6-2015

Automated Aquaponics Design Report

Carson Miller
Union College - Schenectady, NY

Automated Aquaponics
Design Report

By

Carson Miller

* * * * * * * * *

Submitted in partial fulfillment
of the requirements for
Honors in the Department of Electrical Engineering

UNION COLLEGE
June, 2015

ABSTRACT

"Aquaponics is the cultivation of fish and plants together in a constructed, recirculating ecosystem utilizing natural bacterial cycles to convert fish waste to plant nutrients" [1].

Aquaponic systems offer a solution to environmentally conscious, fiscally responsible individuals who wish to produce food products without the use of pesticides, genetic modification, or non-renewable resources. Further, the proliferation of inexpensive microprocessors and high quality sensors offers an opportunity for dynamic monitoring and error correcting within the aquaponic system. These aquaponic and electronic systems will be combined in this project to create an electronically monitored aquaponic ecosystem.

TABLE OF CONTENTS

TABLE OF FIGURES AND TABLES

INTRODUCTION

As the population of the world continues to increase, so does the demand for clean water and high yield farmland. These resources are unfortunately finite, and in especially high demand in cities and underdeveloped areas. As time progresses, both individuals and companies will be looking for more efficient ways to produce food for human consumption that conserve space and water. Electronically monitored aquaponics offers the ability to produce fish and vegetative growth for human consumption in a highly efficient, sustainable ecosystem, thus solving the aforementioned problems. Organically filtered water leads to healthy fish and, as long as high quality fish food and lighting systems are used, fast growing and high quality produce will result. [1].

It was the purpose of this project to create an electronically monitored system capable of producing vegetables and fish for consumption. In addition to operating without the production of any waste water, the system was also designed to provide feedback and monitor the biological components so that any serious problems were detected and displayed user interface. The project will be considered successful when the system operates for over one month with a grow bed full of healthy vegetables and tank full of live fish. Further, temperatures not between 40C and 75C, grow light on times not between 10 and 16 hours, and flood cycles not between 10 and 20 minutes must be detected and displayed on the user interface while also sounding an alert buzzer.

This report explores the design requirements for the project, as well as the final design and alternatives that were considered during the research and construction phase of the project. After providing a background in the field of aquaponic and electronic systems, this report will examine the requirements that the prototype completed at the end of the project must meet to be considered successful. Alternative designs choices that could have been read the reasoning behind the selections that were made will also be discussed in this paper. Finally, the proposed design will be outlined, as well as the progress that has been made thus far.

BACKGROUND

Basic, electronically monitored aquaponic systems have been built in the past by both researchers and corporate entities. In [2], other researchers created an automated aquaponic system. They used a glass fish tank stocked with comet goldfish to provide the aquaculture based element of the system. The plants were grown in a lit grow bed above the fish tank, and temperature sensors were used in conjunction with an Arduino computing platform for monitoring and feedback. Further, they added a 4x20 LCD display to convey temperature information to the system's users. Their project documentation provided an excellent reference for the future research in Aquaponics that will be performed at Union. It was our goal to expand upon the research they performed to create a better electronic system capable of thoroughly monitoring the conditions of the aquaponic system and conveying that information to the user for action if necessary. Differences between the research performed by [2] and the research performed at Union include the addition of temperature and light sensors, water level sensors, indication diodes, pushbutton interfaces, and, eventually, the breeding of edible fish. Further, the Arduino computing platform was not be used in favor of a less expensive microcontroller that still performs all desired system functions [3].

With respect to manufacturability, there were commercially available electronically controlled and monitored aquaponic systems and components. These systems, available from the Kijani Grows company, used the Arduino computing platform and were avaliable in various configurations [4]. The price of complete, commercially available and electronically monitored aquaponic systems ranged from $999 to $1524. Additionally, the aquaponic system platform, without any electronics included, was available for between $500 and $949. Further, various electronic parts were available for purchase with the entire electrical system available for $495-$690, the sensor package available for $205, and their Arduino shield available for $25-$55. The fish tank size of Kijani Grows' aquaponic systems ranges from 75 to 150 gallons and specific productions yields for each system are not specified. These products were different from the system that was built at Union because they use the Arduino computing platform and did not offer the sensing and interface technologies that were implemented at Union [4].

Agricultural production within New York State is overseen and regulated by the New York State Department of Agriculture and Markets. The Department of Agriculture and Markets does not consider roadside stands, on-farm outlets, or farmers markets to be, "retail food stores" and thus, these bodies are not required to meet the strict standards to which retail food production and processing bodies must abide. Since permits are not required for the production or sale of vegetables, government bodies did not need to be contacted for the vegetable production aspect of this project [5]. Additionally, since the goldfish were kept in a, "closed loop, recirculating system," a permit was not required for the aquaculture based portion of the project either [6]. If a commercial venture were to be undertaken, the appropriate permits must be obtained from the New York State Department of Agriculture and Markets and, depending on the fish being produced, the New York State Department of Environmental Conservation [5] [7].

Overall, aquaponics offers great commercial, economic, and environmental opportunities. Economically, aquaponics is feasible because it harnesses several relationships that exist in nature. As a result of these special relationships that exist between the fish, bacteria, and plants, hardware is no longer required for water filtration, and nutritional supplements are no longer necessary as they would be in a hydroponic system [1]. In aquaponics the plants take care of the water filtration and the fish provide the nutrients. Further, as long as high quality, nutritious and organic fish food is used in the system, the consumable fish and vegetation will be natural and organic. The combination of such a highly efficient system with the premium, organic nature of the consumable output makes for an excellent economic opportunity for individuals and businesses alike.

In addition to being a great commercial and economic opportunity, automated aquaponics is also great for the environment, as it mirrors and develops relationships in ecosystems that occur naturally in the wild [1]. Aquaponic systems are highly efficient with water because the only water that leaves the system is lost due to evaporation. When a system is built outdoors or in a greenhouse that allows sunlight, systems become even more efficient by utilizing freely available light for plant growth. Furthermore, since, with the help of beneficial bacteria, the vegetables are filtering the water extraordinarily well, fish can be more heavily stocked and bred without interfering with their health. Similarly, the vegetables can be planted closer together than in a traditional garden because the competition for nutrients between plants is minimal. Competition is low because the abundant fish waste is continuously being converted to nutrients and introduced into the water all around the grow bed [1].

ORIGINALLY PROPOSED DESIGN REQUIREMENTS

Sylvia Bernstein, a trusted source among aquaponic hobbyists, wrote Aquaponic Gardening in 2011 to document effective methods to create and maintain an optimal aquaponic ecosystem. [8] [1]. Her work provided the foundation of knowledge used to design and build the biological components of this project including the grow bed, fish tank, and water circulation system. The originally proposed design requirements, listed below in Table 1, were generated by combining the recommended system design from Bernstein's guide with the desires of our end user.

Growth Optimization	The system must output sufficient light from LEDs to maintain plants
	The system must be able to provide 200 total gallons for aquaculture and hydroponics (Divided 1:1) [1]
	The Tilapia must actively breed and maintain their population independently
	The water level must only decrease due to evaporation
System Specifications	The system water level must not decrease beyond 15% of the tank capacity
	The fish feeding mechanism must output feed between 1% and 1.5% of the total fish weight [9]
	The pH stay at a level between 6 and 9 [9]
	The temperature must be maintained at a level between 79°F and 87°F if tilapia are kept [9]
User Alert	The LEDs must indicate whether system maintenance is required
	The LCD display must display all measured data in real time

Table 1.Originally Proposed Design Requirements

The system overview included above was created to satisfy the requirements listed above in Table 1. As seen in the table, the design requirements are divided into three distinct groups. The design requirements under the 'growth optimization' heading were created to ensure the healthy growth of the fish and plants within the system. Additionally, these requirements, such as the Tilapia breeding requirement, were created to ensure that the system is self-sustaining and providing a substantial amount of edible output.

The ‘system specifications’ related design requirements were created to ensure that the system operates optimally. These specifications were used to define when the user interface should alert the system manager to any potential problems. Finally, the ‘user alert’ related design requirements were created to ensure that system users were alerted to potential problems through the appropriate user interface medium. These requirements were defined by customer desires.

After combining the instructions offered in Bernstein’s book with the design goals for this project, the materials list for this project was generated. Since Tilapia will be cultivated, and a significant amount of food must be produced by the system, Rubbermaid’s 100 gallon stock tank was used as the fish tank [10]. As specified in Bernstein’s guide, a 1:1 fish tank capacity to grow bed size was used. Due to the unavailability of inexpensive grow bed tables, it will be easiest to fabricate a custom table to the exact specifications desired. Connecting the grow bed table and the fish tank will be a system of PVC piping, as well as a pump and pump triggering mechanism. These components constitute the biological system [1].

ORIGINALLY PROPOSED ELECTRONIC DESIGN REQUIREMENTS

The originally proposed electronic system design requirements are included below in Table 2. Most of these requirements were actually consequences of design choices that were made and therefore not a suitable means by which the final system could be evaluated. For this reason, and due to changes resulting from the evaluation of the system as the project progressed, the design requirements were later rewritten and are included below. For the sake of completeness, the original design requirements in Table 1 above and Table 2 below were preserved.

Specific Electronic Design Requirements	The system (excluding off the shelf components) must operate on +5V
	The PIC must communicate with the user interface using an I^2C bus
	The various sensors will be read using the digital or analog (when appropriate) channels of the PIC
	Power must be supplied to all components (excluding off the shelf components) using an AC to DC converter capable of supplying the appropriate current.
	The system must have a 'sleep mode' capability in which the user interface and PIC move into a low power mode. This low power mode will be interrupted if the timer indicates action is required or if a button on the user interface is pushed.

Table 2.Originally Proposed Electronic Design Requirements

Table 2 above shows the originally proposed electronic system design requirements for the project. In order to allow for the simple addition of peripheral components in the future, an I2C bus was implemented. Currently, this I2C bus only handles the communication between the PIC and the user interface display. The data displayed on that user interface comes from sensor information read in using the analog input channels on the PIC as well as the serial port not being used for I2C. Additionally, the ability for the microcontroller to go into a sleep mode for power saving was listed.

In addition to the electronic monitoring system, the lighting system was also designed specifically for this project. The lighting was designed to incorporate the appropriate amount of blue and red light to optimize plant root growth, leaf development, and flowering [11]. Three watt LEDs were selected because, with the appropriate heat sink and drive circuitry, they provide maximal light output while creating a manageable amount of heat. Further, the drive circuitry was created to interface with the 12V DC power supply and output a constant current into the LEDs. Eventually, for research purposes, dimming capabilities and the ability to turn off either color will be added to the system based on an interest from corporate aquaponics farmers [12].

FINAL DESIGN REQUIREMENTS

Since the original design requirements were vague and did not naturally lead to a valid testing plan that could easily determine the overall success of the project, newer design requirements were generated, and are included in Table 3 below. They are very similar to the older requirements listed above, but instead provide numerical requirements for most of the criteria so that the success of the project could be tested.

The biological system requirement numbers were chosen as a parameter by which the system's capability to sustain a substantial amount of life over a reasonable period of time could be evaluated. The budget was determined after ascertaining the general cost of similar systems and calculating the funding that would be necessary to create this system with aggressive cost reductions. The sensor measurement resolution requirements were created by determining how much resolution would be needed to accurately detect errors within the parameters being monitored. The sleep mode requirement was added to allow for experimentation with power reduction. Lastly, the alert triggering requirements were added so that the microcontroller could alert users of issues before they become critical.

Biological System Requirements	At least 10 heads of lettuce growing
	At least 30 days since the last loss of life
Project Budget	System Cost: < $1000
Sensor Measurement Requirements	Temperature measurements to ±1℃
	Water level measurements to ~1cm
	Occupancy detection in room
	Light level detection to 1lux
Electronic Requirements	Microcontroller sleep mode
Alert Triggering	40ºC < Temperature < 75ºC 10 hrs. < Grow light on time < 16hrs. 10min. < Grow bed flood time < 20min.

Table 3.Final Design Requirements

DESIGN ALTERNATIVES

Various design choices were made when determining how to construct and evaluate the overall system. The biological system was designed first so that the electronic monitoring components could be specifically selected to suit the biology. The main source used for the design of the biological system was Sylvia Bernstein's Aquaponic Gardening because it is widely accepted as the definitive guide to creating and maintaining a successful aquaponic system. Further, Sylvia cited 45 references in the book, and is generally regarded as one of the premier leaders in the global aquaponic community.

The system was designed to offer 100 gallons worth of grow bed space at 12" deep, with a 100 gallon fish tank. This design utilized Bernstein's widely accepted 12" grow bed depth theory while also maintaining the 1:1 grow bed to fish tank volume ratio that she suggests for beginner aquaponic gardeners. For the grow bed, a basic flood and drain system was selected because it was the simplest, easiest to implement system that is commonly used among beginners in aquaponics. If the flood and drain setup ever needs to be changed, and a constant water level fish tank is desired, the plumbing system can always be redesigned after the project has been completed [1].

The system hardware was also generally chosen based on Sylvia Bernstein's recommendations. A 100 Gallon Rubbermaid stock tank was chosen as suggested in Bernstein's Aquaponic Gardening. The grow bed was custom built from wood, caulk, and FRP wall board, to keep the project within its $1,000 budget. The basic idea is to build a wooden exterior for support with sealed FRP wall board on the inside walls of the bed to keep the water from leaking [13]. The grow bed must be able to support the weight of the water as well as the grow bed media within which the plants will grow. Expanded clay pebbles were chosen as the grow bed media because they could be found at a relatively reasonable price from Organica in Albany, and are lightweight, ph neutral, and easy to handle. The first batch of expanded clay that was purchased floated when submerged, and had to be replaced by the manufacturer. HydroFarm sent the superior Hydro Corn product as a replacement, which is currently installed in the system. Lastly, goldfish were selected because they are inexpensive and create a great deal of waste that will be converted to nutrients for the plants. Additionally, they do not require a heater and can tolerate a wide range of conditions [1].

The electronic monitoring devices and hardware were also carefully selected to monitor system parameters that could lead to catastrophic failure if not properly responded to in the event of a malfunction. These parameters include grow bed water level, water temperature, LED light levels, and room occupancy. Having decided that these parameters are critical and must be monitored to ensure that the system does not fall into unreported critical failure, each electronic component was selected to ensure adequate monitoring at the lowest possible cost. To that end, the following decisions were made:

Parameter	Selection	Alternatives
Grow Bed Water Level	Ultrasonic Sensor	eTape Liquid Level Sensor – Not chosen because it was too small & expensive [14]
Water Temperature	Temperature sensor by Atlas Scientific	Couldn't find other food safe temperature sensors
LED Light Levels	Light sensor by Atlas Scientific	Couldn't find other food safe light level sensors
Room Occupancy	Infrared Occupancy Sensor	Other infrared sensors were more expensive [15]

Table 4.Electronic System Component Choices

The sensors selected, shown above in Table 4, were chosen because they offered the least expensive solution that was both food safe and met the minimum resolution requirement specified in the design requirement section. The lack of options in food safe monitoring solutions was a complication that caused difficulty in various stages of this project. Thankfully, Atlas Scientific produces high quality, waterproof, and food grade light and temperature sensors that can be submerged indefinitely. These sensors are used to monitor select parameters in this project, along with a PING ultrasonic sensor which does not come in contact with the biological components.

PRELIMENARY PROPOSED DESIGN

It was proposed that the electronic system will perform all monitoring, error correcting, and user alert actions necessary to maintain the unit's ecosystem. The user interface would consist of an LCD display, pushbuttons, and LED indicators. The user interface display from Matrix Orbital was selected because their product fit into the 5.25" bay of a PC tower [16]. The empty PC tower was also set to be used to house other relevant electronic control equipment if necessary. Additionally, plans for constructing a grow light made up of 3W LEDs were created. The system's sensors would interface with the microcontrollers while the grow lights would be connected to the rest of the electronic system. The only system component not controlled and monitored by the microcontroller would be the pump which, due to its line voltage and high current requirements, was controlled safely and externally by 'off the shelf' products. All initially planned major electronic subsystems being used in the project are shown, with their connections, in Figure 1 below. These systems are explained in detail in the final design and implementation section included below.

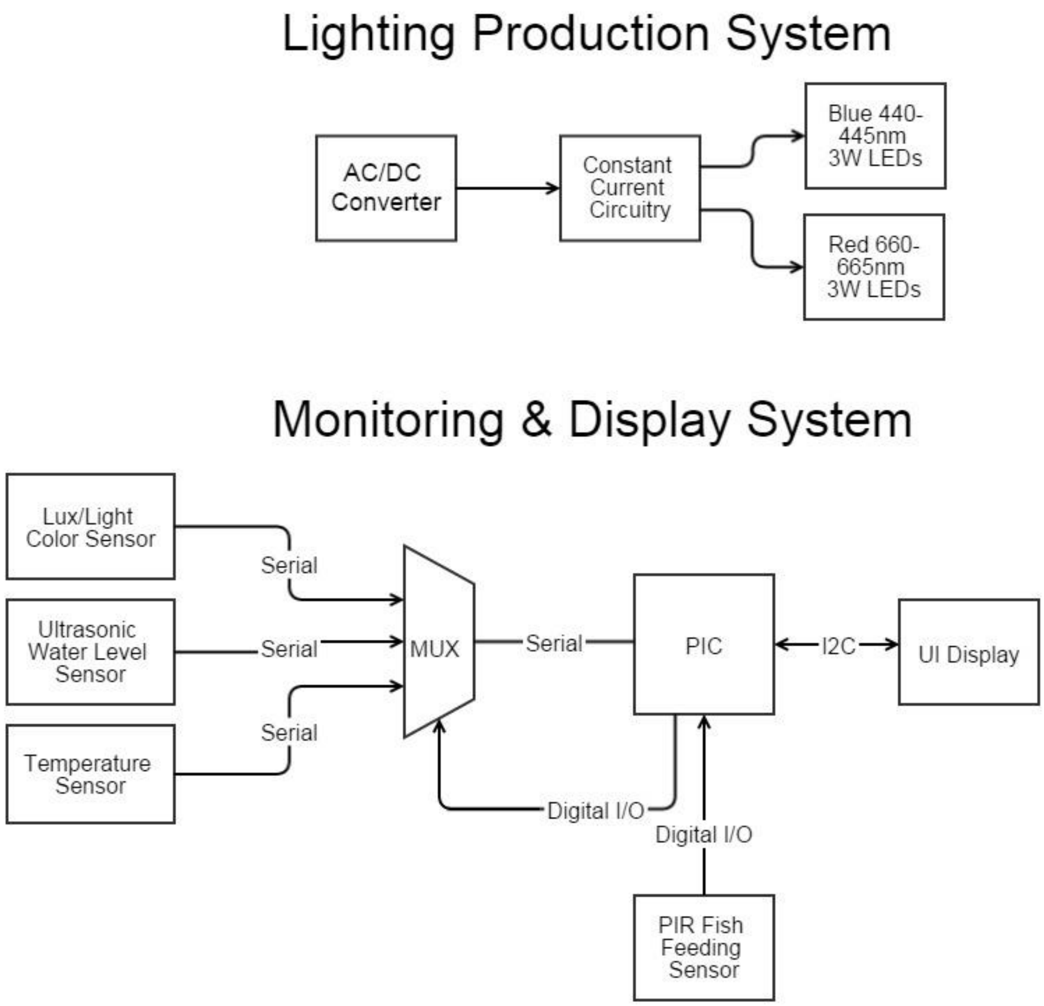

Figure 1.Originally Proposed Electronic System Block Diagram

FINAL DESIGN AND IMPLEMENTATION

The project was presented on February 28th, 2015 to an audience of Union College faculty, students, and members of industry. At that time, the grow bed was filled with sixty-two healthy heads of lettuce, the fish tank was filled with twenty-five healthy fish, and the electronic system was monitoring the temperature and the water level of the grow bed. The user interface was displaying temperature and water level characteristics, while the alarm was hooked up to a switch for demonstration purposes. Further, there was an issue with the interrupt service routine, so the software was changed to simply take measurements. Figure 2 below shows the overall system.

Figure 2.Overall System

The biological system, at the time of the presentation, was in full working condition. The grow bed was filled with twelve inches of expanded clay pebbles and composting worms. The composting worms were added to help break down the solid fish waste, and to decompose dead plant roots. The grow bed was flooded every fifteen minutes with water from the fish tank which, with the help of beneficial bacteria and the composting worms, provided the sixty-two heads of lettuce with an abundance of nutrients. The grow light, which provided the lettuce with energy for photosynthesis, was also fully functional at the time of the presentation. The grow light LEDs operated at a ratio of 16 hours of on time to 8 hours of off time. Lastly, the fish tank was only filled with twenty-five comet goldfish due to a large number of deaths in the fish tank that occurred over the college's winter vacation. The cause of death was likely overfeeding as the automatic fish feeder wasn't properly adjusted before the vacation.

Figure 3.Grow Bed Materials

Figure 4.Goldfish

The electronic system was designed to meet the specifications discussed above with special attention paid to ensuring that all of the electronic components and sensors that may come into connect with the grow bed were properly sealed and food grade. The final electronic hardware included in the system were a user interface with LCD display and push buttons, custom grow LEDs, a temperature sensor, an ultrasonic water level sensor, a light level sensor, and voltage conversion devices. Each of these system components can be seen abstracted into a block diagram in Figure 5 below.

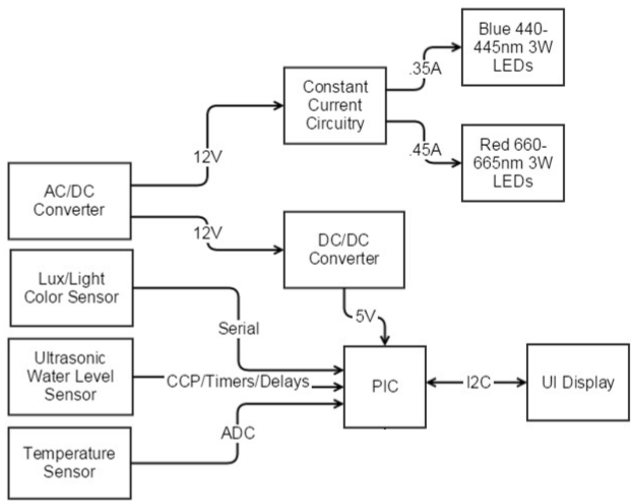

Figure 5.Electronic Block Diagram

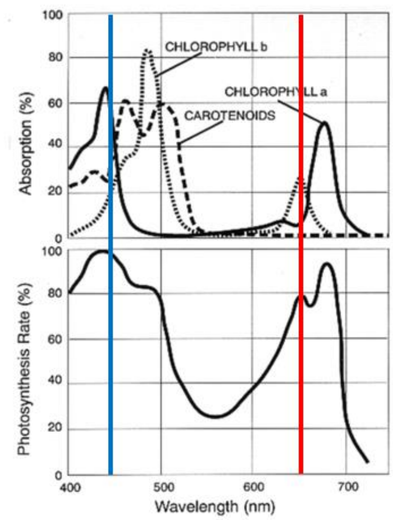

Figure 6.Photosynthesis vs Wavelength

The wavelength of the grow LEDs was specifically chosen to maximize photosynthesis. A graph comparing photosynthesis rate per unit wavelength is shown above in Figure 6. The two peaks of photosynthesis were matched to wavelengths at which LEDs are commonly manufactured. These wavelengths were a 445nm blue light and a 665nm red light. The aforementioned light level and temperature sensors were purchased from Atlas Scientific as the company produces food grade sensors that can be submerged and operated in liquids indefinitely without maintenance. The water level sensor consists of a PING ultrasonic sensor from Parallax mounted at the top of the grow bed looking down into a pipe where the water level can be seen. The data from each of these sensors is displayed on the Matrix Orbital interface. Additionally, the voltage conversion hardware consists of an AC/DC converter to change the line voltage to 12VDC. That 12VDC is used by the LEDs and also fed into a DC/DC voltage converter which steps the voltage down further to 5V. The remainder of the circuit components all operate on the 5V produced by the step down converter.

As previously mentioned, the communication between the various electronic components of this system was complicated by the unique demands of the aquaponic environment. Since all of the sensors had to be food grade or have the ability to operate without coming into contact with the biological system, there wasn't much variety among the few sensor components that fit our needs and the best available component had to be selected. As a result, most of the components communicated with the microcontroller over a different language. The temperature sensor required the use of an ADC, the light sensor communicates over serial, the water level sensor requires capture compare operations with delays, and the user interface communicates over I2C (Inter-integrated circuit) protocol. The electronic system development time was greatly increased as a result of having to setup and execute communication with each instrument with the PIC. Consequently, the occupancy sensor and light level sensor are not yet fully operational.

The goal was to implement the software, written in C, so that the system would sit in a while loop and wait for an interrupt to occur indicating that either it was time to take measurements from the sensors or it was time to turn on the user interface display as occupancy had been detected around the system. As a result of all of the different types of communication protocols and difficulties in implementing the interrupts together, the system would frequently reset and encounter an error when this approach was used. Instead the code was written, and is included in the appendix, so that measurements are being taken and the display is being updated within the main loop. An overview of the code is show in steps below.

1) Pragmas setup basic microcontroller settings and functions (i.e., oscillator speed)

2) A function called 'configure' then included, and it sets up all necessary registers for the communication and input sampling requirements of the system

3) Functions to execute the I2C communication and sensor readings are then included

4) The main function, listed next, turns on the display and then sits in a loop which reads the sensors and sends the result, over I2C, to the user interface

5) The interrupt service routine will be included here when functional, at which time the operations of the main loop will be moved to the interrupt service routine

PERFORMANCE ESTIMATES AND RESULTS

As a monitoring system, it is difficult to estimate how the electronics will perform except to expect that they will function to the specified accuracy and communicate any critical system errors to system users. Biologically, it was estimated that the system should sustain floral and faunal life, as a result of the aquaponic ecosystem combined with electronic monitoring and lights, for extended periods of time. With respect to the cost, initial projections indicated that the entire system would cost about $1000, and thus a budget of $1000 was created at the onset of the project. By the project's completion, the budget was met with a final system cost of $943.42.

The system met most of the final biological and electronic design requirements by the end of the winter term. The biological requirements specified that the system must sustain the life of the fish in the tank and at least 10 heads of lettuce so that, by the final test, there has not been a loss of life in at least four weeks. By the final testing date, there were 63 healthy heads of lettuce growing and it had been 53 days since the last fish had been lost.

The final revision of the electronic design requirements specified that the system should take and relay temperature measurements to 1ºC, water level measurements to 1cm, and light level measurements to 1lux. Further, the system should detect nearby occupancy so that the display can be activated, and enter a sleep mode when not displaying information or taking measurements to save power. The only two requirements that were not met were the detection of light level to 1 lux and the ability of the microcontroller to enter into a sleep mode when not in use. These are, however, simply software modifications that can be made by taking the time to determine how the microcontroller registers need to be initialized in order to meet each requirement. In the future, the system will be modified to meet all requirements.

PRODUCTION SCHEDULE

In order to better facilitate the completion of the automated aquaponics project, the schedule shown in Table 5 below was created. The only major fault in the system production was the choice to include gravel instead of spending the extra money to purchase and add expanded clay pebbles in the system. Almost all sources stated that gravel should not be used in an aquaponic system due to the high possibility that limestone in the rocks may significantly raise the pH of the system [1]. As a result of ignoring this advice, production was delayed after the purchase, washing, and installation of the grow bed material had to be performed twice. Besides the grow bed media issue, all other system choices were generally acceptable and the project production schedule included below was followed.

Dates	Objectives
6/1/14 – 8/7/14	• Aquaponic system built was setup in N101 • Tank cycling and plant growth began • Electronic parts were ordered
8/7/14 – 11/26/14	• Goldfish were purchased • Electronic programming & testing began • System assembly was completed • Website Created
11/26/14 – 1/5/15	• System debugging begins • Electronic system installation begins • Trials with various vegetables and fruits begin
1/5/15 – 3/19/15	• Begin final report

Table 5.Project Schedule

This project was executed with the understanding that the resulting system could not be feasibly produced in industry and sold as a commercial project. The system is too large and complex with an accompanying price point too high given the amount of risk associated putting up the capital necessary to produce this product commercially. Thus, a commercial production schedule cannot be created. Instead, this project was built to facilitate research in targeted LED lighting for aquaponic systems and to show that electronic systems could be implemented to monitor a commercial aquaponic system. With electronic monitoring, aquaponics becomes an efficient and feasible production method for large companies with which indoor food production can be implemented.

COST ANALYSIS

The final cost of the fully assembled system is listed in table 5 below and meets the budget requirement of having a total system cost of less than $1000. If another system were to be constructed, the Matrix Orbital display should be discarded in favor of simply sending the data to a webpage or smartphone app. The other expenses listed however, especially those that comprise the biological system, were all critical to the successful functioning of the system. The electronic expenses were also worthwhile and necessary as their initial cost will be offset by the money and time that they will save by providing system monitoring and error detection.

Item	Quantity	Unit Price	Total Price
Rubbermaid 100 Gallon Stock Tank	1	$65.00	$65.00
GFCI Outlet Adapter	1	$20.00	$20.00
Matrix Orbital UI with Mount	1	$88.00	$88.00
FRP Wall Panel White	2	$33.45	$66.90
Hydrofarm 250GPH Pond Pump	1	$20.00	$20.00
Eco-Bond Farm Safe Sealant	2	$9.00	$18.00
Plywood	1	$30.00	$30.00
Goldfish	37	$0.32	$11.84
Screws	1	$10.00	$10.00
PVC Bulkhead	1	$12.00	$12.00
Automatic Fish Feeder	1	$30.00	$30.00
PVC and Piping	1	$60.00	$60.00
Breadboard Cable	1	$16.89	$16.89
LED Lighting	1	$23.58	$23.58
Hydroton	1	$330.00	$330.00
Color Detector Sensor	1	$40.00	$40.00
Temperature Probe	1	$10.00	$10.00
Motion Sensor (for fish feeder)	1	$9.95	$9.95
Ultrasonic Water Level Sensor	1	$32.99	$32.99
Thermal Adhesive	1	$8.00	$8.00
Resistors, Capacitors, DC Converter	1	$15.27	$15.27
LEDs	1	$25.00	$25.00
		Total	$943.42

Table 6.Expenditures

USER'S MANUAL

This system, and future systems, could be constructed and operated according to the following procedure:

Construction

- Purchase all components listed in Table 5.

- Create the grow bed with the plywood and 2x4s. See [13].

- The grow bed should be lined with the FRP wall board and sealed using food safe caulk. A hole must be drilled and a bulkhead must be installed in the bottom of the grow bed so that the water can drain back into the fish tank.

- The fish tank should be placed under the grow bed table, and a water pump should be added to pump the water from the fish tank into the grow bed. Irrigation may be installed if desired. The pump must be either attached to a timer to create flooding and draining cycles, or an automatic siphon can be used. More information about automatic siphon construction can be found in Sylvia Bernstein's Aquaponic Gardening [1].

- The grow bed should be filled with expanded clay pebbles and a light should be installed above the grow bed to provide the plants with an adequate light source. Plants and fish may then be added to start the aquaponic food production cycle.

- Lastly, the automatic fish feeder and electronic monitoring should be installed so that users are alerted of any errors that may arise. The electronic monitoring system could be implemented in a variety of ways. An overview of the electronic system used in this project can be found in the 'Final Design and Implementation' section of this report.

Operation

The system is designed to operate for extended periods of time without any user intervention. Maintenance should only be necessary under one of the following conditions:

- The fish feeder has run out of food. To remedy: refill the fish feeder.
- The automatic siphon is not properly draining water because plant roots and other solids have clogged the automatic siphon. To remedy: rotate the automatic siphon's gravel guard to break up plant root and solid accumulations.
- The system chemistry is imbalanced. To remedy: purchase an API test kit and take appropriate steps after testing.
- An error has been detected by the electronic system. To remedy: The detected error should be fixed (e.g., plug the water leak, replace the broken fish tank heater, etc.).

DISCUSSION, CONCLUSIONS, AND RECOMMENDATIONS

Aquaponics has not been adopted on a large scale due to a number of factors including the complexity and difficulty with which these systems must be constructed and monitored. It was the purpose of this project to build a system that would provide a test bed for spectrum targeted LED grow light creation and show that scalable aquaponic monitoring systems could be implemented commercially. It was therefore our goal to present beneficial research and proof of concept in the electronic aspects of aquaponics so that hobbyists and companies can continue to improve and expand upon their aquaponic endeavors.

This goal was reached and the system does produce vegetative growth in a sustainable ecosystem with spectrum targeted LED lighting and electronic monitoring and automatic alerts. There were, however, some mistakes made through the course of the project that should be improved upon if a similar project were to be attempted in the future. As mentioned above, the choice to use gravel as a cost saving measure despite many warnings found in preliminary research was a mistake that resulted in a great deal of wasted time and money. If this project were to be attempted again, expanded clay pebbles should be used from the onset. Additionally, to prevent nutrient deficiencies in the plants being grown in the grow bed, the fish tank should be adequately stock according to the ratios outlined in Bernstein's book, even if some of the population may have to be culled or sold later. Lastly, the electronic components should be selected with a specific attention paid to minimizing the overall development time of the system. Not enough focus was placed on the communication protocols used within the electronic subsystem and, as a result, each sensor and the display communicated in a different fashion. As discussed above, each communication protocol must be properly initialized and configured in order to work effectively on the microcontroller. Further, each of those components must operate together in both hardware and software, such as in an interrupt service routine, without interrupting the functional processes any other components. Communication and compatibility is still an area that requires attention in this project, and an area that will be continually examined and improved upon in the future.

Overall, this project met the goals of creating a functional aquaponic system capable of sustaining life while also implementing spectrum targeted LED grow lighting and electronic monitoring. Each of these systems is scalable and offers a significantly less expensive alternative to any commercially available grow lights or basic monitoring systems. This project will be continually modified and improved upon until presented at Union College's Steinmetz Day on May 8th, 2015. At that point, the project will be taken home with its creator, where experimentation and food production will continue.

REFERENCES

[1] S. Bernstein, Aquaponic Gardening, Gabriola Island: New Society Publishers, 2011.

[2] M. Saaid, N. Fadhil, M. Ali and M. Noor, "Automated indoor Aquaponic cultivation technique," *System Engineering and Technology (ICSET), 2013 IEEE 3rd International Conference on,* pp. pp.285,289, 2013.

[3] T.-C. Chia and C.-L. Lu, "Design and Implementation of the Microcontroller Control System for Vertical-Garden Applications," *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on,* pp. pp.139,141, 2011.

[4] Kijani Grows, "Buy Now," 2014. [Online]. Available: http://www.kijanigrows.com/buy_now/. [Accessed 12 5 2014].

[5] New York State Department of Agriculture & Markets, "Sanitary Regulations for Direct Marketing," New York State, [Online]. Available: http://www.agriculture.ny.gov/FS/industry/sanitary.html. [Accessed 13 5 2014].

[6] "Legal Issues," Tilapia Farming at Home, [Online]. Available: http://www.tilapiafarmingathome.com/Pages/LegalIssues.aspx. [Accessed 13 5 2014].

[7] New York State, "Licenses to Collect, Possess or Sell," Department of Environmental Conservation, 2014. [Online]. Available: http://www.dec.ny.gov/permits/28633.html. [Accessed 13 5 2014].

[8] Amazon, "Aquaponic Gardening: A Step-By-Step Guide to Raising Vegetables and Fish Together," [Online]. Available: http://www.amazon.com/Aquaponic-Gardening-Step-By-Step-Vegetables-Together/dp/086571701X/ref=sr_1_1?ie=UTF8&qid=1399944870&sr=8-1&keywords=aquaponic+gardening. [Accessed 12 5 2014].

[9] T. M. L. a. J. E. R. Dennis P. DeLong, "Tank Culture of Tilapia," 6 2009. [Online]. Available: http://www2.ca.uky.edu/wkrec/tilapiatankculture.pdf. [Accessed 21 5 2014].

[10] Tractor Supply Co., "Rubbermaid® Structural Foam Stock Tanks, 100 gal. Capacity," [Online]. Available: http://www.tractorsupply.com/en/store/rubbermaidreg%3B-structural-foam-stock-tanks-100-gal--capacity. [Accessed 11 5 2014].

[11] NASA, "Improving Spinach, Radish, and Lettuce Growth under Red Lightemitting," HortScience, Neil C. Yorio, Gregory D. Goins, and Hollie R. Kagie, Raymond M. Wheeler and John C. Sager.

[12] C. Currin, Interviewee, *Capital Aquaponics CEO.* [Interview]. 17 11 2014.

[13] E. Maelzer, "YouTube," 17 1 2010. [Online]. Available: http://www.youtube.com/watch?v=6eOQOm5k_dE. [Accessed 23 11 2014].

[14] Sparkfun Electronics, "Liquid Level Sensor - 8"," Sparkfun, [Online]. Available: https://www.sparkfun.com/products/10221. [Accessed 14 3 2015].

[15] Amazon, "Sensky BS019B Dc 12v Corridor Toilets Human Body Infrared Detector Motion Switch Wired Ceiling,pir Body Infrared Induction Motion Sensor Light Switch," Sensky, [Online]. Available: http://www.amazon.com/Corridor-Toilets-Infrared-Detector-Induction/dp/B00L589HJG/ref=sr_1_12?ie=UTF8&qid=1426380726&sr=8-12&keywords=occupancy+sensor. [Accessed 14 3 2015].

[16] MatrixOrbital, "BLK204-7T-1U-BK-WB," [Online]. Available: http://www.matrixorbital.com/PC-Bay-Inserts-Serial-Character-PC-Bay-Inserts/c45_21/p856/BLK204-7T-1U-BK-WB/product_info.html. [Accessed 12 5 2104].

[17] MatrixOrbital, "LK204-7T-1U-WB," 12 3 2014. [Online]. Available: http://www.matrixorbital.ca/manuals/LK_Series/LK204-7T-1U/LK204-7T-1U%20%28Rev2.0%29.pdf. [Accessed 8 5 2014].

[18] G. Branwyn, "Lost Knowledge:Wire Wrapping," 27 7 2009. [Online]. Available: http://makezine.com/2009/07/27/lost-knowledge-wire-wrapping/. [Accessed 8 5 2014].

[19] G. D. G. H. R. K. R. M. W. a. J. C. S. Neil C. Yorio, "Improving Spinach, Radish, and Lettuce Growth under Red Light emitting," NASA.

APPENDIX

```
/*******************************************************************

        Union College Automated Aquaponics Program

Last Revised: 10/7/13
Authors: Carson Miller
Written for: PIC18F4525 (Current Version)

*******************************************************************/

#define INPUT 1
#define OUTPUT 0

#define _XTAL_FREQ 4000000                    //Used by the XC8 delay_ms(x) macro

// PIC18F25K22 Configuration Bit Settings

#include <xc.h>              //Includes PIC hardware mapping
#include "GenericTypeDefs.h"    //Includes standard variable types

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

// CONFIG1H
#pragma config OSC = INTIO67     // Oscillator Selection bits (Internal oscillator block, CLKOUT function on
RA6, port function on RA7)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF       // Internal/External Oscillator Switchover bit (Oscillator Switchover mode
disabled)

// CONFIG2L
#pragma config PWRT = OFF       // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = SBORDIS  // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only
(SBOREN is disabled))
#pragma config BORV = 3         // Brown Out Reset Voltage bits (Minimum setting)

// CONFIG2H
#pragma config WDT = OFF        // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTPS = 32768    // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = PORTBE  // CCP2 MUX bit (CCP2 input/output is multiplexed with RB3)
#pragma config PBADEN = ON      // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input
channels on Reset)
#pragma config LPT1OSC = OFF    // Low-Power Timer1 Oscillator Enable bit (Timer1 configured for higher
power operation)
#pragma config MCLRE = ON       // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON      // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = OFF        // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config XINST = OFF      // Extended Instruction Set Enable bit (Instruction set extension and Indexed
Addressing mode disabled (Legacy mode))
```

```
// CONFIG5L
#pragma config CP0 = OFF        // Code Protection bit (Block 0 (000800-003FFFh) not code-protected)
#pragma config CP1 = OFF        // Code Protection bit (Block 1 (004000-007FFFh) not code-protected)
#pragma config CP2 = OFF        // Code Protection bit (Block 2 (008000-00BFFFh) not code-protected)

// CONFIG5H
#pragma config CPB = OFF        // Boot Block Code Protection bit (Boot block (000000-0007FFh) not code-
protected)
#pragma config CPD = OFF        // Data EEPROM Code Protection bit (Data EEPROM not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF       // Write Protection bit (Block 0 (000800-003FFFh) not write-protected)
#pragma config WRT1 = OFF       // Write Protection bit (Block 1 (004000-007FFFh) not write-protected)
#pragma config WRT2 = OFF       // Write Protection bit (Block 2 (008000-00BFFFh) not write-protected)

// CONFIG6H
#pragma config WRTC = OFF       // Configuration Register Write Protection bit (Configuration registers (300000-
3000FFh) not write-protected)
#pragma config WRTB = OFF       // Boot Block Write Protection bit (Boot Block (000000-0007FFh) not write-
protected)
#pragma config WRTD = OFF       // Data EEPROM Write Protection bit (Data EEPROM not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF      // Table Read Protection bit (Block 0 (000800-003FFFh) not protected from table
reads executed in other blocks)
#pragma config EBTR1 = OFF      // Table Read Protection bit (Block 1 (004000-007FFFh) not protected from table
reads executed in other blocks)
#pragma config EBTR2 = OFF      // Table Read Protection bit (Block 2 (008000-00BFFFh) not protected from table
reads executed in other blocks)

// CONFIG7H
#pragma config EBTRB = OFF      // Boot Block Table Read Protection bit (Boot Block (000000-0007FFh) not
protected from table reads executed in other blocks)

//Variables

unsigned int i = 0;
unsigned int j = 0;
unsigned int k = 0;
unsigned int timer = 0;
unsigned int sensorInput = 0;

//Related to water temperature
float waterTemp = 0;
unsigned int waterTempInt = 0;

unsigned int floodNum1 = 0;
unsigned int floodNum2 = 0;

char floodChar1 = '8';
char floodChar2 = '0';

// Related to water level
unsigned long t1; // 16 bit number, 0 - 65,535
unsigned long t2;
```

```
int startH;
int startL;
int endL;
int endH;
int shiftedStartH;
int shiftedEndH;
int startTime;
int endTime;
int echoTime;
char waterLevel1;
char waterLevel2;

unsigned int dummy = 0;
unsigned int dummy2 = 0;


void configure()
{
    //Oscillator Setup
    OSCCONbits.IRCF = 110;    //Sets oscillator to 4MHz

    //I2C Configuration
    SSPADD = 0x33; //19.2kHz Baud clock @4MHz
    SSPCON1bits.SSPM = 1000; //Master Mode
    SSPCON1bits.SSPEN = 1;   //Enables serial port and configures pins

    //Transistor switch control code
    TRISDbits.TRISD2 = OUTPUT;  //Sets RD2 as output (Switching transistor control)
    LATDbits.LATD2 = 0;         //Sets high

    //Temperature sensor control code
    TRISBbits.TRISB4 = OUTPUT;  //Sets RB3 as output (Switching temp sense control)
    TRISAbits.TRISA0 = INPUT;    //Sets AN0 as input
    LATBbits.LATB4 = 0;         //Sets low

    //Water level sensor setup
    LATBbits.LATB3 = 0;         //Sets low

    // Configure timer1 register
    T1CON = 0b00110001; // Enable timer1, Fosc/4 input
    CCP1CON = 0x04; // Capture every falling-edge
    TRISCbits.TRISC2 = 1;   //Enable CCP Input
    TRISDbits.TRISD3 = 0;   //Set PNP FET channel as output
    LATDbits.LATD3 = 1;     //Set Transistor high

    //Analog input configuration
    ADCON0bits.ADON = 1;      //Turns on the ADC
    ADCON1bits.VCFG1 = 0;     //Sets (-) voltage reference as ground
    ADCON1bits.VCFG0 = 0;     //Sets (+) voltage reference as +5V
    //Note PBADEN set to 0 in configuration bits to allow RB4:RB0 to be digital I/O on reset (datasheet p. 253)
    ADCON2bits.ADFM = 1;      //Right justify the result of the A/D conversion
    ADCON2bits.ACQT = 000;    //
    ADCON1bits.PCFG = 1101;     //Enables AN0 and AN1

    //EUSART Serial Communication Port Setup
    TRISCbits.TRISC7 = 1;
```

```
    TRISCbits.TRISC6 = 1;
    TXSTAbits.TX9 = 0;          //Select 8 bit transmission
    TXSTAbits.TXEN = 1;         //Enable transmission
    TXSTAbits.SYNC = 0;         //Sets EUSART to asynchronous mode
    TXSTAbits.BRGH = 1;         //Sets EUSART to high speed transmission
    RCSTAbits.SPEN = 1;         //Enables the serial port
    RCSTAbits.CREN = 1;         //Enables receiver
    BAUDCONbits.RXDTP = 1;      //Incoming data is inverted
    BAUDCONbits.BRG16 = 0;      //8 bit baud rate generator used
    BAUDCONbits.WUE = 1;        //Continuous sampling of the RX pin

    SPBRG = 0x05;
}

void I2CInit(void)
{
        TRISC3 = 1;      /* SDA and SCL as input pin */
        TRISC4 = 1;      /* these pins can be configured either i/p or o/p */
        SSPSTAT |= 0x80; /* Slew rate disabled */
    SSPCON1bits.SSPEN = 1;
    SSPCON1bits.SSPM = 1000; //Master Mode
    SSPCON1bits.SSPEN = 1;   //Enables serial port and configures pins
    SSPADD = 0x33; //19.2kHz Baud clock @4MHz
}

void I2CStart()
{
        SSPCON2bits.SEN = 1;        /* Start condition enabled */
        while(SEN);      /* automatically cleared by hardware */
            /* wait for start condition to finish */
}

void I2CStop()
{
        SSPCON2bits.PEN = 1;        /* Stop condition enabled */
        while(PEN);      /* Wait for stop condition to finish */
            /* PEN automatically cleared by hardware */
}

void I2CRestart()
{
        RSEN = 1;        /* Repeated start enabled */
        while(RSEN);     /* wait for condition to finish */
}

void I2CAck()
{
        ACKDT = 0;       /* Acknowledge data bit, 0 = ACK */
        ACKEN = 1;       /* Ack data enabled */
        while(ACKEN);    /* wait for ack data to send on bus */
}

void I2CNak()
{
        ACKDT = 1;       /* Acknowledge data bit, 1 = NAK */
        ACKEN = 1;       /* Ack data enabled */
```

```
        while(ACKEN);    /* wait for ack data to send on bus */
}

void I2CWait()
{
        while ((SSPCON2 & 0x1F ) || ( SSPSTAT & 0x04 ) );
   /* wait for any pending transfer */
}

void I2CSend(unsigned char dat)
{
        SSPBUF = dat;    /* Move data to SSPBUF */
        while(BF);       /* wait till complete data is sent from buffer */
        I2CWait();       /* wait for any pending transfer */
}

unsigned char I2CRead(void)
{
        unsigned char temp;
/* Reception works if transfer is initiated in read mode */
        RCEN = 1;        /* Enable data reception */
        while(!BF);      /* wait for buffer full */
        temp = SSPBUF;   /* Read serial buffer and store in temp register */
        I2CWait();       /* wait to check any pending transfer */
        return temp;     /* Return the read data from bus */
}

UINT16 get_adc(UINT8 channel)
{

  char x;     // local counter
  ADCON0bits.CHS = channel;       // select ADC channel
  NOP();                          // wait a little for acquisition.
  NOP();
  NOP();
  NOP();
  NOP();                          // wait a little for acquisition.
  NOP();
  NOP();
  NOP();
  GO_nDONE = 1;                   // Start the conversion
  // wait for conversion
  for(x=200;x;x--)                // avoid getting stuck
    if(GO_nDONE == 0) break;      // exit as soon as results are ready
   return (ADRESH << 8 | ADRESL);  // return the result
}

void findLevel()
{
   startH=TMR1H;
   startL=TMR1L;
   LATDbits.LATD3 = 0;
   __delay_us(5);
   LATDbits.LATD3 = 1;
   __delay_ms(20);
```

```
endL = CCPR1L;
endH = CCPR1H;

//Bitwise shift to prepare for bitwise OR
shiftedEndH = endH << 8;
shiftedStartH = startH << 8;

//Bitwise OR

startTime = startL^shiftedStartH;
endTime = endL^shiftedEndH;

echoTime = endTime-startTime;

if (echoTime<150)
{
   waterLevel1 = '9';
   waterLevel2 = '9';
}
else if (echoTime<165)
{
   waterLevel1 = '9';
   waterLevel2 = '0';
}
else if (echoTime<180)
{
   waterLevel1 = '8';
   waterLevel2 = '0';
}
else if (echoTime<195)
{
   waterLevel1 = '7';
   waterLevel2 = '0';
}
else if (echoTime<210)
{
   waterLevel1 = '6';
   waterLevel2 = '0';
}
else if (echoTime<225)
{
   waterLevel1 = '5';
   waterLevel2 = '0';
}
else if (echoTime<245)
{
   waterLevel1 = '4';
   waterLevel2 = '0';
}
else if (echoTime<260)
{
   waterLevel1 = '3';
   waterLevel2 = '0';
}
else if (echoTime<275)
{
```

```
        waterLevel1 = '2';
        waterLevel2 = '0';
    }
    else if (echoTime<290)
    {
        waterLevel1 = '1';
        waterLevel2 = '0';
    }
    else
    {
        waterLevel1 = '0';
        waterLevel2 = '0';
    }

}

void tempRead()
{
    LATBbits.LATB4 = 1;         //Turns on
    __delay_us(1500);
    waterTemp = get_adc(0); //Changed to 0
    LATBbits.LATB4 = 0;         //Turns off
    waterTemp = ((waterTemp*5000)/1004);        //Convert to mV
    waterTemp = ((waterTemp*0.0512)-20.5128);   //Convert to deg C
    waterTemp = (((waterTemp*9)/5)+32);           //Convert to deg F
    waterTempInt = waterTemp;
    floodNum1 = waterTempInt/10;
    floodNum2 = waterTempInt-(floodNum1*10);

    //Need to convert to char

    if (floodNum1 == 6)
        floodChar1 = '6';
    else if (floodNum1 == 5)
        floodChar1 = '5';
    else if (floodNum1 == 7)
        floodChar1 = '7';

    if (floodNum2 == 0)
        floodChar2 = '0';
    else if (floodNum2 == 1)
        floodChar2 = '1';
    else if (floodNum2 == 2)
        floodChar2 = '2';
    else if (floodNum2 == 3)
        floodChar2 = '3';
    else if (floodNum2 == 4)
        floodChar2 = '4';
    else if (floodNum2 == 5)
        floodChar2 = '5';
    else if (floodNum2 == 6)
        floodChar2 = '6';
    else if (floodNum2 == 7)
        floodChar2 = '7';
    else if (floodNum2 == 8)
        floodChar2 = '8';
```

```
    else
        floodChar2 = '9';


}

void updateScreen()
{
//This loop below spits out I2C because we're not waiting for the display
        //to initalize and turn on.  We could just send the I2C once but the display
        //would have to be initialized for that to work and thus we would want to
        //set our own startup screen so that the matrix orbital screen doesn't pop
        //up in between durign initalization

        for(j=0;j<1; j++)
        {
            I2CInit();
            I2CStart();
            I2CSend(0x50);

            I2CSend(254);

            I2CSend(88);
            //This loop sends spaces
            //for(i=0;i<20; i++)
            //{
            //I2CSend(' ');
            //}

            //First Line

            I2CSend(' ');
            I2CSend(' ');
            I2CSend('U');
            I2CSend('N');
            I2CSend('I');
            I2CSend('O');
            I2CSend('N');
            I2CSend(' ');
            I2CSend('A');
            I2CSend('Q');
            I2CSend('U');
            I2CSend('A');
            I2CSend('P');
            I2CSend('O');
            I2CSend('N');
            I2CSend('I');
            I2CSend('C');
            I2CSend('S');
            I2CSend(' ');
            I2CSend(' ');
            //I2CSend(' ');

            //Second Line

            I2CSend(' ');
```

```
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');
I2CSend(' ');

//Third Line

I2CSend(' ');
I2CSend(' ');
I2CSend('W');
I2CSend('A');
I2CSend('T');
I2CSend('E');
I2CSend('R');
I2CSend(' ');
I2CSend('L');
I2CSend('E');
I2CSend('V');
I2CSend('E');
I2CSend('L');
I2CSend('=');
I2CSend(' ');
I2CSend(waterLevel1);
I2CSend('.');
I2CSend(waterLevel2);
I2CSend(' ');
I2CSend(' ');

//Fourth Line

I2CSend(' ');
I2CSend(' ');
I2CSend('W');
I2CSend('A');
I2CSend('T');
I2CSend('E');
I2CSend('R');
I2CSend(' ');
I2CSend('T');
I2CSend('E');
I2CSend('M');
```

```
            I2CSend('P');
            I2CSend('=');
            I2CSend(' ');
            I2CSend(floodChar1);
            I2CSend(floodChar2);
            I2CSend('F');
            I2CSend(' ');
            I2CSend(' ');

            I2CStop();
        }
}


void main()
{
    configure();

    LATDbits.LATD2 = 0; //Turns on display --- temporary

    while(1)
    {

        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);
        __delay_ms(100);




        tempRead();
        findLevel();
        updateScreen();

    }

}
```