circuit. This subsystem will also contain an Xbee module for receiving the water signal from the central HUB. The solenoid subsystem will only receive data from the central HUB station and will only turn on if the water signal being transmitted is a 1. Software is written with the arduino boot loader burned into the 328 chip. The 328 chip will offer 32kB of flash memory while the boot loader will use 2kB of that memory, leaving 30kB of flash for this software to run on.

The hardware design will include an Atmega328 DIP chip, several electrical components, a wireless Xbee module, a water solenoid valve, and a power supply. The user will not be interfacing with this subsystem which makes this system very simple compared to the other two subsystems. An arduino boot loader is burned onto the 328 chip just like the other subsystems to offer access to the premade libraries. The voltage required for the 328 is between 3.3V to 5V to operate as intended. The main pins that are used on the 328 chip are receive, transmit, and the digital IO pins. An Xbee module is mounted on this subsystem to take advantage of the receiving and transmitting pins on the 328 chip. The Xbee will have its own addressing modes through Unicode so an address is available for the central HUB subsystem. The Xbee will connect to the 328 chip and will require 3.3V to operate as intended. The 328 chip will then process the data and determine which pins go HIGH and which pins go LOW. The water solenoid valve will open if the threshold voltage is reached. Since the threshold voltage is much higher than the system voltage requirement, we plan to incorporate a separate power module just for the valve. The power module is connected through a relay and we will have a pin on the arduino control the relay. Common voltage requirements for a water solenoid valve range from 6V to 12V DC. To avoid too much water consumption by the valve, a process is written to start a delay timer that will represent the duration of the valve being open. The benefits for this strategy are efficient water use, and not drawing power from the separate power supply when it’s not needed. For this board, the maximum voltage needed is between 5V to 7V for all the components, while an additional 6V to 12V is supplied for the water solenoid valve only.

The software design is programmed onto the 328 chip for the use of the libraries that arduino has to offer. The software will take advantage of the pin status for the receiver pin and digital IO pins. The receive pin is used by the Xbee module while the digital IO pins are used by the relay and water solenoid valve. The software is written as a single loop that will activate when the water signal from the central HUB subsystem reads HIGH. Once the delay timer expires, the relay is set to open and the solenoid valve should close until the next HIGH signal is read. This board will most likely try to take advantage of a low power mode, but since the Xbee is the only device on, it might not be worth putting this system to sleep. The Xbee uses low current draw and can remain on to listen for signals by the central HUB subsystem. The power mode options are still being decided, but will most likely not be incorporated on an initial design and only be added as an addition feature in the final design. The code should fit easily onto the flash memory since we’re planning on 32kB to be plenty of memory for this board.

**5.0 Project Hardware and Software Design**

5.1 Central HUB Subsystem

The central hub system will incorporate an LCD module, wireless module, and button inputs. The hardware and software will coexist to create a user interface for the user. The schematics and their descriptions can be found below. The software for this design can be found below the hardware section.

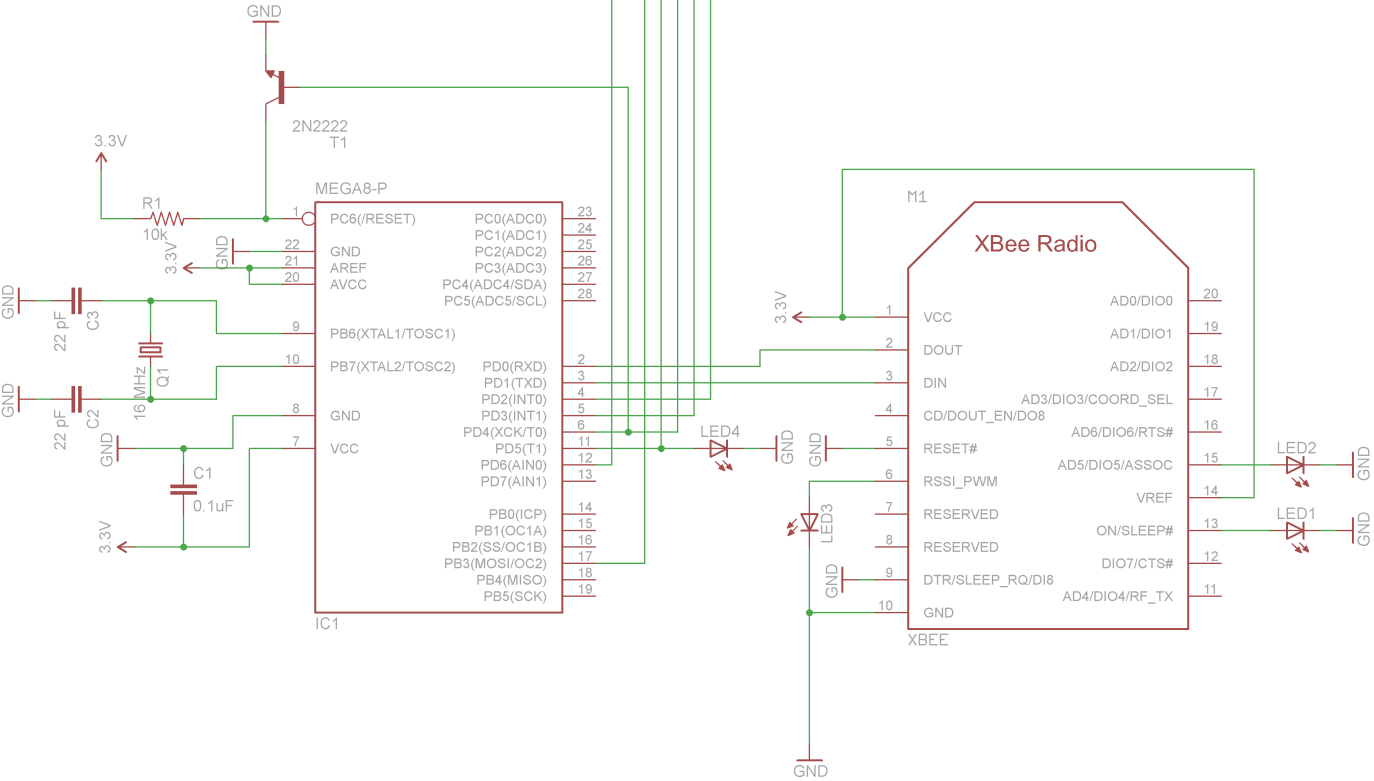


Figure 5.1.0: Atmega 328 DIP chip to Xbee module

The Atmega 328 is connected to the Xbee module as shown in Figure 5.1.0. The Xbee uses the 328’s digital IO pins along with receive and transmit pins. The Xbee will operate off of 3.3V while the Atmega 328 will run off of 5V and stepped down to 3.3V when needed. The Xbee is connected to a few LED’s for debugging during transmission mode and receiving mode. The reset pin is grounded due to the 328 chip will have this feature already implemented. The digital out pin is connected to the receiving pin of the 328. The arduino community has provided a library for interfacing the Xbee module. This is the connection during the normal operation but this may layout may need to be reversed to achieve a wireless programming implementation.

The circuit to the left of the 328 chip is used to couple the oscillator pins that the crystal uses. The 328 will use a 16MHz crystal for operation. A 0.1uF capacitor is placed between VCC and GND to prevent current loopback during the active stages. The 328 chip will offer a 5V signal and a 3.3V signal which are already common sources for the rest of the board. This chip is mounted as a DIP on the PCB for ease of access. Shown below is the schematic for the Atmega328 chip connecting to the LCD module.

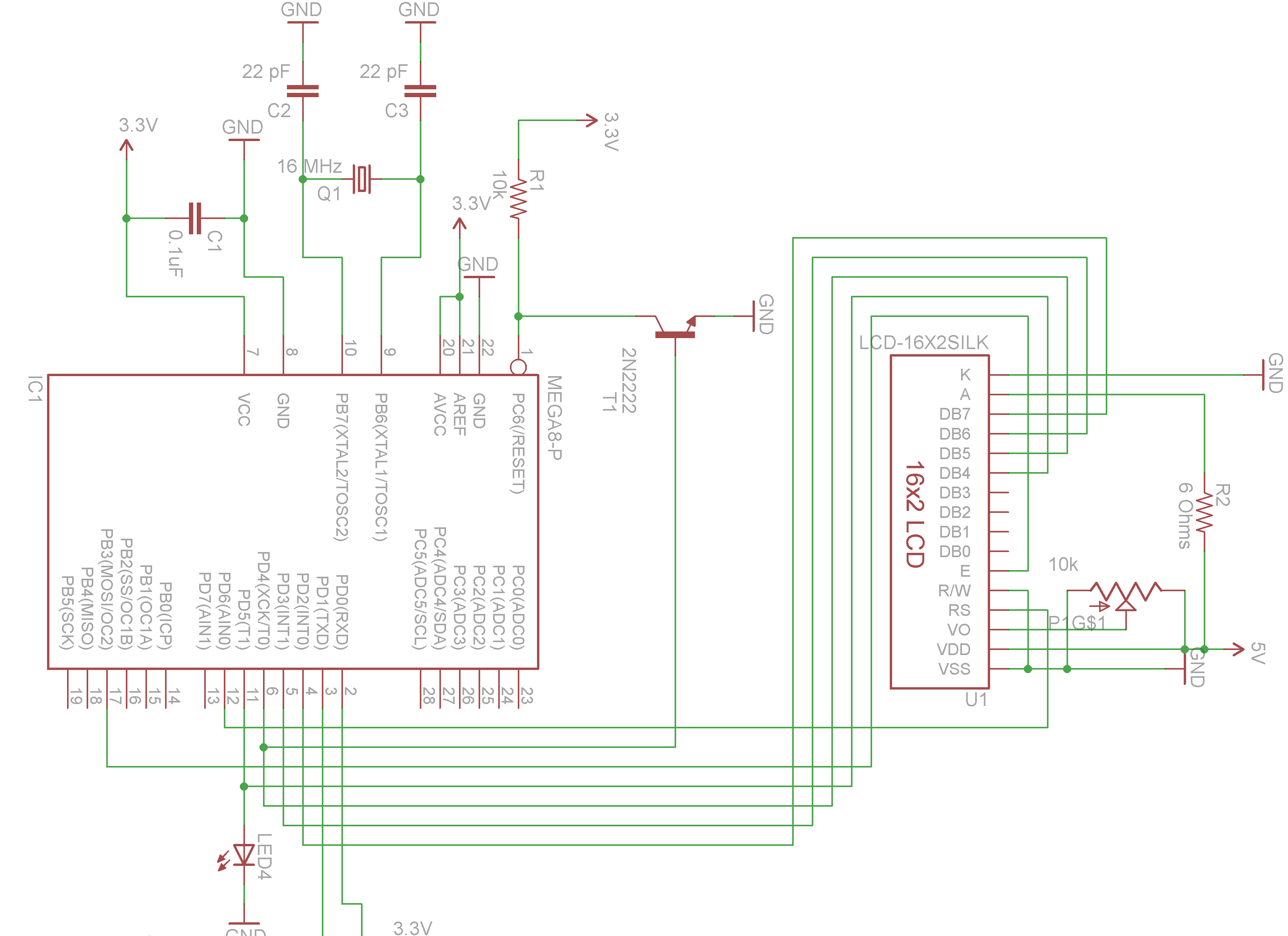


Figure 5.1.1: Atmega328 DIP chip to GDM1602k LCD module

The Atmega 328 is connected to the GDM1602k LCD module through several digital IO pins as shown in Figure 5.1.1. The LCD module will run off of a 5V source while branching to pin 15 (A) with a 6 ohm resistor to create a source of 4.2V. Pin 15 is used as the backlight driving voltage and allows the LCD module to have LED backlighting. This setup will prove useful during dark and can be turned off at any time. A potentiometer is connected to pin 3 (V0) on the LCD module to adjust the character lighting. The LCD module can be interfaced through the 328 chip using libraries provided by the arduino community. The libraries make it easy to write to the LCD screen and offer a few options for scrolling text and space management on the screen. The libraries are written into the 328 chip along with the Xbee libraries. Again, the circuit on the 328 chip has the same purpose as explained above in the 328 to Xbee schematic. Shown below is the Atmega 328 chip connected to both the LCD module and the Xbee module.

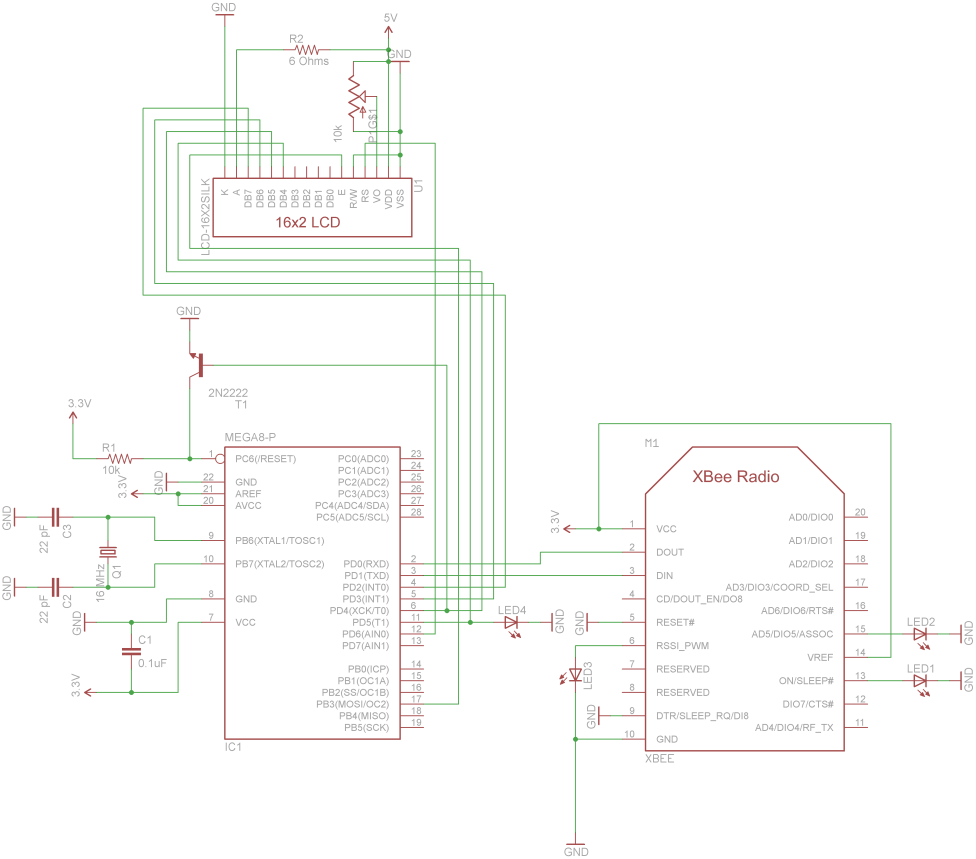


Figure 5.1.2: Atmega 328 connected to Xbee module and LCD module

As shown in Figure 5.1.2, the LCD and Xbee can be connected simultaneously without having to share any of the pins. All the modules are using their respective power source and each module can get data from the 328 without interfering with each other. Along with the schematics, software will have to be implemented into the 328 chip to be able to use the LCD and Xbee modules. Functions that are used to interface to the respective devices are shown in table 5.1.3.

|  |  |
| --- | --- |
| Function | Purpose |
| Receive Data | Receiving incoming data from sensor subsystem using Xbee |
| Transmit Data | Transmit data to sensor subsystem or solenoid subsystem using Xbee |
| Store in Memory | Stores incoming sensor data into flash |
| Button Input | Reads user input and redirects to another function |
| Extract Data | Parses incoming data to respective variables |
| Display Statistics | Displays incoming sensor data to LCD module |
| System Timer | Delay timer, system only fully active when expired, otherwise sleeping |
| Moisture Flag | Algorithm to determine if water is needed using moisture data |
| Temperature Flag | Algorithm to determine if water is needed using temperature data |
| Humidity Flag | Algorithm to determine if water is needed using humidity data |
| Pressure Flag | Algorithm to determine if water is needed using pressure data |
| Rain Flag | Algorithm to determine if water is needed using rain data |
| Sum Flags | Algorithm to determine if water is needed using calculated flag values |
| Water Signal | Data to be transmitted to determine the solenoid valve status |
| Override | User can manually push a button to start a watering process |

Table 5.1.3: Software functions being implemented in the Atmega 328 chip

5.2 Sensor Subsystem

Hardware: The sensor sub system will include the microcontroller ATMega328, a wireless module, a power source, and all of the sensors needed to sense the environment.

ATMega328 and Xbee: The microcontroller in figure 5.2.1 is responsible for reading the sensors, storing their data, then transmitting the data through the wireless module; all while conserving as much power as possible. Each sensor that is connected to the microcontroller will have a set of pins connected to it. The Xbee will receive the data through the serial connection on the ATMega328 from pins 2 and 3. The ATMega328 has an external clock of 16 MHz to regulate the processing in the core. A .1uF bypass capacitor is used to help filter out noise connected to the VCC of the microcontroller. For optimal bypassing, it is best to place this capacitor as close as possible to the ATMega328.

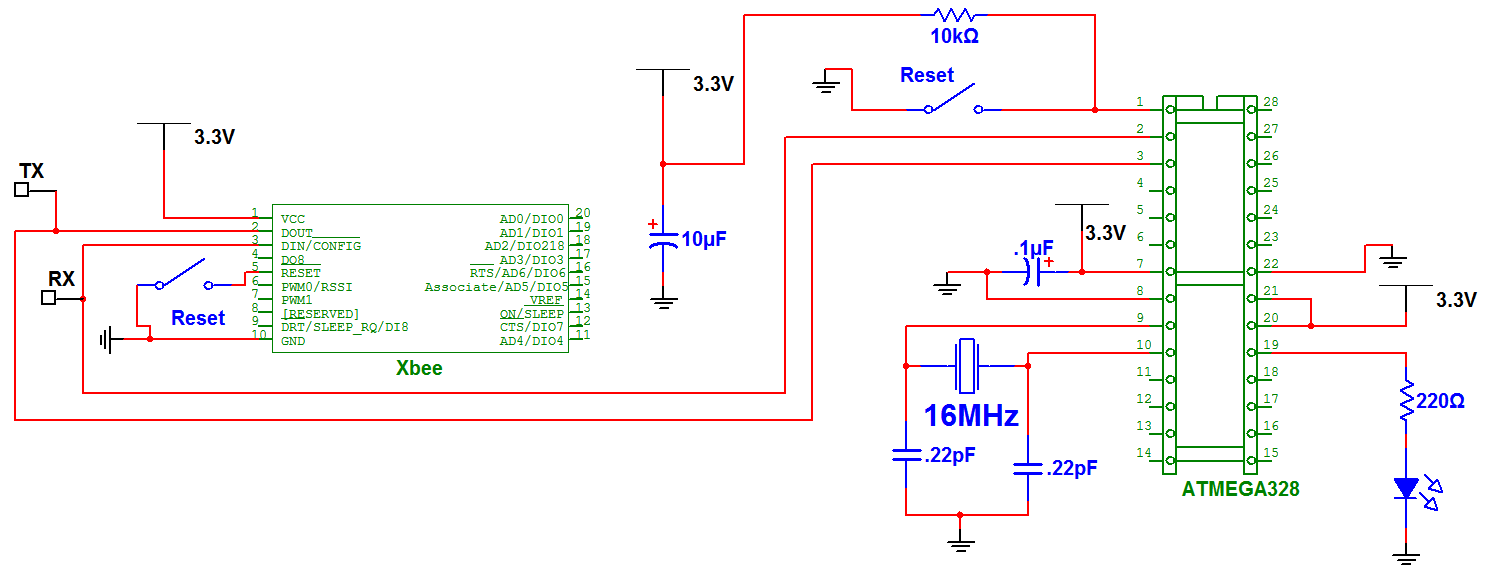


Figure 5.2.1: Xbee schematic

Power: For the power, we are using lithium polymer batteries in series giving 7.4 volts nominal. However we have different components that require different voltages. To obtain the voltages we need without having to have multiple sets of batteries we will need to use a DC to DC converter. For one option, we can use a linear voltage regulator such as the LM337. This regulator is adjustable to give a large range of output voltages by changing the resistor values. To determine which resistor values to obtain, first we determine the output voltage that we want. Then select the resistor 1's value, which is the resistor on the output pin. Then plug everything into this equation:

Selecting R1 as 200 Ω and Vout at 3.3 volts will give R2 as 328 Ω. However there is not a 328 Ω resistor in the real world so using 178 Ω and a 150 Ω in series to get that resistance. Alternatively if we select R1 as 120 Ω, then R2 would be 196.8 Ω. We would need a 150 Ω and a 47 Ω Ω in series to get that resistance.

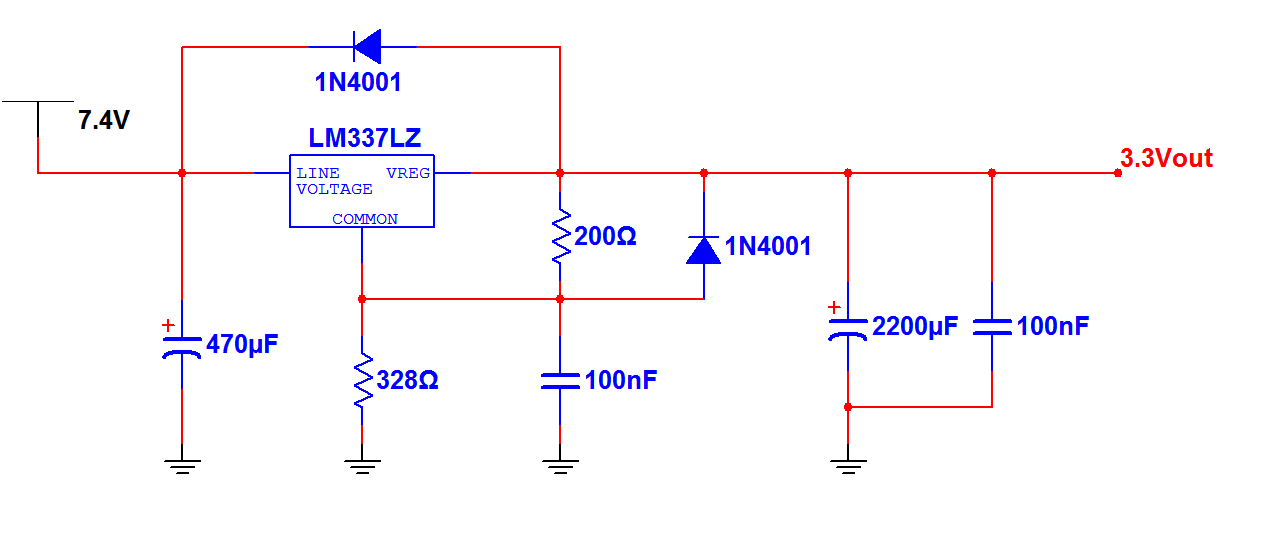


Figure 5.2.2: Regulator Schematic

Linear regulators, such as the one in figure 5.2.2 get their output voltage by burning off the excess energy as heat. For a small voltage difference between the Vin and Vout, it is not that big of a deal. But for large differences with high current pull, then a large heat sink would be required. For example, 7.4 volts in and 3.3 volts out gives a 4.1 volt difference. If .5 amps were being pulled through the regulator then using Power = Voltage x Current gives 2.05 watts of heat. That is a large amount of wasted energy that would be better used in powering the device. Another problem with linear regulators is that if the difference between the input voltage and output voltages becomes too small, then the output voltage drops. This can lead to brownouts and parts turning off. To avoid this, there are LDO or low drop out voltage regulators that use a PNP transistor inside rather than two Darlington NPN transistors. This lowers the drop out voltage (voltage difference between input and output) by removing a transistor because the dropout voltage is the voltage across the transistors in the regulator. However doing so causes some stability issues internally in the circuit. This may or may not cause problems with the power and noise on the output. This choice is fine for our project, but next is a slightly more expensive but greatly more efficient choice for DC-DC conversion.

Switch mode conversion allows for a higher efficiency voltage conversion. This is done by the internal op-amp that pulses a switch that is connected to an internal pulse generator that switches a transistor on and off. When the transistor is on, the energy from the input is stored in an external magnetic field energy storage device called an inductor. A capacitor is used to help clean the noise from the voltage and smooth out the signal curve. Compared to the above linear voltage regulator, this conversion method is more power efficient 75+% because it is not dissipating unwanted power as heat. From the late 1980s, the efficiency has increased due to the invention of the FET, which has higher and more efficient switching frequencies than power bipolar transistors that have greater switching loss and require a more complicated driver circuit. Because we are trying to maximize our battery life, this increased efficiency is beneficial to increasing the running time of our devices.

The circuit in figure 5.2.3 uses a LM2592HV switching regulator that has a fixed voltage output of 3.3 volts. This also has a switch pin that can turn the device on and off. The pin 4 is a feedback line that monitors the voltage after the inductor. It’s important that this connection is away from the inductor due to mutual inductance that would cause interference on the chip. The diode is a Schottky diode from ground to the voltage out. The reason this is used is because the voltage drop is between approximately 0.15–0.45 volts and a normal diode has a voltage drop of 0.6–1.7 volts. This lower voltage drop can provide higher switching speed and better system efficiency. With a higher efficiency, and no heat sink (or a small one) needed, this regulator would be a better choice for our project that relies heavily on being power efficient.

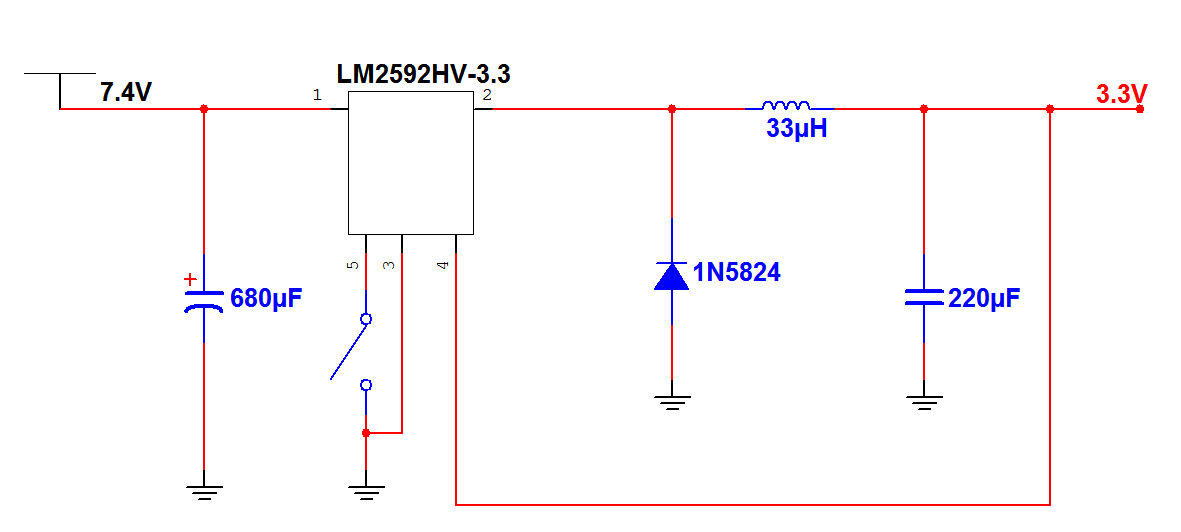


Figure 5.2.23: Regulator Circuit

Bypass capacitors: In figure 5.2.4, a bypass capacitor is used to filter out AC noise on a DC power or signal line. It is important to make sure that these bypass capacitors are as close as possible to the device needing them. For the choice of which type of capacitor to use, aluminum electrolytic capacitors are infrequently used in switching applications because they hold a charge too well. This does not suit them for the rapid cycling of the signal. Using ceramic capacitors are a good choice for fast switching applications due to their low cost, good performance, and many values. To calculate the capacitances needed for the bypass capacitors to filter out the ripples, we will use the following equation.

In this equation, v(t) is the voltage across the capacitor, q(t) is the change held in the capacitor, and i(t) is the current in the capacitor. This equation shows that the voltage is proportional to the current through the capacitor, and inversely proportional to the capacitance (C). The first part shows that a capacitor as expected, acts like an energy reservoir. Like a reservoir, adding a charge into the capacitor raises the voltage and pulling a charge lowers it. Also, the more capacitance there is, the longer it would take to charge it. We use the symbol ┤├ for a schematic representation of an ideal capacitor, but in the real world, a capacitor looks like the figure below.

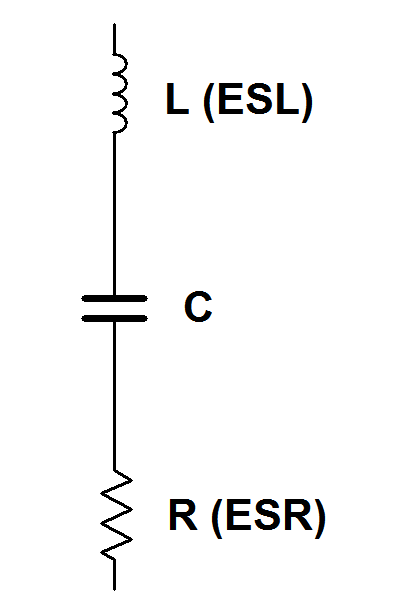


Figure 5.2.4 Equivalent circuit of bypass capacitors.

In figure 5.2.4 the Equivalent Series Resistance (ESR) and an Equivalent Series Inductance (ESL) are the two hidden components in a capacitor that are needed for calculating the bypass capacitance. In the figure 5.2.5, it is a made up transient response of a capacitor graphing the frequency of the signal verses the impedance of the bypass capacitor. To the left (lower frequencies) of the Series Resonance Frequency (SRF), the impedance drops according to the 1/ωC expression. But at higher frequencies above the SRF along the ωL slope of inductance, the impedance increases. This is important to choosing the correct bypass capacitors for each of the components that need them. While higher capacitance helps keep voltage ripples small, the higher inductance does just the opposite: above the SRF point, a higher inductance results in more voltage ripples for the same change in current. So if we just randomly choose values because they seem good, we could be actually adding more noise in the system which is not desirable. Bypass capacitors that have lower inductances are better at bypassing.

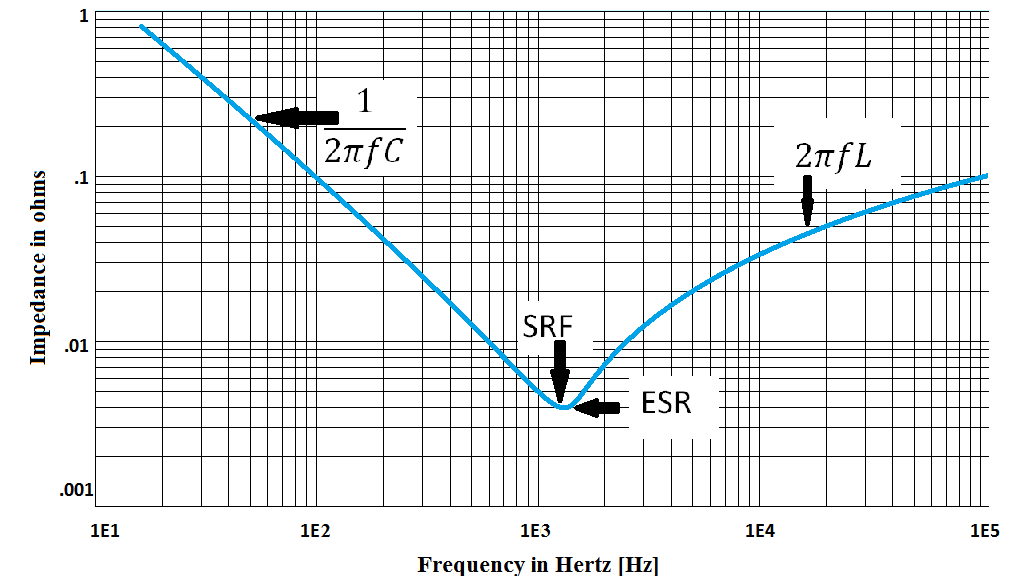


Figure 5.2.5: Frequency diagram

Moisture sensor: As this style of moisture sensor acts like a resistor, if a DC voltage source is used, a simple voltage divider can be used to obtain a usable voltage reading proportional to the moisture content of the soil. This voltage is sent to the micro controller for data processing. Figure 5.2.6 shows the circuit design of resistive moisture sensor.

  
Figure 5.2.6 Resistive Moisture Sensor Circuit

The voltage values sent to the microcontroller are proportional to the resistance cause by the soil and proportional to R1 by the following formula.

Where R2 is the resistance from the soil between the sensor probes and R1= 100KΩ. R1 = 10KΩ

The addition of a capacitor in parallel with the sensor probes would take care of noise from both the DC voltage source and noise from the sensor probes. The noise filter will have a cutoff frequency of around 5 hertz.

With R1 = 100KΩ the capacitor will have to be around 310pF. 330pF is the closet common capacitor value and are acceptable for this filter.

Ground temperature sensor*:* The TMP36 comes in three packet types, the two mounted chip types SOT-23, SOIC\_N, and the commonly used for transistors package, TO-92. To measure the ground temperature the TMP36 needs to be in the soil and away from other devices that may produce heat. Because of this there needs to be a way to secure the device from the elements other than the ambient temperature of the soil.

To create the mounting needs for the TMP36 temperature sensor, we are creating a custom probe to be inserted into the soil. To create this we are using an empty pen tube, wire bus, and silicon for sealing. We will pull the bus through the pen tube exposing individual wires at the end. The exposed wires then need to be connected to the appropriate terminals of the temperature sensors then soldered. The individual terminals are then separated by electrical tape to insure they never connect to each other unintentionally. The sensor then is to be pulled into the pen tube until only the end of the sensor is exposed. Next we will add in silicon chalk till it completely fills in all gaps in the pen tube and let dry for 24 to 72 hours. For added aesthetic value the sensor probe will then be completely covered in black electrical tape.

Rain sensor: In figure 5.2.7 uses a comparator to compare the resistance from the conductor pad. Compares with a calculated resistance value using a voltage divider. When the measured value is greater than the reference resistance, the comparator turns on, passing about 3 volts through. This will go to the microcontroller to signal that rain has been detected.

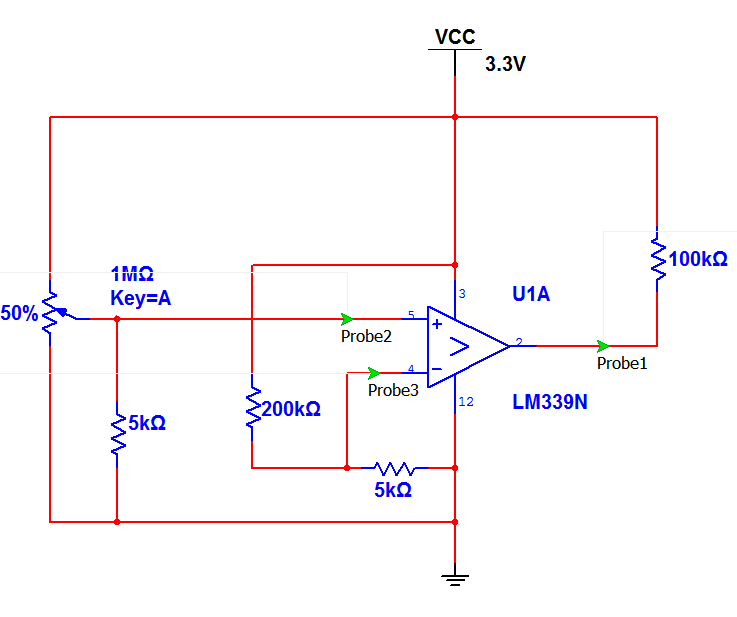


Figure 5.2.7: Rain sensor circuit

5.3 Solenoid Subsystem

The solenoid system includes the water valve and the electronics needed to receiving signals to instruct this node what to do. The solenoid water valve is under pressure from the building's water supply. When a signal is received, the microcontroller will turn the valve on for a given period of time. As a safety measure, in the event that the turn off signal is lost or never sent, the solenoid after a period of time is set to turn off the valve. But if the signal is received before the timer is up, it will turn the valve off.

The microcontroller cannot put out enough current and voltage to run the solenoid valve, so transistor acting like a relay will boost the power from a separate power source. The transistor in figure 5.2.1 is a TIP120 and can support up to 60 volts at 5 amps. This should be more than enough power to run any device in our project using the microcontroller as a switch. The diode is reversed to protect the transistor when the solenoid turns off, because of the feedback voltages. Without it, the transistor will take the hit and after repeated uses, it may damage it.

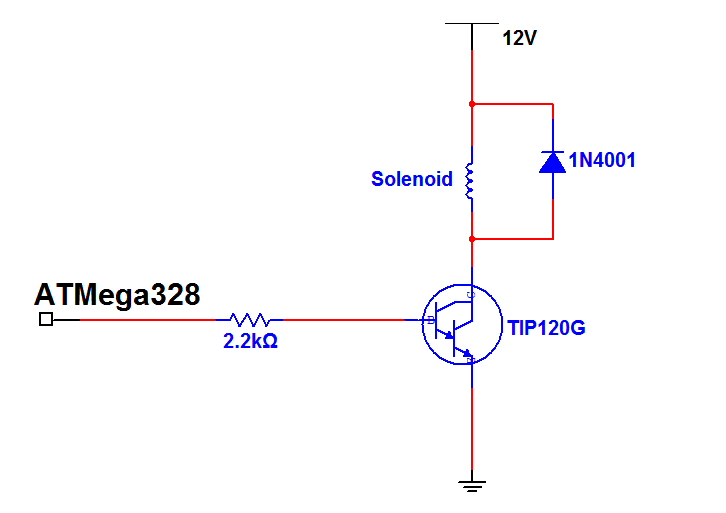


Figure 5.3.1: TIP120 transistor

**6.0 Prototype Construction**

6.1 System Housing

Each subsystem will have its own unique housing. The main features for each of the subsystems are different from each other and must be built accordingly. The central HUB subsystem will need a thin design so it can be mounted on a wall. The sensor subsystem will need a sealed compartment for the electronics and an exposed compartment for the applicable sensors with wiring connecting the two compartments. The solenoid subsystem will need a compact design as it is placed either on the spigot of the home or on the wall next to the spigot. Models shown below were made using Google SketchUp with the Educational Licensing. These models are somewhat basic and are only to show the format and functionality that this group would like to attain for each subsystem. The top image of each subsystem is the internal organization while the bottom image of each subsystem is the outer box. These models are subject to change to be much more efficient when actual components are bought and fitted.

6.1.1 Central HUB Subsystem

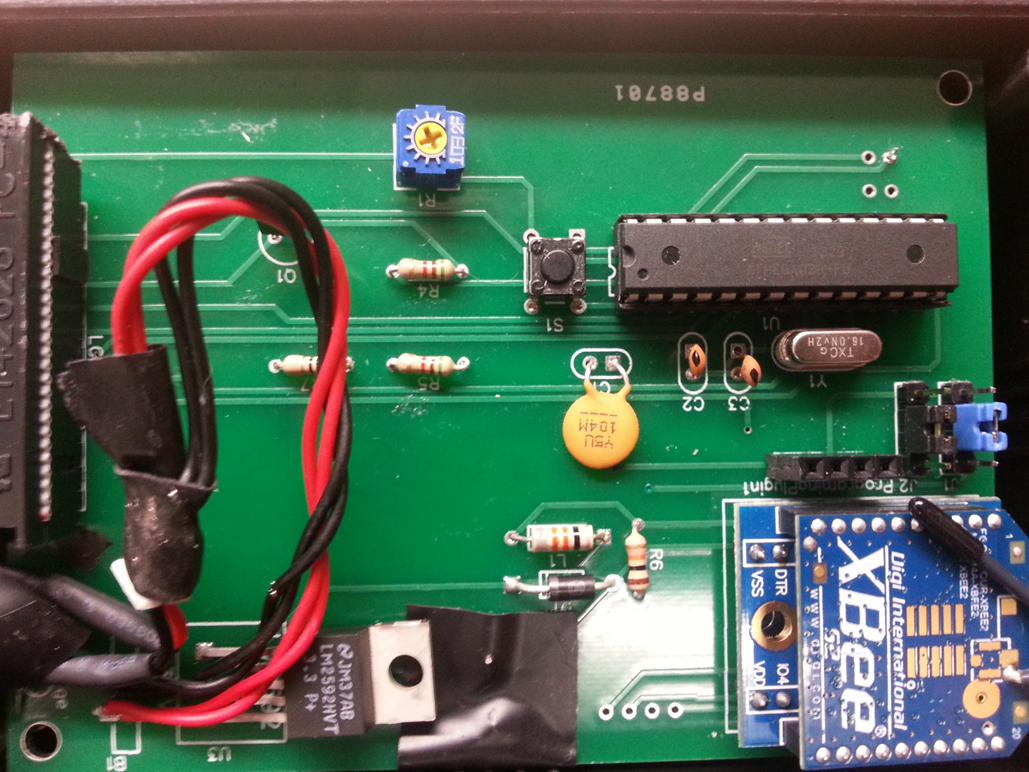


Figure 6.1.1.1: Internal Central HUB Subsystem

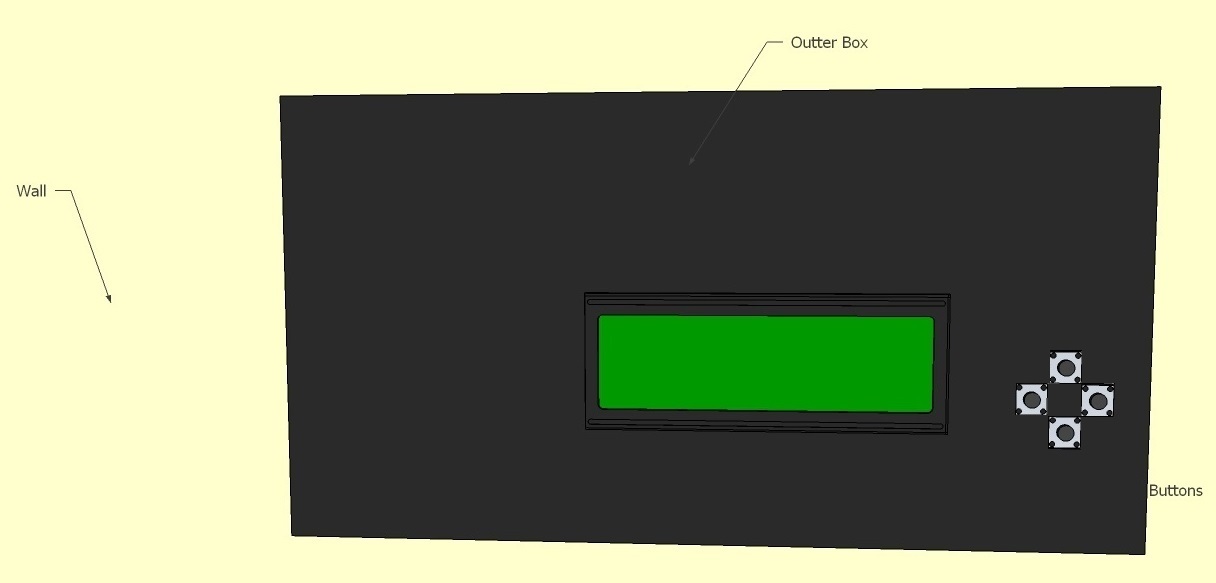


Figure 6.1.1.2: External Central HUB Subsystem

The central HUB subsystem will have a thin design of approximately 2” to 3” wide. This housing needs to be light, portable, and mountable. The user is interfacing with this subsystem so the user interface needs to be simple and intuitive. The LCD module and push buttons are mounted outward a little so a cover plate can fit over the electrical section of this subsystem as shown in Figure 6.1.1.2. We would like to have this module a solid color or design and to not be transparent so the user won’t be bothered with LED lighting and wires around the housing.

6.1.2 Sensor HUB Subsystem

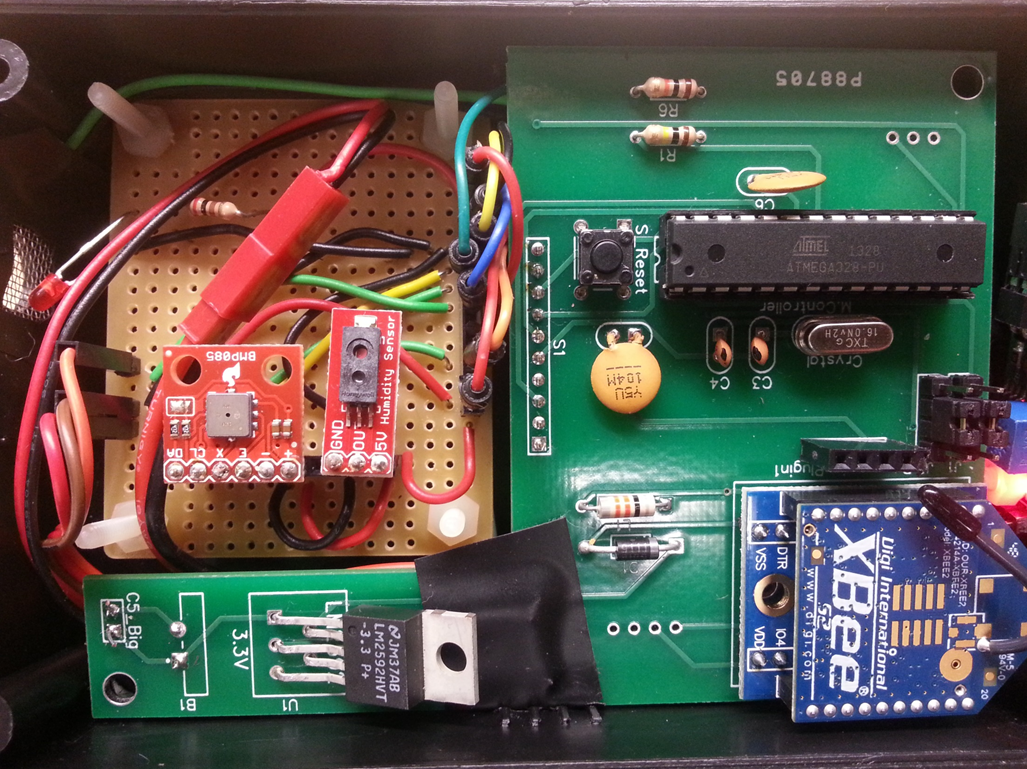


Figure 6.1.2.1: Internal Sensor HUB Subsystem

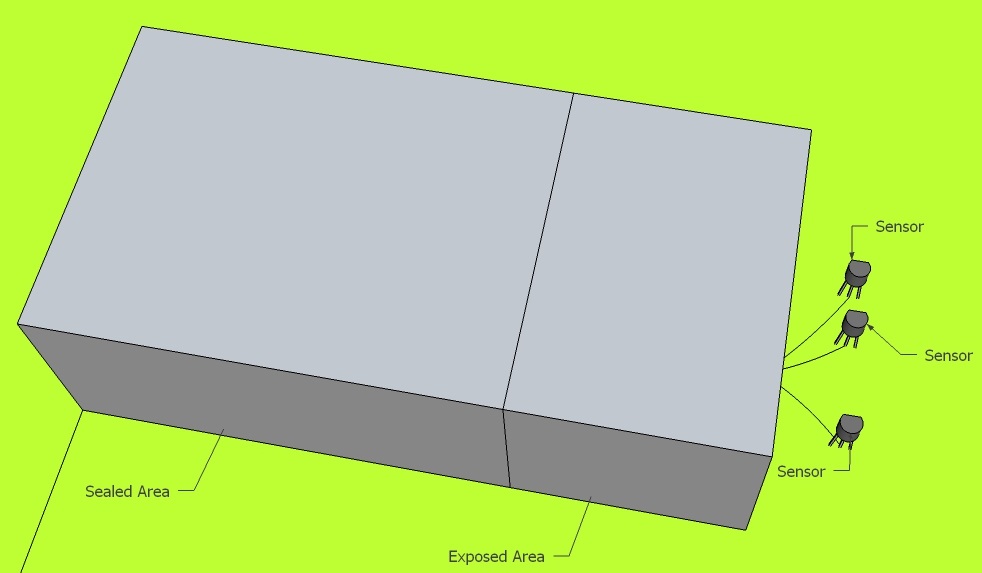


Figure 6.1.2.2: External Sensor HUB Subsystem

The sensor HUB subsystem housing will have a complex design, being the most complicated subsystem of the project. As shown in Figure 6.1.2.1, the housing will contain a sealed area along with an exposed area. The sealed area is to prevent water from leaking in and shorting our electronics since this housing is placed in the way of the sprinkler system during watering periods. The exposed area as shown in Figure 6.1.2.1 is for the sensors that require exposure to the elements without getting soaked by water in order to operate as intended. There are sensors outside the housing to be mounted in the ground for purposes such as water detection and moisture level monitoring. There is a hole drilled between the sealed and exposed sections of the housing. Wires are fed through the hole and the hole is sealed when testing and operation should take place. This approach is still subject to change due to the housing research on a modular design.

6.1.3 Solenoid Subsystem

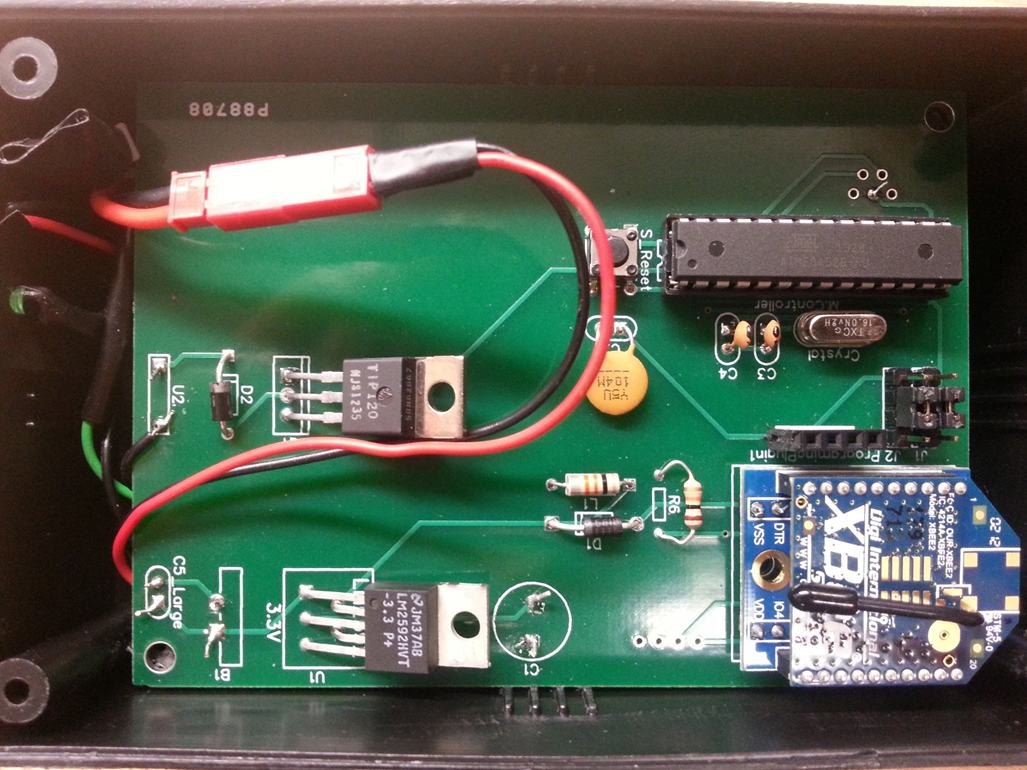


Figure 6.1.3.1: Internal Solenoid Subsystem

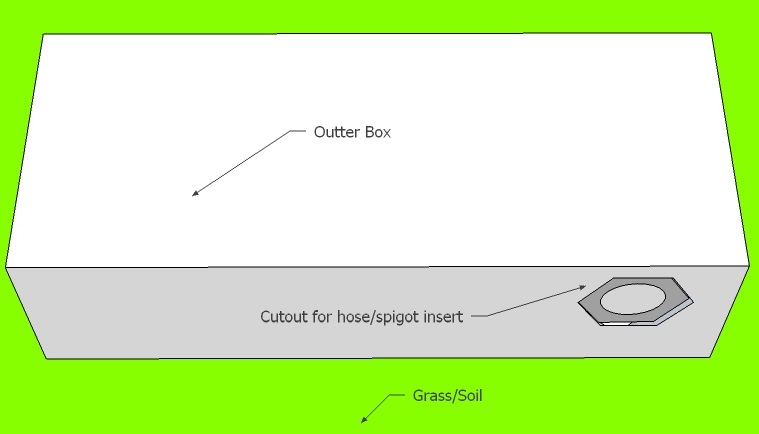


Figure 6.1.3.2: External Solenoid Subsystem

The solenoid subsystem housing will have a small and compact design. This housing will most likely be mounted on the spigot of the home or on the wall next to the spigot depending on the solenoid valve chosen. As shown in Figure 6.1.3.1, the main feature for this housing is to be only as wide as the solenoid valve. Cutouts are made on the sides of the box for hose and spigot inserts for easy connectivity. An optional feature which has come up was to seal off the electrical portion from the solenoid valve in the event that the spigot leaks or the solenoid valve leaks due to human error. We would like this box to be small enough that it can’t be seen easily since the user will not interact with this subsystem during operation.

**7.0 Prototype Testing**

7.1 Microcontroller

The testing of each section of our system may seem like a waste of time, but doing the simple tests first can prevent problems for the more complicated tests. These tests will isolate any problems to the small section that is being tested before we conduct the whole system test. If we were to skip these tests and go straight into system testing, if a problem occurs, we would waste time trying to find the needle (problem) in the haystack (whole system), whereas if small sections were tested as suggested above, then the haystack would be the size of your hand. Which finding a needle in a handful of hay is easy. Once all of the sections are complete successfully, the whole system test should go smoothly since we would have solved the simpler problems beforehand.

Prototype Testing**:** Before any other testing can begin, the microcontroller needs to be physically tested. For the peripherals that include sensors, LED, solenoid valve, push buttons, switches, and everything else that is connected to the microcontroller are connected. Along with all solder joints, traces, and passive components are secure, testing of the microcontroller can begin. For these tests, a multi-meter and oscilloscope is used to verify proper function and improve any problems that may arise. The first test is to turn the microcontroller on and measure the voltage and current going into the chip. The measured value should be in the range as specified in the datasheet. As we will need to send data to and receive data from the microcontroller, it is imperative that the connection between the serial port and the microcontroller is tested. The best way of doing this is just to send a small program to the microcontroller. Loading a program onto the chip such as print "hello world" on the serial interface of 9600 baud rate using Serial.println() and a blinking LED program are two simple examples that shows the connection works as expected. Testing sensors at this point may be problematic since using these sensors is new to everyone in our group and determining if they work properly would be harder than determining if an LED works. Therefore, using an LED with resistor on every digital and analog pin would be a much simpler to work with. The analog pins need to be tested with a potentiometer so make sure they can read variable data, which is what the sensors are providing to them. The varied resistor values can be displayed on the serial output (with a baud rate of 9600). To prevent any power problems in the future, such as brownouts which can make us believe the problem is with the program, the VCC connecting to the microcontroller needs to be tested. This can be done with the oscilloscope. Any abnormal ripple currents on the signal wave can be measured and removed by means of bypass capacitors to filter out the ripples.

Timers and Interrupts: The three different timers can be set to do a subset of the program after a period of time. Rather than using a counter to blink an LED, it could be done with a timer. This is important for the project since it is important to use timers to activate different things. Once the timers programming is working as expected, the interrupts can be tested. There are two different types of interrupts, hardware and software. The only software interrupts are the timers and this is how the LED can be made to blink. As for the hardware interrupts, a digital pin can be set so that when a signal is sent to it, it activates some sub-section of code to run. The interrupts are important because it will allow us to schedule different activities with a timer and when a sensor sends a signal to the microcontroller.

Sleeping (Power modes): The microcontroller is asleep for long durations and need a way to notify the program to wake the core up. Run some activity, possibly transmit some data, and then go back to sleep. Depending on what the power modes offer in respects to power usage and functionality, we need to decide on which modes to use. These modes can be tested and verified by running code that should not be able to run such as using a component that is shutdown like a core timer. Another way to verify the power level is to measure the change in power being used by the microcontroller entering a different power mode. This can be done with a microcontroller. Lastly, we will measure the current going to the microcontroller while entering different power modes, and then record the results.

Microcontroller testing: For this, it would involve testing the program on each microcontroller, then all together. The initial test would be to send a message from the computer through the USB then to the microcontroller and get a message in return. This would be done for each node to ensure that the software on the computer is able to run on each of the microcontrollers. The next test would be to read sensor data for the sensor node and inputs from the push buttons for the central HUB subsystem. This test would show that they are able to receive input data. The next set of tests would be communication between the nodes. The central HUB subsystem would be connected to the computer, and the other nodes will send a message or sensor data to it. The received data is verified by viewing it on the computer's serial input screen. After these tests are complete, further programming and algorithm development can begin since we know that the nodes are able to communicate properly.

7.2 Software

For software testing, the compiler is the main piece of software that needs to be tested for this project. For ease of use, we are using the Arduino compiler for the Arduino development boards. Because it is a simple interface, it has a great community to refer to, and it’s easier to use than a normal AVR programmer that is also much larger in size. The AVR programmer has more functionality than the Arduino one, however we do not need that functionality for the scope of this project.

Testing the compiler with a simple program is important before any complicated programming is done. We need to ensure that we are able send the program from the computer to the microcontroller. Then to be able to receive a response from the microcontroller on the serial communication port (using a baud rate of 9600) on the computer such as "Hello world" or display some counter value. This test will show that we able to send and receive data from the microcontroller. Now more complicated testing can begin. On the PWM pins, an LED can be placed so that the microcontroller will vary the signal to dim the LED. This should test that compiler can control the PWM channels on the microcontroller.

Functionality: The next step is to make sure the compiler can control hardware attributes such as assigning different power levels or to set timers. Once this is done, the testing on the microcontroller testing section can start since it involves the ability to control the behavior of the microcontroller and not just sending a message or making LEDs blink.

7.3 Sensor Testing

The sensors are the only way that Eco-Sense knows what is happening. Without proper working sensors Eco-Sense will not be able to work properly, because of this it is imperative that all the sensors are test to verify that they are connected to the system and giving off proper and accurate results for each sensor.

7.3.1 & 7.3.2 Thermometer and Barometer

This section will contain the testing for both the system thermometer and the barometer because they are on the same component, BMP085. The ground probe thermometer is covered with the ground moisture sensor because of their dependence on each other. Before incorporating the sensors into a single system, several tests are performed on an experiment board to confirm the proper function of each of the components. These tests are testing for proper powering of the device, three separate temperature measurements separate by at least 5 minutes and confirmed by an outside calibrated thermometer, and 3 separate barometric pressure measurements separated by 5 minutes and confirmed by an outside, calibrated barometer. The accuracy of the temperature readings must be within the ±1.0⁰C for temperature and ±3hPa for the barometer each of these given by the data sheet. Confirm that after the three tests there is no overheating. Table 7.3.1 shows the test plan for this section of testing.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Procedure | Pass | Fail | Comments |
| Power Received | X |  | Success |
| Temp. Measurement 1 | X |  | Success |
| Barometer Measurement 1 | X |  | Success |
| Temp. Measurement 2 | X |  | Success |
| Barometer Measurement2 | X |  | Success |
| Temp. Measurement 2 | X |  | Success |
| Barometer Measurement 3 | X |  | Success |
| Overheating | X |  | Success |

BMP085 Tests  
Figure 7.3.1

7.3.3 Hygrometer

The hygrometer will require an outside thermometer to be present for calculations as this hygrometer does not give relative humidity measurements without temperature readings being incorporated with the readings. The tests to be performed on this sensor before it is integrated into the sensor subsystem contains checks for power connectivity, three separate sensor readings are to be confirmed with an independent calibrated humidity measurement. All humidity sensors need to be within the accuracy rating given by the HIH4030 datasheet, ±3.5%. After these three tests the hygrometer are checked for any overheating. Table 7.3.2 shows the tests planned for the HIH-4030 humidity sensors.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Procedure | Pass | Fail | Comments |
| Power Received | X |  | Success |
| Humidity Measurement 1 | X |  | Success |
| Humidity Measurement 2 | X |  | Success |
| Humidity Measurement 3 | X |  | Success |
| Overheating | X |  | Success |

HIH-4030 Tests  
Table 7.3.2

7.3.4 Ground Moisture

The Ground moisture sensor readings are closely related to the ground temperature. Because of this both the ground temperature probe and ground moisture sensor are to be tested at the same time. The ground moisture sensor, being created from scratch, has no general calibration, because of this, testing has to be completed to obtain the soil field capacity and the effects of eat on the moisture sensor. The following tests are to be performed on an experiment board prior to installation in the sensor subsystem.

To determine the effect that heat has on the resistive moisture sensor, testing has to be completed to determine these effects. How these effects are going to be determined is by measuring samples of the same soil with the same water content at different temperatures. To do this several samples from the same soil source are distributed into different cups. These cups are left to dry out for 24 hours prior to the tests. Three ounces of water is applied to the soil at different temperatures created from either refrigerating or microwaving the water before addition to the soil. To make sure that the same amount of water is added and water is not lost due to evaporation, a measuring dropper is used to apply the water after heating or cooling. After allowing the water to sit in the cup of soil for 2 minutes moisture and temperature measurements are to be taken and recorded. This temperature configuration is attempted again to get a second measurement near the same temperature. After 10 pairs of these tests are completed the data is to be interpolated to create a formula to relate heat of the soil and the moisture sensor reading. Table 7.3.3 is the chart used to record the data for this test.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Temperature | Moisture | Test | Temperature | Moisture |
| 1 | 77.00F | 803 | **11** | 77.00F | 804 |
| 2 | 78.00F | 815 | **12** | 78.50F | 822 |
| 3 | 81.00F | 850 | **13** | 78.00F | 817 |
| 4 | 81.00F | 852 | **14** | 81.00F | 845 |
| 5 | 78.00F | 817 | **15** | 77.00F | 809 |
| 6 | 77.00F | 805 | **16** | 81.00F | 857 |
| 7 | 78.50F | 821 | **17** | 78.00F | 815 |
| 8 | 81.00F | 849 | **18** | 77.00F | 808 |
| 9 | 78.50F | 819 | **19** | 77.00F | 800 |
| 10 | 77.00F | 799 | **20** | 78.50F | 818 |

Heat Effect Determining Tests  
Table 7.3.3

To determine the soil field capacity, the bucket method is to be used. Four 5 gallon buckets are poured in four separate locations on the soil to be measured. After 24 hours, with no rain, the moisture content is to be taken at all four locations, and averaged together to find the average soil field capacity. If there is rain during the 24 hours wait period, 24 hours have to be waited until this procedure is restarted. Table 7.3.4 is the test plan for determining the soil field capacity.

|  |  |  |
| --- | --- | --- |
| Location | Moisture | Comments |
| 1 | 324 / 80% | Success |
| 2 | 421 / 92% | Success |
| 3 | 376 / 85% | Success |
| 4 | 331 / 81% | Success |

Soil Field Capacity Determining Tests  
Table 7.3.4

7.3.5 Rain Detector

The rain detector is an important piece to Eco-Sense’s sensor subsystem and needs to be tested properly. There are two tests that need to be completed; rain limit calibration and comparator test. The first test is testing the functionality of the comparator. This test will require a voltage applied to the Vin of the comparator to test if triggering occurs. Make sure the comparator does not overheat. The second test is to determine the amount of rain accumulated on the resistive sensor required to send the rain signal. This test is accomplished by applying the desirable amount of moisture to the sensor and changing the resistor values until the sensor detects this moisture. Wipe the sensor and let it dry, then apply water until the sensor triggers to confirm the triggering point. These tests are to be performed before instillation to the sensor subsystem circuitry, table 7.3.5 shows the acceptance test plan for the rain detector.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Procedure | Pass | Fail | Comments |
| Power Received | X |  | Success |
| Comparator Functioning | X |  | Success |
| Comparator Overheating | X |  | Success |
| Rain Detector Sensitivity | X |  | Success |

Rain Detector Tests  
Table 7.3.5

After all these tests are completed and results are all positive, these individual components are to be mounted in the sensor subsystem circuitry and housing where they are to be tested again to make sure there properly functioning. Table 7.3.6 shows the test plan for the sensor system functionality. All measurements are to be confirmed by an outside calibrated meter like in previous tests. For the moisture readings use three separate soil samples with the same water content but different temperatures similar to the approach used in heat effects test for moisture sensors. Make sure these three measurements at different temperatures yield similar moisture readings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BMP085 | | | | |
| Testing Procedure | **Pass** | **Fail** | | **Comments** |
| Mounted Properly | X |  | | Success |
| Connections Made | X |  | | Success |
| Power Received | X |  | | Success |
| Temp. Measurement | X |  | | Success |
| Barometer Measurement | X |  | | Success |
| Overheating | X |  | | Success |
| HIH-4030 | | | | |
| Mounted Properly | X |  | | Success |
| Connections Made | X |  | | Success |
| Power Received | X |  | | Success |
| Humidity Measurement | X |  | | Success |
| Overheating | X |  | | Success |
| Ground Temperature | | | | |
| Connections Made | X |  | | Success |
| Power Received | X |  | | Success |
| Temp. Measurement | X |  | | Success |
| Ground Moisture | | | | |
| Connections Made | X |  | Success | |
| Power Received | X |  | Success | |
| Moisture Reading 1 | X |  | Temperature: 77.00F | |
| Moisture Reading 2 | X |  | Temperature: 81.00F | |
| Moisture Reading 3 | X |  | Temperature: 78.50F | |
| Rain Detector | | | | |
| Mounted Properly | X |  | Success | |
| Connections Made | X |  | Success | |
| Power Received | X |  | Success | |
| Comparator Overheating | X |  | Success | |
| Rain Detection | X |  | Success | |

Sensor System Functionality Tests  
Table 7.3.6

7.4 Solenoid Testing

The solenoid valve will need to go through various testing procedures to determine if it’s compatible with Eco-Sense’s goal. The water valve is tested before putting it into the system and is tested to ensure that the valve can open and close on the threshold voltages that are provided on the data sheet. The solenoid should be able to oscillate at a reasonable rate to meet the demands of the programming process. A test is done to ensure the valve can remain open for an extended period of time in case the need arises. Several tests will also be done to check for leaks in the threading of the valve itself to ensure this project remains a water efficient design. If available, different spigots and hoses are used to check compatibility with a variety of users. Shown below in table 7.4.1 are the tests that are conducted before putting the valve into the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Procedure | Pass | Fail | Comments |
| Open | X |  | Success |
| Close | X |  | Success |
| Oscillation Possible | X |  | Success |
| Thread Leak Test | X |  | Success |
| Extended Open Signal | X |  | Success |
| Hose compatibility >3 | X |  | Success |
| Spigot Compatibility >3 | X |  | Success |

Table 7.4.1: Solenoid Testing Procedure (Initial)

The testing procedure shown above was performed before mounted on the housing. No boards were connected to this module at that time. If any of the tests fail, reevaluation will take place in determining a fix or choosing another solenoid water valve. The ‘thread leak test’ is the most important test after opening and closing due to this project being all about water efficiency. The compatibility tests are just for our group to know if we should include adapters to the final product or if this system will work with a wide range of hoses and spigots due to different sizing. Shown below in table 7.4.2 are the tests that are conducted after putting the valve into the system and connecting to the Atmega 328 pins.

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Procedure | Pass | Fail | Comments |
| Open | X |  | Success |
| Close | X |  | Success |
| Oscillation Possible | X |  | Success |
| Extended Open Signal | X |  | Success |
| Timer Delay Close | X |  | Success |

Table 7.4.2: Solenoid Testing Procedure (Final)

The testing procedure shown above was performed after mounted on the housing. All the components should be connected to the solenoid as if it was a product for the user. The different tests from before include a timer delay close test. This test will only be performed if the open and close tests pass. The ‘timer delay close’ will work with the code on the Atmega 328 to determine if a delay timer can be set to automatically close the valve. If this test does fail, our group can rely on the oscillation test to oscillate the valve for a period of time instead.

7.5 Communications

Wireless Sensor node testing: To test the system a three step process is used to help isolate any potential problems in smaller sub-systems. First each node is connected to a computer to ensure that the program is running as intended.

Sensor Node - Data acquisition:For the sensor nodes, they will gather the raw sensor data then display it through a serial connection on the computer. This is done one sensor at a time to help isolate any problems with the code and sensors. If all were run at the same time for testing, it would be harder to determine if one sensor is affecting another or the code for using the senor could be incorrect. On the serial console, the output should be in the expected tolerances of each sensor. For example if the room temperature is currently 77°F and the temperature sensor reads 79°F but has a ±3°F tolerance then that would be acceptable. But if the temperature was reading 84°F, then it could be a software or hardware problem. It could have a loose connection or the code could have the wrong resistance values to determine the temperature.

Central HUB subsystem - User interface: Before any data is sent to the central HUB subsystem, it must be able to display any data from the serial port to the LCD since Zigbee uses the serial port. To test this, the central HUB subsystem is connected to a computer and a message such as "hello world" is sent from the computer to the central HUB subsystem via USB then displayed on the LCD. Once this works, testing of the communication from the sensor nodes to the central HUB subsystem can take place (now that is it is possible to see any messages sent on the LCD screen).

Sensor Node - Communication: After each sensor is tested and debugged, the information needs to be sent from the sensor node to the central HUB subsystem. The Zigbee or Xbee module needs to be thoroughly tested to make sure that the signal is in the range as specified in the datasheet and that all of the data received on the central HUB subsystem. For a first test, we will use the temperature sensor to send the temperature to the central HUB subsystem and have it displayed on the screen. If this is successful, then the next step can be done.

Range:As mentioned in the power section, it is important that we have adequate range on our communication devices. For range testing, the simplest thing to do is keep sending an incrementing number (a timer count) from the sensor node to the central HUB subsystem. Then try transmitting the data at different ranges 20 feet, 50 feet, 100 feet etc. between the sensor node and central HUB subsystem. The range should be around those said in the datasheet gives both indoor and outdoor ranges for the Zigbee. If they are significantly lower than what is expected, then maybe the hardware design is causing interference or the antenna is not placed correctly on the modules. These types of things can be considered once our prototype is designed to not have these problems.

Data integrity:Once we determine a safe maximum range for the sensor nodes to be away from the central HUB subsystem, we must ensure all the data that is sent is exactly the same data that is received. This can be done by sending sensor data to the central HUB subsystem from a range that is near the maximum range, and at different locations.

Sensor Node - Power modes: Once all of the above test are completed, the power modes can be tested. The reason why the above ones must be done first is because changing the power of the module could affect how far the sensor node and transmit. To test the different modes, we will connect a multi-meter to the power for the sensor node. When the microcontroller switches to different modes such as idle, or powered down, we should see a change in the power usage for the system. Next, we will test the microcontroller and the transmitter since they will consume the majority of the power. When these components switch modes the power should change as specified in their data sheets. The most important mode to ensure that it meets the specifications is the powered down (sleep) mode. Because this mode is what allows the sensor to maintain such a long life time. The expected current usage for sleep mode should be <1mA for the whole system.

Sensor Node - Transmission Cycles:Now the "fun" part, putting all of the tests together. In figure 7.5.1 shows a cycle that the sensor node is following. For this, the sensor node is programmed to follow a schedule similar to the flow chart below. Once on, the sensor node will sleep for a defined period of time, read which ever sensors are needed to be read, then back to sleep. This cycle will occur n times and store the data into memory. Each sensor will have its own delay on being read because some sensors do not need to be read as often as others. The data read from the sensors is stored into memory rather than being transmitted each time the sensors are read. This is because waking up the transmitter, transmitting the data, the going to sleep uses more power than storing several time periods of data into memory and transmitting multiple samples of data at the same time. This will save power which is desirable to do.

Figure 7.5.1: Sensor node cycle

Solenoid Valve Node - Communication:In order to water the lawn automatically, it is important that the central HUB subsystem node can communicate to the solenoid value node. With the central HUB subsystem connected to the computer via USB, a command is given to tell the solenoid to open. On the pins for the solenoid, a multi-meter is connected to verify that the voltage increases when the signal is transmitted, and voltage decreases (near 0) when its told to close. If the value is in an acceptable range for the transistor (TIP120 or similar) to turn on, then the solenoid valve can be connected and tested. Once further programming involving the LCD is done on the central HUB subsystem, the central HUB subsystem will have a manual override option to turn on the water. This test can be repeated with using that command. The final test in the testing of the solenoid value is to actually let the system run. When a lawn watering is scheduled and the proper conditions are met, the value should hopefully open automatically for the predetermined time period then close when done.

All Nodes: The final communication test is conducted once each module has had rudimentary programming completed, most if not all soldering completed, and the previous tests completed. This test will involve the sensor node(s) transmitting data to the central HUB subsystem. A lawn watering schedule based on a date and time. The central HUB subsystem will make any decisions on when to water or not from the sensor data. Then if the conditions are met, the water valve will open, water the grass, then close. This test will allow our group to fine tune anything related to the communication. If transmissions are missed or the wrong data is received, this is the time to fix it. The system log mentioned in the SD card section would be imperative to debug the system at this point, because this test is long. It will span a few days recoding the data, and it is not something that you want to monitor all day every day, but rather only at the end of the test. That’s where the system log comes into play. After this test is complete and the program on each module is optimized, a real world test can be done. This is when the system is running for 1-2 weeks without any human interaction what so ever. After the time period is done as with the test that lasted a few days, the logs can be read to make any (hopefully none) changes to the programs. This concludes the communication testing.

7.6 User Interface Testing

The user interface that is tested belongs to the central HUB subsystem. The interface is a collection of components that when put together offer ease of access and control of the overall system. The components included in the user interface are the push buttons on the front panel, the LCD module, the stability of the housing, and the functionality of the components working together. The push buttons are tested for accuracy and response along with functionality when connected to the Atmega 328 pins. The LCD module is tested by displaying “Hello World” in a variety of patterns to ensure the LCD has the customizability that Eco-Sense will need. The LCD will also be tested for the varying backlight and also the brightness of the characters through the potentiometer. Shown in table 7.6.1 are the tests that were conducted before putting the module into the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | Testing Procedure | Pass | Fail | Comments |
| Push Button | Accuracy | X |  | Success |
|  | Response Time | X |  | Success |
| LCD Module | Turn On | X |  | Success |
|  | Backlight Usable | X |  | Success |
| Housing | Pressure Test | X |  | Success |

Table 7.6.1: User Interface Testing Procedure (Initial)

The testing procedure shown above was performed before being mounted on the housing. No boards were connected to this test at that time. If any of the tests fail, reevaluation will take place to determining a fix or choosing alternative components to use while time permits. The push button test is used with LED’s to ensure they work as intended. The LCD module will still be separated from other components so “Hello World” cannot be displayed just yet, so tests are done to determine if the device can turn on and off successfully and check if the backlight works as intended. The housing tests are for safety of the user and the components which will eventually be inside. Pressure test will not be a brutal test, but rather pushing against the housing to make sure it won’t collapse to crack. Shown below in table 7.6.2 are the tests that were conducted after putting the user interface together in the system and connecting it all to the Atmega 328 pins.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | Testing Procedure | Pass | Fail | Comments |
| Push Button | Accuracy | X |  | Success |
|  | Response Time | X |  | Success |
|  | Functionality | X |  | Success |
| LCD Module | Turn On | X |  | Success |
|  | Backlight Adjustable | X |  | Success |
|  | Hello World | X |  | Success |
| Housing | Component Safety | X |  | Success |

Table 7.6.2: User Interface Testing Procedure (Final)

The testing procedure shown above in table 7.6.2 were performed after mounted on the housing and connected to the Atmega 328 chip. The different tests include functionality, hello world, and component safety. Functionality will refer to the correctness in response to the code provided (such as a simple function to toggle the display on and off). Hello world is a test to check if characters can be written to the LCD module and character brightness will also be part of this test using the potentiometer. Component safety is used to test whether the components can fit safely into the housing and no electrical hazards are in place. This test is for user safety since this subsystem is inside the home.

7.7 Full System Testing

The full system test involves testing the functionality of the system as a whole. This test assumes that the functionality of each individual system as operating properly, this can be assumed from the completion of sections 7.1 through sections 7.6. With the assumption that all individual systems work properly the full system test will not need to include retesting of individual systems. All testing for this section has to be completed in a manner that isolates and controls the inputs to the sensors. The first test is to test that the sensor subsystem starts sensing at the normal time interval of 30 minutes. After this is confirmed, the sensor subsystem microcontroller is temporarily reprogramed for the sensing cycle to begin after 5 minute intervals instead of 30 minutes, this will allow for quicker testing of the system.

The next test in the full system test is the algorithm that determines if it is necessary to water, Time to Water Algorithm. This algorithm makes use of the current sensor data of temperature, humidity and soil moisture content, as a percentage of the soil field capacity, and any water time restrictions programed into the system. To test for this we have to separate this algorithm into its triggering points and test for with watering or no watering based on the inputs given. The first triggering point is temperature and humidity. When the temperature is high and the humidity is low, soil water is at a higher point of evaporation, this is not a good time to water. The algorithm states that times not to water are if the temperature is above 32⁰C, the temperature is above 25⁰C and humidity is less than or equal to 50%, and if the temperature is below 5⁰C. The second triggering point for this algorithm is moisture content. If the moisture content is below 75% of the soil field capacity then watering needs to occur. The last triggering point, it programed in by user, is if watering can happen based on user specific watering times (USWT). To test each triggering point all other measurements that affect other triggering points need to be isolated at or below their watering trigger point. Table 7.7.1 shows the acceptance test procedure for these tests. Each test should vary the experimental value at least three times in both triggering and non-triggering values to verify proper algorithm use. An “X” in USWT signifies that either the user has defined this time for watering or there is no programed watering time and a bold figure signifies constant values.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Temperature | | | | | | |
| Temperature | **Humidity** | **Moisture** | **USWT** | **Time to Water?** | **Pass** | **Fail** |
| 50⁰C | **60%** | **60%** | **X** | No | x |  |
| 40⁰C | **60%** | **60%** | **X** | No | x |  |
| 32⁰C | **60%** | **60%** | **X** | No | x |  |
| 31⁰C | **60%** | **60%** | **X** | Yes | x |  |
| 15⁰C | **60%** | **60%** | **X** | Yes | x |  |
| 6⁰C | **60%** | **60%** | **X** | Yes | x |  |
| 5⁰C | **60%** | **60%** | **X** | No | x |  |
| 3⁰C | **60%** | **60%** | **X** | No | x |  |
| 1⁰C | **60%** | **60%** | **X** | No | x |  |
| Humidity & Temperature | | | | | | |
| 26⁰C | 70% | **60%** | **X** | Yes | x |  |
| 26⁰C | 60% | **60%** | **X** | Yes | x |  |
| 26⁰C | 51% | **60%** | **X** | Yes | x |  |
| 26⁰C | 50% | **60%** | **X** | No | x |  |
| 26⁰C | 40% | **60%** | **X** | No | x |  |
| 26⁰C | 30% | **60%** | **X** | No | x |  |
| 25⁰C | 70% | **60%** | **X** | Yes | x |  |
| 25⁰C | 60% | **60%** | **X** | Yes | x |  |
| 25⁰C | 51% | **60%** | **X** | Yes | x |  |
| 25⁰C | 50% | **60%** | **X** | Yes | x |  |
| 25⁰C | 40% | **60%** | **X** | Yes | x |  |
| 25⁰C | 30% | **60%** | **X** | Yes | x |  |
| Moisture Content | | | | | | |
| 25⁰C | **60%** | 50% | **X** | Yes | x |  |
| 25⁰C | **60%** | 60% | **X** | Yes | x |  |
| 25⁰C | **60%** | 74% | **X** | Yes | x |  |
| 25⁰C | **60%** | 75% | **X** | No | x |  |
| 25⁰C | **60%** | 80% | **X** | No | x |  |
| 25⁰C | **60%** | 90% | **X** | No | x |  |
| User Specified Watering Time | | | | | | |
| 25⁰C | **60%** | 60% | Yes | Yes | x |  |
| 25⁰C | **60%** | 60% | No | No | x |  |
| 25⁰C | **60%** | 70% | Yes | No | x |  |
| 25⁰C | **60%** | 70% | No | No | x |  |

Time to water Algorithm Tests  
Table 7.7.1

When the time to water algorithm triggers that it is indeed time to water, the weather prediction algorithm will activate to determine if there is weather that can produce rain storms on the way or already here. This algorithm makes use of barometric pressure, humidity, temperature, and the rain detector sensor. The prediction algorithm is under development and is completed before implementation of the full Eco-Sense system. The first triggering mechanism of the algorithm is the barometric pressure. If the slope of the barometric pressure is above the threshold set by the algorithm, the possibility of rain is highly probable. The second trigger mechanism is the slop of the humidity. The next trigger is if the slope of temperature measurement is over the requirements of the algorithm. The last trigger for this algorithm is if the rain detector detects the presence of rain. A proper test plan cannot be developed without the algorithm being fully defined.

7.8 Long Term Test

To test the system for long term duration, Eco-Sense is deployed in the yard of one of Eco-Sense’s creators. This test will begin as soon as all other miles stones are completed. The sooner these miles stones are completed, the more time can be given to this long term system test. Before this test begins the memory used for storage of data is erased so only the data collected while Eco-Sense is deployed for its long term test. After the data is collected from this long term test, it is processed to determine the effectiveness of Eco-Sense in collecting the sensor data, the effectiveness of the watering and weather prediction algorithms, and the approximate amount of water used during this long term test compared to traditional automated irrigation systems. Before and after shots are taken to compare the state of the lawn before and after the deployment of Eco-Sense.

**8.0 Administrative Content**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Table 8.1: Bill of Materials

8.2 Division of Labor

Our group consists of 2 computer engineers (Jamie, Daniel) and 1 electrical engineer (Neil). The division of labor was based on experience and interest in each of the topics. Note that even though each section has a name associated with it, it is everyone’s responsibility to help complete each subsystem regardless if it’s their main role. This project is meant to be a learning experience and we try to make the tasks diverse so everyone gets hands on experience with most of the project. The list below shows the labor being divided up between the subsystems and example responsibilities for each of the members.

Central HUB Subsystem

* Wireless Transmission (Daniel)
* Housing Construction (All)
* LCD Module Interfacing (Jamie)
* Xbee Module Interfacing (Daniel, Jamie)
* Atmega 328 Programming (Daniel, Jamie)
* Power Systems (Neil)
* Schematic Design (Jamie)
* Electrical Wiring and Assembly (All)
* PCB Design (All)
* Subsystem Testing (All)

Sensor HUB Subsystem

* Wireless Transmission (Daniel)
* Housing Construction (All)
* Atmega 328 Programming (Daniel, Jamie)
* Schematic Design (Daniel, Neil)
* Sensor Configuration (Neil)
* Power Systems (Neil)
* Algorithm Processes for Sensors (Neil, Jamie)
* Electrical Wiring and Assembly (All)
* PCB Design (All)
* Subsystem Testing (All)

Solenoid Subsystem

* Wireless Transmission (Daniel)
* Housing Construction (All)
* Atmega 328 Programming (Jamie, Daniel)
* Schematic Design (All)
* Solenoid Configuration (Jamie)
* Power Systems (Neil)
* Electrical Wiring and Assembly (All)
* PCB Design (All)
* Subsystem Testing (All)

8.3 Senior Design II Milestones

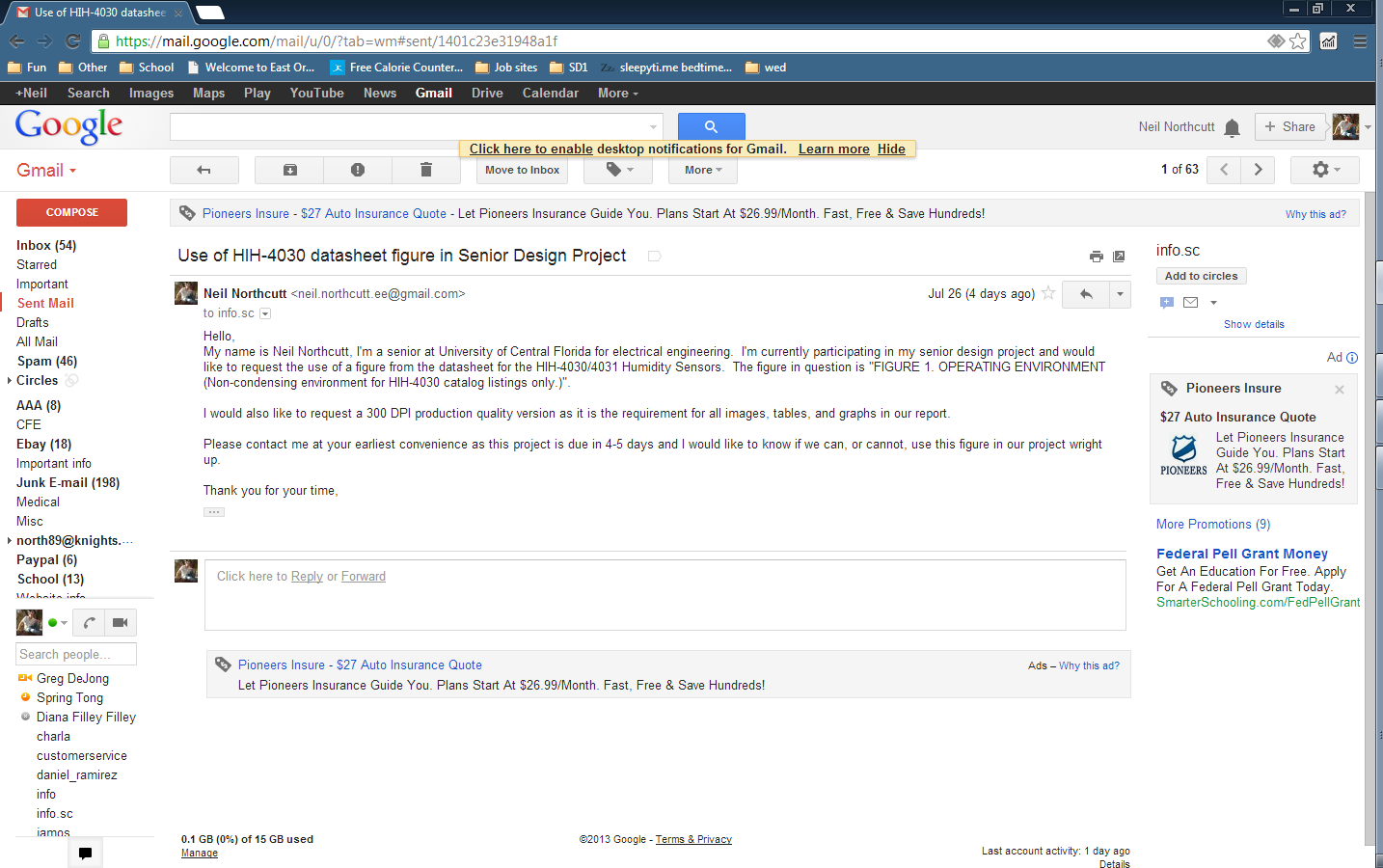
Listed below in table 8.3.1 is our groups projected deadlines for each of the subsystems to complete the final design.

|  |  |
| --- | --- |
| Topic | Deadline |
| Group Meeting SD2 | August 23, 2013 |
| Schematic Testing | August 31, 2013 |
| Part Acquisition and Samples | September 13, 2013 |
| Part Testing | September 20, 2013 |
| PCB Layout | September 27, 2013 |
| PCB Acquired | October 4, 2013 |
| User Interface | October 11, 2013 |
| Sensor Configuration | October 11, 2013 |
| Solenoid Configuration | October 11, 2013 |
| Wireless Transmission | October 11, 2013 |
| Power Systems | October 11, 2013 |
| Atmega328 Programming 25% | October 18, 2013 |
| Atmega328 Programming 50% | October 25, 2013 |
| Atmega328 Programming 75% | November 1, 2013 |
| Atmega328 Programming 100% | November 8, 2013 |
| Central HUB Subsystem Complete | November 15, 2013 |
| Sensor HUB Subsystem Complete | November 15, 2013 |
| Solenoid HUB Subsystem Complete | November 15, 2013 |
| Testing | November 22, 2013 |
| Buffer Period | 9 Days |
| Last Day | December 2, 2013 |

Table 8.3.1: Senior Design II Milestones

**9.0 Appendices**

9.1 Appendix A - Copyright Content



9.2 Appendix B - Bibliography

, . /www.ti.com/lit/an/snla135/snla135.. 10 Dec. 2011. Texas Instruments. 28 Jul. 2013. <http://www.ti.com/lit/an/snla135/snla135.pdf>.

Lewinsohn, H..IEEE Xplore - Active addressing of passive matrix displ. 14 Dec. 1994. IEEE. 28 Jul. 2013.<http://ieeexplore.ieee.org/xpl/login.jsp?reload=true&tp=&arnumber=404668&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs\_all.jsp%3Farnumber%3D404668>.

, . /www.sparkfun.com/datasheets/LCD/HD44780..0 ------ 1999. Hitachi. 28 Jul. 2013. <http://https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.

, . Basic 16x2 Character LCD - Black on Green 5V - SparkFun Electron. 0 ------ 2008.Sparkfun. 28 Jul. 2013. <http://https://www.sparkfun.com/products/255>.

, . LCM-S01602DSR/A LumexOpto/Components Inc | 67-1768-ND | Digi.0 ------ 2013.DigiKey. 28 Jul. 2013. <http://www.digikey.com/product-detail/en/LCM-S01602DSR%2FA/67-1768-ND/469799>.

, . LCM-S01602DTR/M LumexOpto/Components Inc | 67-1781-ND | Digi.0 ------ 2013.DigiKey. 28 Jul. 2013. <http://www.digikey.com/product-detail/en/LCM-S01602DTR%2FM/67-1781-ND/469805>.

, . /www.sparkfun.com/datasheets/LCD/HOW%20DOES%20IT%20WORK..0 ------ 2010.HantouchUSA. 28 Jul. 2013. <http://https://www.sparkfun.com/datasheets/LCD/HOW%20DOES%20IT%20WORK.pdf>.

, . Frequently Asked Questions About Pressure Washing - P.S.I. Fac. 0 ------ 2008. PSI Factor. 28 Jul. 2013. <http://www.psifactor.ca/faqs.php>.

Miranda, F.R, et al. /www.irnase.csic.es/users/jefer/pdf/Miranda\_2005.. 21 Mar. 2006. Science Direct. 28 Jul. 2013. <http://www.irnase.csic.es/users/jefer/pdf/Miranda\_2005.pdf>.

Beckmann, Benjamin and Gupta, Ajay ./www.cs.wmich.edu/gupta/pubsPdf/pansy%20icsi06%20and%20jisas07.. 21 Mar. 2007. http. 28 Jul. 2013. <http://www.cs.wmich.edu/gupta/pubsPdf/pansy%20icsi06%20and%20jisas07.pdf>.

, . /www.acromag.com/content/application-note-autonomous-irrigation-control-8500. 28 Jul. 2013. Acromag. 28 Jul. 2013. <http://www.acromag.com/content/application-note-autonomous-irrigation-control-8500586>.

, . Magnetic resonance imag. 4 Jan. 2002. How does it work? 30 Jul. 2013. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1121941/

, . /www.decagon.com/assets/Manuals/10HS-Manual. 4 Nov. 2010. http. 30 Jul. 2013. <http://www.decagon.com/assets/Manuals/10HS-Manual.pdf>.

, . A cheap soil moisture sensor - Garden. 28 Jul. 2012. . 30 Jul. 2013. <http://gardenbot.org/howTo/soilMoisture/>.

, . /www.coastal.org/spec\_sheets/s1201-SoilMoistureSensor.. 12 Dec. 2011. Costal. 30 Jul. 2013. <http://www.coastal.org/spec\_sheets/s1201-SoilMoistureSensor.pdf>.

, . /www.sjrwmd.com/floridawaterstar/pdfs/SMS\_field\_guide.. 15 Aug. 2012. 30 Jul. 2013. <http://www.sjrwmd.com/floridawaterstar/pdfs/SMS\_field\_guide.pdf>.

, . /www.cheapvegetablegardener.com/tag/ardui. 30 Jul. 2013. 30 Jul. 2013. <http://www.cheapvegetablegardener.com/tag/arduino/>.

, . /pdfserv.maximintegrated.com/en/ds/DS18S20.. 30 Aug. 2012. 30 Jul. 2013. <http://pdfserv.maximintegrated.com/en/ds/DS18S20.pdf>.

, . Yetanother Arduino Thermome. 3 Apr. 2013. Yetanother 30 Jul. 2013. <http://www.yetanother.eu/arduino/articles/temp\_basic.php>.

, . /www.analog.com/static/imported-files/data\_sheets/TMP35\_36\_37.. 29 Nov. 2010. Analog Devices 30 Jul. 2013. <http://www.analog.com/static/imported-files/data\_sheets/TMP35\_36\_37.pdf>.

, . LM35DZ Datash. 25 Aug. 2013. http. 30 Jul. 2013. <http://www.futurlec.com/Linear/LM35DZ.shtml>.

, . /www.analog.com/static/imported-files/data\_sheets/AD22105.. 28 Sep. 2006. Analog Devices30 Jul. 2013. <http://www.analog.com/static/imported-files/data\_sheets/AD22105.pdf>.

, . /www.ti.com/lit/ds/symlink/lm45.. 13 Apr. 2013. Texas Instruments 30 Jul. 2013. <http://www.ti.com/lit/ds/symlink/lm45.pdf>.

, . /www.analog.com/static/imported-files/data\_sheets/AD22103.. 10 Sep. 2010. Analog Devices 30 Jul. 2013. <http://www.analog.com/static/imported-files/data\_sheets/AD22103.pdf>.

, . /www.not2fast.com/electronics/component\_docs/dims/to92\_dim.. 10 Mar. 2007. Not2fast 30 Jul. 2013. <http://www.not2fast.com/electronics/component\_docs/dims/to92\_dim.pdf>.

, . DIY Waterproof DS18B20 Temperature Sensor | Practical Ma. 31 Jul. 2013. Practical Maker. 30 Jul. 2013. <http://www.practicalmaker.com/sensor/diy-waterproof-ds18b20-temperature-sensor>.

, . BMP085 Digital Pressure Sensor Data Sheet. 31 Jul. 2011. Adafruit. 30 Jul. 2013. <http://www.adafruit.com/datasheets/BMP085\_DataSheet\_Rev.1.0\_01July2008.pdf>.

, . Arduino Barometric Pressure Sensor - BMP085 Break. 27 Jul. 2013. Hobby King. 30 Jul. 2013. <http://www.hobbyking.com/hobbyking/store/\_\_26917\_\_Arduino\_Barometric\_Pressure\_Sensor\_BMP085\_Breakout.html?gclid=CPqH1baKtrgCFenm7AodPW8AGA>.

, . /www.freescale.com/files/sensors/doc/data\_sheet/MPL115A1.. 27 Feb. 2013. Freescale. 30 Jul. 2013. <http://www.freescale.com/files/sensors/doc/data\_sheet/MPL115A1.pdf>.

, . 10µA/second 2.375 V to 5.5 V Supply RoHS compliant This is the SFE MPL115A1 Barometric Pressure Sensor. Now mo. 27 Jul. 2013. 47593 SFE MPL115A1 Barometric Pressure Sensor Digital barometric pressure sensor 50 kPa to 115 kPa absolute pressure Average current consumption. 30 Jul. 2013. <http://www.eshopglobe.com/en/miscellaneous/47593-sfe-mpl115a1-barometric-pressure-sensor.html>.

, . /www.hoperf.com/upload/sensor/HP03M.. 29 Aug. 2012. Hope RF. 30 Jul. 2013. <http://www.hoperf.com/upload/sensor/HP03M.pdf>.

, . /www.wunderground.com/resources/pressure\_records.asp?M. 1 Feb. 2013. 30 Jul. 2013. <http://www.wunderground.com/resources/pressure\_records.asp?MR=1>.

, . bildr » Sensing Barometric Pressure | BMP085 + Ardu. 16 Jun. 2011. Bildr. 30 Jul. 2013. <http://bildr.org/2011/06/bmp085-arduino/>.

Haby, Jeff. WEATHER PREDICTION EDUCAT. 16 Dec. 2007. The Weather Prediction. 30 Jul. 2013. <http://www.theweatherprediction.com/habyhints/18/>.

, . /www.micro4you.com/files/sensor/DHT11.. 16 Aug. 2010. D-Robotics UK. 30 Jul. 2013. <http://www.micro4you.com/files/sensor/DHT11.pdf>.

, . /www.sensirion.com/fileadmin/user\_upload/customers/sensirion/Dokumente/Humidity/Sensirion\_Humidity\_SHT1x\_Datasheet\_V5.. 16 Apr. 2012. Sensirion. 30 Jul. 2013. <http://www.sensirion.com/fileadmin/user\_upload/customers/sensirion/Dokumente/Humidity/Sensirion\_Humidity\_SHT1x\_Datasheet\_V5.pdf>.

, . /www.sparkfun.com/datasheets/Sensors/Temperature/HH10D.. 16 Apr. 2012. Hope RF. 30 Jul. 2013. <http://https://www.sparkfun.com/datasheets/Sensors/Temperature/HH10D.pdf>.

Tushec, Simon. Interfaceing HH10D with arduino. 17 Jun. 2012. Arduino. 30 Jul. 2013. <http://tushev.org/electronics/arduino/item/47-interfacing-hh10d-with-arduino>.

, . 385 - $12.50 :Adafruit Industries, Unique & fun DIY electronics and k. 17 Jun. 2012. Adafruit Industries. 30 Jul. 2013. <http://www.adafruit.com/products/385#Technical Details>.

, . Humidity sensor - HH10D - SparkFun Electron. 28 Jul. 2013. Sparkfun. 30 Jul. 2013. <http://https://www.sparkfun.com/products/10239>.

, . /www.sparkfun.com/datasheets/Sensors/Weather/SEN-09569-HIH-4030-datasheet.. 28 Jul. 2013. Honeywell. 1 Mar. 2008. <http://https://www.sparkfun.com/datasheets/Sensors/Weather/SEN-09569-HIH-4030-datasheet.pdf>.

, . Sensing Humidity with the HIH-4030. 25 Nov. 2012. Bildr. 1 Mar. 2008. <http://bildr.org/2012/11/hih4030-arduino/>.

, . HIH-5030/5031 Series Low Voltage Humidity Sensors. 1 Mar. 2010. Honeywell. 30 Jul. 2013. <http://sensing.honeywell.com/index.php?ci\_id=49692>.

, . Humidity Sensor - HIH-4030 Breakout - SparkFun Electron. 25 Jul. 2013. Sparkfun. 30 Jul. 2013. <http://https://www.sparkfun.com/products/9569>.

, . HIH-4000 Humidity Sensors. 10 Feb. 2010. Honeywell. 30 Jul. 2013. <http://sensing.honeywell.com/index.php/ci\_id/51626/la\_id/1/document/1/re\_id/0>.

, . Effects of over-watering.Red Hen Turf Farm. 30 Jul. 2013. <http://www.redhenturf.com/pdfs/Effects%20of%20over-watering%20lawns.pdf>.

Bellis, Mary. History of the Barometer. About.com. 30 Jul. 2013. <http://inventors.about.com/od/tstartinventors/a/Barometer.htm>.

Goldstien, Mel. Moisture and Humidity. 2002. InfoPlease. 30 Jul. 2013. <http://www.infoplease.com/cig/weather/moisture-humidity.html>.

, . Rain Detector Module for arduino.InfoPlease. 30 Jul. 2013. <http://www.engineeringshock.com/rain-detector-module---arduino.html>.

, . Low Power Low Offset Voltage Quad Comparators.Nov. 1999. Texas Instruments. 30 Jul. 2013. <http://www.ti.com/lit/ds/symlink/lm339-n.pdf>.

" Comenzando con ZigBee. | C.I.r.E.."C.I.r.E..N.p., n.d. Web. 6 July 2013. <http://webdelcire.com/wordpress/archives/1714>.

"8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash."ATMEL.N.p., n.d. Web. 26 June 2013. <www.atmel.com/Images/doc8161.pdf>.

"Comparing Low-Power Wireless Technologies - Wireless Solutions | Digikey Technology Zones." DigiKey Electronics - Electronic Components Distributor. N.p., n.d. Web. 12 July 2013. <http://www.digikey.com/us/en/techzone/wireless/resources/articles/comparing-low-power-wireless.html>.

"Design with Microcontrollers: SD/SDHC Card Interfacing with ATmega8 /32 (FAT32 implementation)." Design with Microcontrollers. N.p., n.d. Web. 25 July 2013. <http://www.dharmanitech.com/2009/01/sd-card-interfacing-with-atmega8-fat32.html>.

ErazoVillegas , Miguel Angel, Seok Yee Tang , and Yi Qian. "Wireless Sensor Network Communication Architecture for Wide-Area Large Scale Soil Moisture Estimation and Wetlands Monitoring ." -. University of Puerto Rico, n.d. Web. 6 July 2013. <users.cis.fiu.edu/~meraz001/TR-NCIG-0501.pdf>.

Grady , Steve . "Zero Power Wireless Sensors Using Energy Processing."Energy Harvesting Technology.N.p., n.d. Web. 26 June 2013. <www.mouser.com/applications/storing\_harvested\_energy/>.

Hempstead, Mark, Michael Lyons, David Brooks, and Gu-Yeon Wei."Survey of Hardware Systems for Wireless Sensor Networks." -. N.p., 10 Jan. 2008. Web. 6 July 2013. <www.eecs.harvard.edu/~dbrooks/hempstead\_JOLPE08.pdf>.

"How Solar Cells Work." PENN ELECTRIC RACING.N.p., n.d. Web. 6 July 2013. <https://fling.seas.upenn.edu/~electric/dynamic/?page\_id=244 http://www.enviroshop.com.au/info/36>.

"I2C Protocol - How It Works, and What to Watch Out For - I2C Protocol Home page." I2C Protocol - How It Works, and What to Watch Out For - I2C Protocol Home page. N.p., n.d. Web. 12 July 2013. <http://www.i2cprotocol.com/>.

"I2C Tutorial."Devantech Ltd. N.p., n.d. Web. 12 July 2013. <http://www.robot-electronics.co.uk/acatalog/I2C\_Tutorial.html>.

"Incredible DIY: Building a SD Card Temperature logger (ATMEGA328 + SDCARD + LM35)." Incredible DIY.N.p., n.d. Web. 25 July 2013. <http://www.incrediblediy.com/2013/03/building-sd-card-temperature-logger.html>.

Kickliter, Jay. "Chasing 'trons - Main - Energy Meter."Chasing 'trons - Main.N.p., n.d. Web. 6 July 2013. <http://chasingtrons.com/main/2011/3/24/energy-meter.html>.

Kusy, Brano. "Autonomous Systems Lab, CSIRO ICT Centre ." OPAL Sensor Node: Computation and Communication Advances in WSN Platforms. CSIRO, n.d. Web. 7 July 2013. <www.cooperating-objects.eu/fileadmin/dissemination/joint-seminars/20120427-kusy.pdf>.

Ladgaonkar, B. P, and A. M Pawar."DESIGN AND IMPLEMENTATION OF SENSOR NODE FOR WIRELESS SENSORS NETWORK TO MONITOR HUMIDITY OF HIGH-TECH POLYHOUSE ENVIRONMENT." -. Version 2231-1963. International Journal of Advances in Engineering & Technology, 1 July 2011. Web. 6 July 2013. <www.ijaet.org/media/0001/1I3DESIGN-AND-IMPLEMENTATION-OF-SENSOR-NODE-FOR-WIRELESS-SENSORS-NETWORK-TO-MONITOR-HUMIDITY-OF-HIGH-TECH-POLYHOUSE-ENVIRONMENT-Copyright-IJAET.pdf>.

"Low-Power Wireless Sensor Networks - Sensor Solutions | DigiKey Technology Zones ."DigiKey Electronics - Electronic Components Distributor. N.p., n.d. Web. 25 June 2013. <http://www.digikey.com/us/en/techzone/sensors/resources/articles/low-power-wireless-sensor-networks.html>.

"Microcontrollers and Wireless Connectivity - Wireless Solutions | DigiKey."DigiKey Electronics - Electronic Components Distributor. N.p., n.d. Web. 25 June 2013. <http://www.digikey.com/us/en/techzone/wireless/resources/articles/microcontrollers-and-wireless-connectivity.html>.

Novak, Istvan. "Inductance of Bypass Capacitors, Part I."PCB Design 007.QuietPower column, n.d. Web. 16 July 2013. <www.electrical-integrity.com/Quietpower\_files/Quietpower-6.pdf>.

"Overview of solar panel types | EEP."EEP - Electrical Engineering Portal | Energy and Power.N.p., n.d. Web. 7 July 2013. <http://electrical-engineering-portal.com/overview-of-solar-panel-types>.

Raghunathan, Vijay, Curt Schurgers, Sung Park, and Mani a. "Energy-Aware Wireless Microsensor Networks." -. N.p., n.d. Web. 25 June 2013. <www.cs.wmich.edu/wsn/doc/power/power.pdf>.

Schmitz, Tamara . "Don't Byass Your Capacitor." Electric design.N.p., 12 Sept. 2011. Web. 25 July 2013. <electronicdesign.com/boards/don-t-bypass-your-capacitor>.

Smith, Phill. "Comparisons between Low Power Wireless Technologies." -. N.p., n.d. Web. 12 July 2013. <www.csr.com/sites/default/files/white-papers/comparisons\_between\_low\_power\_wireless\_technologies.pdf>.

"The comparison of Wi-Fi, Bluetooth and ZigBee « SENA BLOG." SENA Bluetooth Serial Adapter, Wireless Serial Device Server, Bluetooth Serial Module, Bluetooth Access Point, Bluetooth RS232, Device Networking, Serial Server, Bluetooth Dongle, Cable Replacement, Wireless Serial. N.p., n.d. Web. 12 July 2013. <http://www.sena.com/blog/?p=359>.

Topic. "IEEE Xplore - Error Page." IEEE Xplore - Home.N.p., n.d. Web. 25 June 2013. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05395793>.

"Types of Solar Panels."EnviroShop Sustainability Solutions.N.p., n.d. Web. 7 July 2013. <http://www.enviroshop.com.au/info/36>.

Wang, W. S., R. O’Keeffe, N. Wang, M Hayes, B O’Flynn, and C O’Mathuna."Practical Wireless Sensor Networks Power Consumption Metrics for Building Energy Management Applications." -. N.p., n.d. Web. 6 July 2013. <zuse.ucc.ie/forumbau2011/papers/19.pdf>.

"i/o blog: Solar powered, Wireless autonomous radiation sensor network node." i/o blog. N.p., n.d. Web. 6 July 2013. <http://theiopage.blogspot.ca/2013/01/solar-powered-wireless-autonomous.html>.

nuumio. "Weather station | Let's Make Robots!." Let's Make Robots! | Let's Make Robots!.N.p., n.d. Web. 6 July 2013. <http://letsmakerobots.com/node/25773>.