EE175ABC Final Report

I.L.A.T.S

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

Project Team Member(s)	Mark Philip Logston, Ahmed El Gabaly, Moheb Amini
Date Submitted	4/24/2014
Section Professor	Ping Liang
Revision	Revision 10
URL of Project Wiki/Webpage	https://www.youtube.com/watch?v=8rx8W-GGqlw

Summary

This report presents I.L.A.T.S. Below you will find all the documentation that was used in the creation of I.L.A.T.S.

Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
1	First draft	Mark Logston, Moheb Amini, Ahmed El Gabaly	04/21/2014	
1.2	Edited Introduction	Mark Logston	05/14/2014	
1.3	Edited Design Considerations and Conclusion	Mark Logston	5/22/2014s	
1.4	Edited Exec. Summary and High Level Design	Moheb Amini	5/15/2014	
1.5	Edited Proj. Management, Acknowledgements, and References	Moheb Amini	5/22/2014	
1.6	Edited Low Level Design	Ahmed El Gabaly	5/20/2014	
1.7	Edited Experiment Design and test plan and Experiment Results and test report	Ahmed El Gabaly	5/22/2014	
1.8	Edited Introduction, Design Considerations, and Conclusion	Mark Logston	6/1/2014	
1.9	Edited Proj. Management, Acknowledgements, References, Exec. Summary, and High Level Design	Moheb Amini	6/2/2014	
2.0	Edited Low Level design, Experiment Design and test plan and Experimental results and test report	Ahmed Elgabaly	6/5/2014	
2.1	Fixed formatting, Spelling, and some grammar issues throughout whole report	Mark Logston	6/5/2014	

Table of Contents

1 Executive Summary

Our project, known as I.L.A.T.S. (Infrared Light Asset Tracking System), was developed with certain goals in mind for its completion. The main technical objective was to be able to track the location of two separate, movable asset transmitters within two rooms and one hallway receiver.

Our system was able to detect asset movement using an accelerometer, which outputs XYZ position values for computation. This goal is critical for knowing when to transmit the signal from the transmitters, each of which had their own unique signature in frequency. The transmission of the signal must have a range that includes eight feet of transmission (requirement) to the receiver. Eight feet transmission requirement is necessary in order for the receiver to receive the transmission at its maximum distance.

Once the signal is received there is a process in which the signal is analyzed in order to compute the signature of the asset passing through. Once the asset is identified it will then be updated onto an executable on the computer.

Our method was chosen carefully in order to ensure low power consumption. The key features of our proposed method includes: infrared signal transmission through LED on the asset, transmission occurs only when movement is sensed by the accelerometer, battery checker gauge on asset, FFT analysis of received signal, communication via Ethernet to online server, and executable GUI to view locations of assets.

After individually testing each feature until it was as accurate as possible we ended with a full system test. The results of testing our system were successful. Our system met every single one of our goals (requirements)

Our individual achievements are broken up within our individual tasks. Moheb successfully created movement-sensing transmission with the accelerometer and the FFT analysis of the received signal. Ahmed successfully created the online server and its communication with microcontroller through Ethernet protocol and implemented a battery checker on the asset transmitter. Mark successfully created the GUI that accessed the online server and setup a database for the rooms and assets. Through these individual achievements of key features within the project we were able to bring together the completion of all our goals and technical objectives.

2 Introduction

2.1 Design Objectives and System Overview

I.L.A.T.S is an object tracking system using infrared light. The main design objective of the system is to be able to track an object moving through a series of rooms using door frame mounted receivers. The tracking of the objects will be displayed on a GUI on a computer. When the objects are in motion an infrared LED will light up letting the receiver know the object's location. This project, light asset tracking is a low power solution to track objects moving throughout a building. The project is intended to be able to track objects moving throughout a hospital. To be able to accomplish this design, we have to be able to utilize signal processing techniques to extract the data being sent from the object, optics, embedded systems and telecommunications. The goal of this project is to be able to track two objects moving between two rooms and a hallway.

The design of our system is broken up into three subsystems: transmitter, receiver, and the GUI. The receiver based on information from the accelerometer communicate to the microcontroller that the object is moving. When this occurs the microcontroller will send pulses to the infrared LED at a specific frequency. Also on the transmitter there is a battery life checker. To check the battery the switch on the board will need to be turned on causing the RGB LED to turn on with a color representing the life of the battery. For the receiver, a phototransistor will receive the signal which will be sent to the first microcontroller. This microcontroller will calculate the FFT and send out what the frequency that was received to the second microcontroller. The second microcontroller would send the object frequency and object room number to a server. The GUI will process the data on the server and display the data.

The technical design requirements for our project are: being able to track two objects as they move throughout two rooms and a hallway, sense any object motion in x, y, and z coordinates using an accelerometer, transmit and receive a signal at least 8 feet away, compute the FFT of received signal to identify each signature have a server response time of at least 400 milliseconds, and have low power consumption on the transmitter side.
Responsibilities:

Moheb Amini:

- Communicate Infrared LED and Phototransistor
- Communicate Accelerometer and microcontroller
- Create the Circuits that are needed
- Create FFT analysis of incoming signals
- Design and build Transmitter PCB
- Design and build the door Mount

Ahmed El Gabaly:

- Set up the microcontroller of the Receiver to send the incoming data to Ethernet
- Communication from the Receiver microcontroller to the Server
- Create a Battery Life Checker
- Design and build the Receiver PCB (Figure 2.1)

Mark Logston:

- Create the GUI
- Create the Database
- Create the Circuit that are needed
- Align the Infrared sensors
- Help in creation of Transmitter and Receiver PCBs

Moving Object Receiver

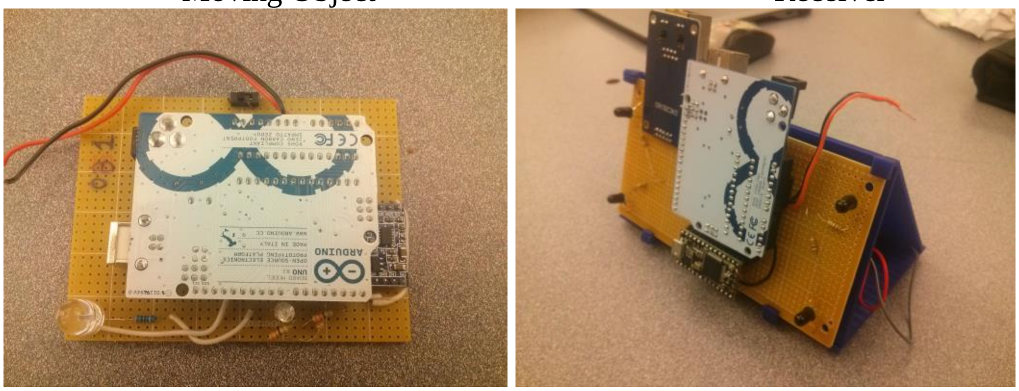


Figure 2.1 Images of moving object and receiver

2.2 Backgrounds and Prior Art

Performing research I found that there is not any project that uses light to track objects. But, the research uncovered other asset tracking systems that use GPS, Wi-Fi, or RFID. These systems work by tracking an object by using a GPS signal to connect a number of different devices that are used to monitor the location of the object. The Wi-Fi model will connect to a Wi-Fi router which will be able to track where the object is. RFID uses short range transmission based on when a sensor is close to the RFID.

The advantages of our design are low power consumption compared to most of the other options. Both Wi-Fi and GPS will take a considerable amount of power to be able to track where the objects are for a long period of time. Our solution will be able to track objects for long periods of time without the need to change the battery. The one downside is that using our method the LED and the sensor has to be able to see each other. Both the GPS and Wi-Fi just have to be able to connect to the GPS satellite or the Wi-Fi router which does not need to be in the line of sight of the object being tracked.

2.3 Development Environment and Tools

The design environments needed for this project are: Arduino IDE, MATLAB®

Hardware and Software tools that this project needs are: Arduino, Processing, Teensy, and the Ethernet.

The test equipment needed for to test this project is: a Multimeter, a voltage power supply, an oscilloscope, and computers.

2.4 Related Documents and Supporting Materials

These are the industry standards that were used in I.L.A.T.S:

- I2C
- SPI
- TCP/IP
- IEEE 802.3 (Ethernet)

2.5 Definitions and Acronyms

I.L.A.T.S: Infrared Light Asset Tracking System
GUI: Graphical User Interface
PCB: Portable Circuit Board
RGB: Red, Green, and Blue
FFT: Fast Fourier Transform
GPS: Global Positioning System
LED: Light Emitting Diode
RFID: Radio-Frequency Identification
IR: infrared
SPI: Serial Peripheral Interface
IDE: Integrated Development Environment
I2C: Inter-Integrated Circuit
SPI: Serial Peripheral Interface
TCP/IP: Transmission Control Protocol/Internet Protocol
IEEE: Institute of Electrical and Electronics Engineers
ADC: Analog to Digital Conversion
VCC: IC Power-Supply Pin
PHP: Personal Home Page (coding language)
MCU: Microcontroller

3 Design Considerations

3.1 Assumptions

For our project we assume that the receivers will be powered from a separate source than batteries. We used batteries so that we could test the receivers but we have assumed from the beginning that the receivers would be powered from a different source. We also assume that anywhere where I.L.A.T.S will be used has an internet port near the receivers as well. We also assume that anyone needing to use I.L.A.T.S will have a computer available to run the executable file which is the GUI. The database the GUI uses will also be on the same computer that the GUI is running on.

3.2 Realistic Constraints

One of the major constraints that we have is that the ram be on the Arduino Uno. The only reason that we have to use two microcontrollers on the receiver is because the Arduino does not have enough ram to for the FFT code and to transmit the values got from the FFT to the server.

Another constraint is that our server needs to be able to transmit the data fast enough to send the data when an object is transmitting it. We had to make sure the server that we used was able to send the data the first time an object was seen.

We needed to make sure that we could get a range of 8 feet from the transmitter to the receiver. This range is necessary for all types of objects that could move under the sensor. The range is set by the fact that most doors are 8 feet tall.

For the transmitter we wanted to make sure that we picked frequencies that are not used that often in other systems. For example, we kept the frequency under 1 kilohertz so that remotes would not affect our sensors. Most remotes work at a frequency of over 1 kilohertz. This frequency range is also suited for the hospital setting. The equipment form the hospital will have a frequency greater than 1 kilohertz which will let I.L.A.T.S function without much interference.

For the FFT we had to pick a sampling rate so that all of the frequencies that we use will be read correctly. Picking this sampling rate is tricky because the higher the speed the better the separation of the frequencies. But, having a higher sampling rate will cause our sensor to read values from other objects that use infrared light.

3.3 System Environment and External Interfaces

I.L.A.T.S needs to use I2C for the communication of the accelerometer to the Arduino. Without using I2C the number of pins needed to communicate the accelerometer to the Arduino would exceed the amount of pins that are available on the Arduino. This is because the Arduino has a very limited number of pins that can be used for inputs and outputs, and some of the pins are used for the infrared LED and the battery life checker.

The Ethernet chip that we have communicates using SPI with the Arduino. The Ethernet chip uses TCP/IP to gain access to the internet. To do this the Ethernet card uses the IEEE 802.3 which lets the Aurdino have access of the Ethernet port.

The GUI uses MATLAB and only the functions that MATLAB allows. This means that the GUI cannot function in real time as MATLAB does not support multiprocessing

The FFT has to function within the compiler that Arduino uses. This means that the entire math that is used in the FFT code has to be able to be within the limits of the Arduino Uno.

3.4 Industry Standards

We used the following industry standards: I2C, SPI, TCP/IP, and IEEE 802.3 (Ethernet).

I2C helped with our design as it allowed us to communicate with the accelerometer and the Arduino. Using I2C we were able to calibrate the accelerometer so that we could get values that we were able to use for motion detection. Similarly, we used SPI on the Ethernet card to set up the IP address which allowed us to communicate with the server. The Ethernet chip uses IEEE 802.3 to access the internet which allows the Arduino to set an IP address to talk to the server.

To code on the Arduino we had to use the Arduino IDE. This allowed us to program the Arduino to implement the tasks we needed the Arduino to do. This allowed us to be able to program the FFT and let us detect motion based on the accelerometer values. Similarly we had to use MATLAB to program the GUI. This lead the GUI process being a bit simpler to create as MATLAB has it own GUI creation tool. But, coding these different elements was a bit more confusing as understanding how the elements worked was needed to be learned.

3.5 Knowledge and Skills

Mark:

Engineering Circuit Analysis, Electronic Circuits, Embedded Systems, Signals & Systems, Digital Signal Processing, Introduction to Digital Control, C++ Programming
I had to learn how MATLAB GUI creation. This part was a struggle as MATLAB deals with creating a GUI weirdly. Interacting with the elements can be cumbersome at times which lead to the code not working all the time.

Moheb:

Relevant courses were Engineering Circuit Analysis, Electronic Circuits, Embedded Systems, Signals & Systems, Digital Signal Processing, Introduction to Digital Control, C++ Programming, and Data acquisition, instrumentation, and process control.
I had to learn how to implement digital signal processing on a microcontroller and I had to learn the implementation of a communication protocol I2C.

Ahmed:

Relevant courses were: intro to embedded systems, intermediate embedded systems, data acquisition, instrumentation and process control, Electronic circuits, C++.
I learned about the Ethernet protocol and how to get a microprocessor online to communicate with a server, I had to learn the PHP language to program the response of the server.
I learned about detecting the voltage of a battery powering a microcontroller using software.
.

3.6 Budget and Cost Analysis

The budget for I.L.A.T.S is broken down into two sections, the receiver and the transmitter. For the transmitter the budget is 35 dollars. Most of this cost comes from having to buy an Arduino. For the transmitter the budget is 55 dollars. Similar to the transmitter, most of the budget comes from having to get an Arduino and a Teensy. This means that the total budget for one transmitter and one receiver is 90 dollars.

Bill of Materials can be found in Appendix A.

3.7 Safety

We had to make sure that the mounts are strong enough to be able to hold the weight of the receiver PCB and the battery. If we do this then we know that the receiver would be able to handle the weight for a long period of time, especially if the battery was replaced with a different power source.

We had to make sure that the connections that we made on our PCBs are secure. The importance of these secure connections is when moving the objects around or moving the receivers. Both the objects and the receivers need to be able to be moved around without shocking the mover. This also means that the wire connections to the battery need to be secured so that is not a short circuit going through the circuit.

3.8 Performance, Security, Quality, Reliability, Aesthetics etc.

The considerations that we made were all based on I.L.A.T.S being used in a hospital. We knew that I.L.A.T.S had to be able to work every single time that the object moved under the sensor. Some machines will go into a room and stay there until that patient ends up leaving, and I.L.A.T.S needs to know that the object is in the room. If we did not end up picking up objects every time that an object moved under the sensor then our tracking would not be correct. To ensure that objects were always detected we had to calibrate the sensors to point in the correct position. Since the placement of the sensors that we used were not always the same, we had to adjust the sensors to look towards the opening of the door.

To achieve the needed performance we kept in mind how fast the server was able to respond. If the server responded too slowly then the object being tracked would be seen but would not have its position updated on the server. This problem would cause the objects to appear to be jumping from place to another randomly. This means that I.L.A.T.S would not work to its full potential. To keep the objects being tracked correctly then the server that is used needs to be tested to see if the server can respond within 400 milliseconds.

For aesthetics we wanted to make sure that the mount would be able to support the sensor circuit but would also not stick out too much. To do this we ended up using a dark blue color for the 3D printed mount compared to the original bright orange. We also decided to make the mount hollow so that we could put the batteries inside the mount so that the batteries are hidden.

3.9 Documentation

The process for generating and maintaining technical documents was dependent on the different parts of the project. For example, we created and maintained all of the documentation when it depended on the GUI. When it came to the circuit design I managed the documentation that was used for creating the circuits and adjusting the circuits that we used throughout the design process.

When it came to the code we have many versions stored so that we know where we had things at certain weeks. For example we have some of the early works of the FFT code compared to the version that we use for the final product. Another reason that we decided to keep the code is to have a backup if anything went wrong. This helped out when we made a change and could not figure out why the code stopped working.

Before we made the PCBs we made sure to map out where we wanted each piece to go. This let us know that we had enough room for all of the components to fit on the PCB. Mapping the PCBs out also helped with making sure that the connections were right.

3.12 Understanding of Professional and Ethical Responsibility

Moheb:

If our project was to become a commercial product used in industrial settings there would be no major ethical implications though there may be some minor implications. The ethical implication that might occur is the argument that this product can be modified to track people rather than objects. While this may be true and even possible it can’t be considered the fault of the product or us as the engineers who designed the product if someone was to corrupt our product. In that regard the ethical standpoint on the individual’s privacy is not affected by our product.

The ethical issue of a person’s health may also be argued against our project. By this I mean the infrared exposure that the infrared light exposes to people. This however is taken into consideration in our project and we strived throughout the design process to make exposure as minimal as possible to the extent that exposure is not an issue. We did this by making the infrared transmitter only transmit a signal when the transmitter is being moved. In the idle position the signal is not transmitted and there is no infrared exposure.

Mark:

We learned that you need to be aware of what implication that you project can create. For example, using infrared lights is good because the light is not visible to human eyes and is reliable. But, looking directly at infrared light is bad for your eyesight. In turn we made sure that the infrared light would only be turning on when the object is moving. Since people will not be looking directly at the LED when the object is moving, this will help lessen the exposure to people looking directly into the infrared light.

We also learned that thinking of these situations is our responsibility. These situations are for us to think about so that we can adjust how I.L.A.T.S functions which results in the smallest implications. Some of these implications cannot be avoided, such as people thinking that I.L.A.T.S has the ability to track people. These situations can only be solved from people knowing how I.L.A.T.S works only with infrared light.

We also learned that we had a responsibility that our product does not infringe on another companies design. I.L.A.T.S is the system that we created and we need to make sure that it stays a unique product instead of infringing our design on other companies that makes similar products.

The last thing that we learned is that our test data needs to be real. We cannot change the test results so that we get the desired result. The data that we tested needed to be able to work outside of our team testing I.L.A.T.S. I.L.A.T.S was made to work in an industrial setting, which means that our tests need to work without any changes made to make it look like I.L.A.T.S is working.

Ahmed:

If this project were to be a commercial project, then from the ethical point of view, its sole purpose would be to track the devices in the hospital and would never be used to track the people in the hospital because this would mean hacking their privacy. Another thing is that the design of the project would have to be perfected such that this real time locating system would have to instantly track the locations of the devices in the hospital because the lives of the patients might depend on how fast the devices are moved around the hospital, for example, the blood pressure devices or the heart rate monitoring devices. The professional responsibility on my part in the project is to make sure that the server responds to the incoming devices as fast as possible to make sure the GUI is always updated with the latest values as the quickness if this process could save the life of a patient.

3.13 Global, Economic, Environmental and Societal Impact

Mark:

If our product ended up becoming commercial then locating objects in large buildings would become easier. When an object needs to be located then the workers of the building would be able to go look at the GUI to find out where the object is. Being able to store information about the object will also help prioritize where to put objects as well. In a hospital setting I.L.A.T.S would help with the issue of knowing what machines can be taken out of rooms. Having both the location and the description will help the hospital workers know if they can move say a blood pressure machine to another room. If a machine does end up being move out of the room that it should be in then you could easily find the machine using the GUI.

With I.L.A.T.S being power efficient, the number of batteries that are used to keep I.L.A.T.S running would be small. This means that the environmental impact that our I.L.A.T.S has on the world will be very small. Also, having the battery checker the users of I.LA.T.S would know when they would need to change the batteries instead of just letting the object stop working. Having the batteries last for such a long time also means that the objects will be tracked for extended period of times. This will cause the users to not have to worry above the objects to stop being tracked. The stress levels of the users for I.L.A.T.S will end up not worry about the battery life. But, if the user wants to check the life of the battery periodically then they can with the battery life LED that exists on our transmitters.

Moheb:

A societal impact is people's reactions to have the receivers mounted above the doors to the rooms. Some might believe that our receivers look like a device that is used to monitor them as they travel throughout the building when I.L.A.T.S only tracks objects that have infrared LEDs on them. Another social impact that I.L.A.T.S affects is how efficiently a hospital setting can work. For example if there was a hospital emergency and the patient's life depends on a certain machine that is in short supply at the hospital and the hospital employees did not know where one was. With the I.L.A.T.S. system this problem can be solved immediately by looking up the machine on the GUI and finding out where the machine was and save the patient's life as soon as possible.

All of these impacts and more caused by our project shows that it can greatly improve the quality of life with minimal amount of energy usage. There may be some ethical debate on exposure to infrared light but the exposure is minimal and non-threatening. For this reason I.L.A.T.S. is beneficial for commercial use.

Ahmed:

If the Project were to become a commercial one then it can have a global impact; this project offers a cheap effective way not only to track devices in hospitals but in general to quickly and efficiently track objects in crowded places or buildings because it uses Infrared Light and Ethernet which are cheaper than using RFID which is commonly used today to track moving objects.

3.14 Contemporary Engineering Issues

Mark:

One of the main contemporary issues that were faced for me was which programming language to use for the GUI. I ended up using MATLAB because of initial plans of doing heavy mathematical calculations on the computer side. We ended up changing the calculation to a microcontroller instead of the computer. I had the GUI almost completed so we had the decision to keep the GUI in MATLAB instead of changing to another programming language. The result of keeping MATLAB as the main language for the GUI ended up causing the issue that the GUI could not be real-time.

Another contemporary issue is that having the GUI be as lightweight as possible. The reason for this is that we had an idea about porting the GUI to an android tablet. Being on a tablet the GUI needed to be lightweight so that it could function without any performance issue. But, once again because of choosing MATLAB, porting the code that I wrote for the GUI would not work at all in android.

For the transmitter we had to make sure that it had a low power output. The transmitter needs to be able to last for long period of time. To do this we tried to make sure that the transmitter used the least amount of parts.

Moheb:

A crucial contemporary issue that I faced when taking part in this project was the task of implementing the Fourier transform on a microcontroller. Not only have I never implemented the FFT digitally I had never done high level math on a microcontroller. After doing a considerable amount of research and studying the different approaches to implementing FFT on a microcontroller I settled for the Cooley Turkey method of implementation. Once the math of the FFT was fully implemented within the program I had to make sure that the accuracy was correct enough to be able to implement our design with it.

Another contemporary engineering issue I faced with the implementation of the FFT code on a microcontroller was the size of the FFT code. Due to its high level computation it took up most of the memory on the microcontroller. However there was still space needed for the Ethernet code and so we made the decision to separate the codes so that there was no interference with the two codes.

Ahmed:

One of the Contemporary engineering issues that I faced was using the Arduino as a server to store the data base of the objects using HTML, it did not work because it was not possible to transmit data from the Arduino to an IP address using HTML so the alternative was to use a remote server that runs PHP code, where the Arduino would send variables to the server within a link and then using PHP ability to easily manage CSV files to store the variables to the CSV file.

3.15 Recognition of the need for and an ability to engage in lifelong learning

Mark:

This project helped me recognize the need to engage in lifelong learning because there are still many aspect of electrical engineering that I do not understand. For example I did not know how to make a GUI in MATLAB, which can be used in many different situations, and know I have the tools to be able to do so. This project also helped me realize that if I do not have, or lose, the need to engage in lifelong learning that I will not be able to further create any other projects. This pushes me to keep engaged to keep on learning every single day.

This project helped me develop the skills to be able to engage in lifelong learning. It was achieved from have to research how to create a GUI that interacts with the database that I created. This painstaking process caused me to understand that engaging in lifelong learning can be tedious. But, I still need to push through and keep learning about the aspects that I still do not understand. Having the knowledge helps me figure out the problems that occurred in the project so that the project could be completed. This means that I will need to keep learning so that when problems occur I am able to figure out a solution to fix them.

Moheb:

This project helped me recognize the need and developed the ability of lifelong learning. As soon as this project began I realized that in order for this project to be a success each of us must be able to learn aspects of engineering on our own outside of our courses. This made me focus and push to learn all that I could within the time frame needed for my aspects of the project. I also recognized the need for asking others for help. Before the beginning of this project I had never understood the need for asking others for help, but after I began working on the project it became clear that that type of stubborn ideology would not allow me to fully succeed within this project. It was aspects like this that allowed me to grow and develop abilities in lifelong learning.

Ahmed:

This project made me recognize that the purpose of learning electrical engineering is not limited to the field of electrical engineering but also to continue to learn how to apply this knowledge in computer engineering or computer science, for example, I did not learn networking or how to program a device to communicate with an online server but I used my knowledge of embedded systems and programming to easily learn the PHP language and to program the Arduino Uno to communicate with a server.

3.16 Importance of Teamwork

Mark:

Without working in a team the project would have taken longer to complete. Being able to have each group member make changes at the same time made it simple to test and modify I.L.A.T.S. The group was able to test and adjust the I.L.A.T.S until it function correctly when an object was moving underneath the sensor.

Having team members made managing time spent on the project easier. The progress that occurred each week was able to be focused in the direction of the group member that had the most time. This lets the stress decrease when a group member did not have that much time during a given week. This kept each member able to keep working on the time that had available compared to being forced to have parts even if they did not have the time.

Moheb:

This design project showed me the importance of team effort. By working in a team I was able to understand that in any situation in which a member of a team needs help the team is there for help and can solve the problem together. So rather than being stuck at a section and not being able to continue and maybe ultimately failing we can work together to solve the problem and finish.

Ahmed:

The importance of teamwork is that when we broke down the project into individual tasks divided between all three of us. It made the project work easily because each one of us spent time perfecting his task, so when we put them together, the tasks were completing each other. So it is about knowing how to play your role in a project as an engineer.

4 High Level Design

4.1 Conceptual View

The high level conceptual system is depicted in the block diagram shown in Figure 4.1. The diagram is comprised of four main blocks (subsystems): the object block, the room block, the internet block, and the computer block. These blocks were split up into relevancy to each other. The object block is comprised of the transmitter and its components that transmit the frequency signal. It is further broken down to the battery checker, the accelerometer, the infrared LED, and the microcontroller that ties it all together. The room block is made up of the receiver that receives the transmitted signal from the asset. This block is comprised of infrared phototransistors that receive the signal, the microcontroller that computes the Fourier transform of the signal, and the microcontroller connected to the Ethernet adapter that sends the information to the online server. The internet block is mostly the online server where the information that had been compiled by the receiver block is taken and organized so that the information can be readily viewed by a user. This happens within the computer block in which the GUI extracts information from the online server and compiles a database in which the information is compared to previous information and outputs an update for the user to view.

Moheb was responsible for the accelerometer and transmission of the object block (see Reference 10). He was also responsible for receiving the signal and the FFT analysis of the room block (see Reference 11).

Ahmed was responsible for the battery checker of the object block (see Reference 8 and 9). He was also in charge of the Ethernet communication protocol between the room block and the internet block and he was responsible for the online server(see Reference 2 - 7).

Mark was responsible for the communication between the internet block and the computer block. He was also responsible for the database and GUI of the computer block (see Reference 12).

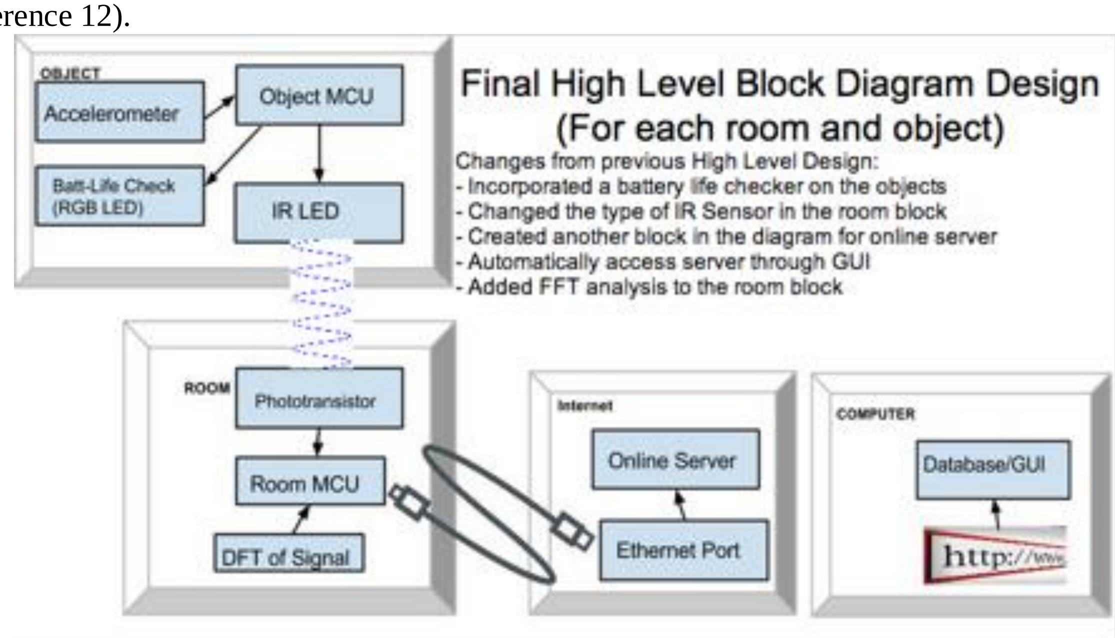


Figure 4.1 System High Level Block Diagram

4.2 Hardware

The hardware aspect of our project consisted of a 3D printed wall mount and two PCB designs: a transmitter and a receiver. The wall mount is meant for the mounting of the receiver PCB in each room. The receiver PCB is derived of four infrared phototransistors each connected to the microcontroller communicating with another microcontroller through SPI which communicates to the Ethernet adapter to an online server. The transmitter PCB consists of an accelerometer that connects to a microcontroller using I2C. The microcontroller also controls an RGB LED that signals the life of the battery and an infrared LED that is connected to the digital pin of the microcontroller. Moheb was responsible for the creation of the 3D model and the design of the transmitter PCB. Ahmed was responsible for the design of the receiver PCB. Mark was responsible for implementing the designs of the PCB.

4.3 Software

Moheb:

Used the Arduino IDE to implement the I2C communication between an Arduino Uno and an accelerometer (see Reference 1 and 10), also on the same Arduino code was the transmission code. Used the Arduino IDE to implement the Fourier Transform analysis and ADC conversion on a Teensy 3.0 (see Reference 11)

Ahmed:

Used the Arduino IDE to implement the Ethernet protocol to communicate between the receiver and the online server, used PHP to organize and control the online server (see Reference 2 - 7), and used the Arduino IDE to create the battery checker code (see Reference 8 and 9).

Mark:

Used Matlab to implement a communication between the online server and the database and used Matlab to create the database and the GUI to interact with the database using CSV file manipulation. (see Reference 12)

5 Low Level Design

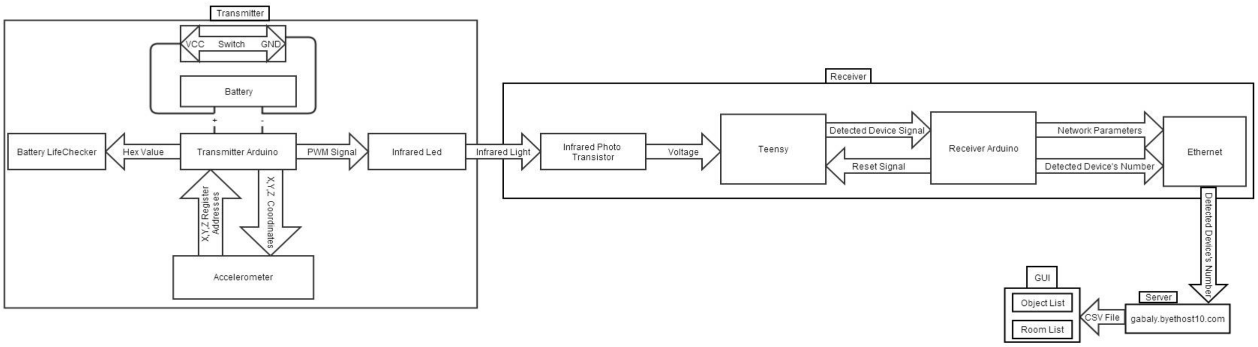


Figure 5 Low Level Design block diagram

5.1 Module 1/Moheb Amini

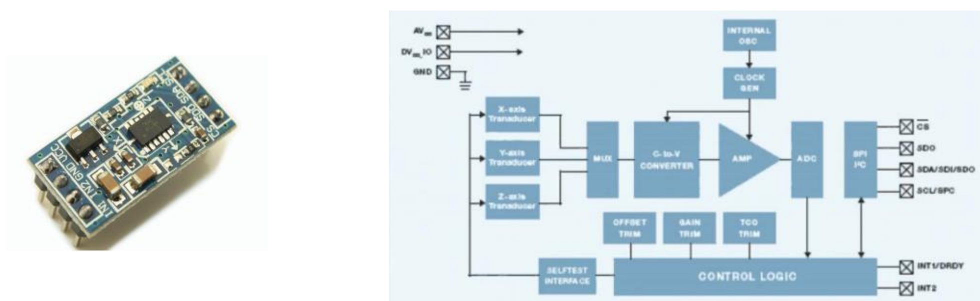


Figure 5.1 Module 1: MMA7455L 3-Axis Low-g Digital Output Accelerometer

5.1.1 Module 1: Processing description

The MMA7455L 3-Axis Low-g Digital Output Accelerometer, it records the x, y, z coordinates of the object's position in space.

5.1.2 Module 1: Interface description

The Accelerometer receives the addresses of the registers where the x, y, z coordinates of the objects position in space are stored as input from the Transmitter Arduino module and outputs the x, y, z coordinates of the object to the Transmitter Arduino Module.

5.1.3 Module 1: Processing Details

The Accelerometer communicates with the Transmitter Arduino module using the I2C bus. The Accelerometer receives the start sequence from the Transmitter Arduino module. The Accelerometer receives the address of the x register where the x coordinate of the object is stored. The x coordinate of the object's position is read. The Accelerometer receives the address of the y register where the y coordinate of the object is stored. The y coordinate of the object's position is read. The Accelerometer receives the address of the z Register where the z coordinate of the object is stored. The z coordinate of the object's position is read. The Accelerometer receives the stop sequence from the Transmitter Arduino module to end the transmission.

5.2 Module 2/Ahmed El Gabaly

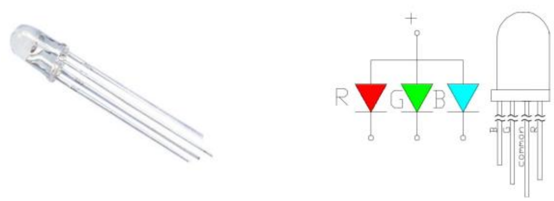


Figure 5.2 Module 2: Left (RGB Led) Right (RGB Led Schematic)

5.2.1 Module 2: Processing description

The Battery life checker, it shows the battery life of the battery powering the Transmitter.

5.2.2 Module 2: Interface Description

The Battery life checker receives a hex value from the Transmitter Arduino as input and outputs visible light

5.2.3 Module 2: Processing Details

The Battery life checker outputs light in the visible light spectrum that corresponds to the hex value that it receives from the Transmitter Arduino.

5.3 Module 3/Ahmed El Gabaly

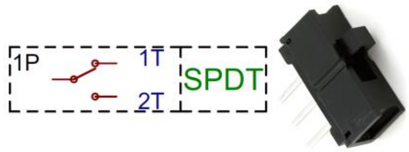


Figure 5.3 SPDT Switch

5.3.1 Module 3: Processing description

The Switch, The switch module controls the Battery Life Checker module.

5.3.2 Module 3: Interface Description

The Switch module receives voltage as input from the battery module and outputs this voltage to the Transmitter Arduino module.

5.3.3 Module 3: Processing Details

The switch module turns inputs either HIGH (5 volts) or LOW (0 volts) to the Transmitter Arduino Uno module depending on which direction it is moved. If it is moved towards VCC, the input signal to the Arduino Uno module is a HIGH and if it is moved towards GND, the input signal to the Arduino Uno module is LOW.

5.4 Module 4/Moheb Amini

Figure 5.4 Module 4: Battery

5.4.1 Module 4: Processing description

The battery, it powers the Transmitter module

5.4.2 Module 4: Interface Description

The battery module outputs 6 volts to the Transmitter Arduino module

5.4.3 Module 4: Processing Details

The battery module consists of 4 AA 1.5V batteries used to power the Transmitter Arduino module, the Accelerometer module, the battery life Checker module, the switch module and the Infrared Led module.

5.5 Module 5/Moheb Amini

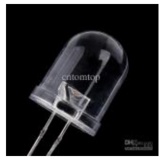


Figure 5.5 Module 5: Infrared Led

5.5.1 Module 5: Processing description

The infrared Led. It is the signature of the devices.

5.5.2 Module 5: Interface Description

The Infrared Led module receives a pulse width modulated signal as input from the Transmitter Arduino module as input and outputs Blinking Infrared light to the Receiver module.

5.5.3 Module 5: Processing Details

The infrared Led module blinks the infrared light at a rate equal to the frequency of the pulse width modulated signal it receives, each moving object is set to emit Infrared light from the Infrared light Led module at a unique frequency which identifies the object in the system.

5.6 Module 6/Moheb Amini, Ahmed El Gabaly

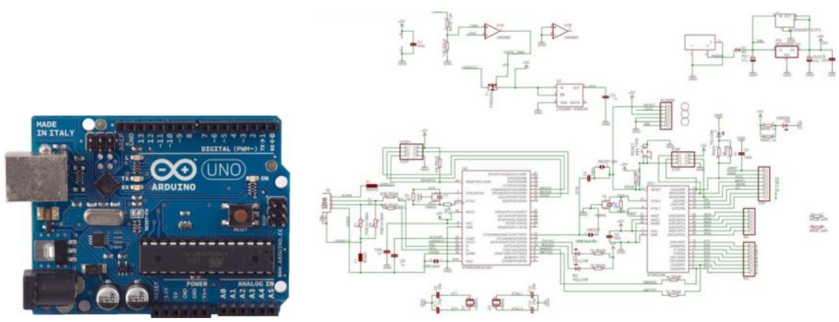


Figure 5.6 Module 6: Transmitter Arduino Uno

5.6.1 Module 6: Processing description

The Transmitter Arduino module is an Arduino Uno which is the core of the transmitter module, it detects the movement of the object and transmits the unique frequency of that moving object.

5.6.2 Module 6: Interface Description

The Transmitter Arduino receives the x, y, z coordinates of the object's position in space from the Accelerometer module and the Battery life checker switch voltage from the switch module as input and outputs a hex value to the Battery life Checker module and a PWM signal to the Infrared Led Module.

5.6.3 Module 6: Processing Details

The Transmitter Arduino module Reads the x, y, z coordinates of the object's position from the Accelerometer module through the I2C bus and reads the Switch module voltage and transmits the unique device frequency to the Infrared Led based on the following conditions, if the switch module input is HIGH, the x coordinate value is less then 0.13 Units and the y coordinate value is less than 0.13 Units the Battery_RGB function is called which calculates the VCC value of the Transmitter Arduino microcontroller according to the micro controller internal reference voltage according to the following equation:

int results = (((InternalReferenceVoltage * 1023L) / ADC) + 5L) / 10L

The result is mapped on the Battery Checker Module based on the HEX value sent to that module. At least three conversions have to be done to calculate an accurate VCC value. The calculated VCC value falls in the range of 0 to 500, this range is divided into 3 sub ranges which are mapped on the RGB LED;

0-200 = Red, 200-400 = Yellow and 400-500 = Green.

Otherwise if the switch module input is LOW, the x coordinate value is greater or equal to 0.13 Units and the y coordinate value is less than 0.13 Units, the Battery_RGB function is not called and the Batter life Checker module is turned off and the unique device’s frequency is transmitted to the Infrared led module as a PWM signal.

Therefore, the Battery life Checker module can only work when the object is not moving otherwise, it will interrupt the PWM signal sent to the Infrared Led module and the Object might not be accurately detected.

5.7 Module 7/Mark Logston

Figure 5.7 Module 7: Infrared Phototransistor

5.7.1 Module 7: Processing description

The Infrared Phototransistor Module detects Infrared Light.

5.7.2 Module 7: Interface Description

The Infrared Phototransistor module receives the Infrared light coming from the Infrared Led module as input and outputs approximately 5 volts across it to the Teensy module.

5.7.3 Module 7: Processing Details

If the Infrared Photo Transistor module receives Infrared light from the Infrared Led module, it allows current to pass through the resistor connected to the transistor and therefore an approximate voltage of 5 volts is read from the SIGNAL OUT junction shown above, but if no Infrared light is received from the Infrared Led module then no current passes through the resistor connected to the transistor and therefore an approximate voltage of 0 volts is read from the SIGNAL OUT junction shown above.

5.8 Module 8/Moheb Amini

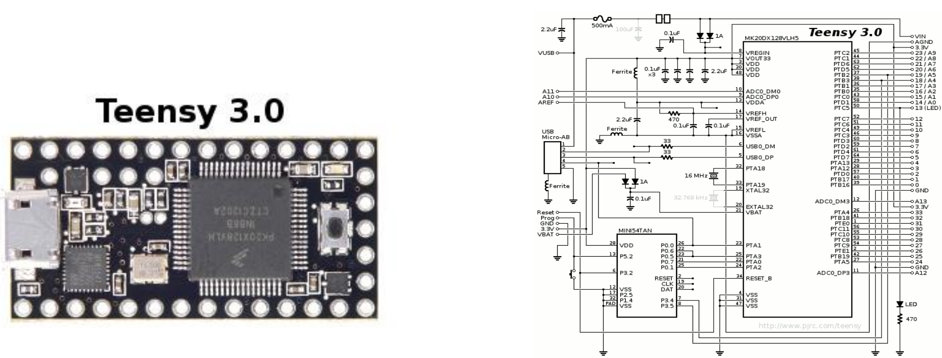


Figure 5.8 Module: 8 Teensy 3.0

5.8.1 Module 8: Processing description

The Teensy module is the Teensy 3.0 microcontroller which is used to identify the frequency of the incoming continuous time domain signal.

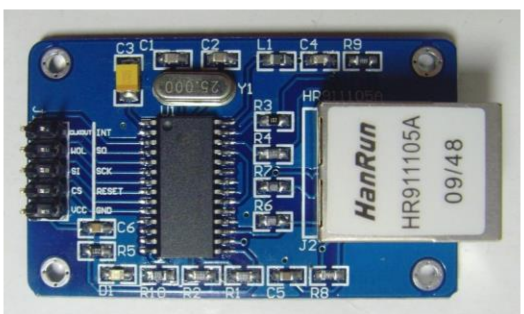
5.8.2 Module 8: Interface Description

The Teensy module receives the moving objects unique continuous time domain electrical signal from the Infrared Phototransistor module and the reset signal from the Receiver Arduino module as input and outputs a "detected device" signal to the Receiver Arduino module

5.8.3 Module 8: Processing Details

The Teensy module receives the continuous time pulse signal from the Infrared Phototransistor module, performs Analog to digital conversion on this continuous time domain signal to convert it to a discrete time domain signal. Then it performs the 128 point Cooley–Turkey FFT algorithm on this discrete time domain signal at a frequency of 1 kHz to detect its frequency which represents the moving objects id. The Teensy module outputs a high to one of the pins of the Receiver Arduino module depending on the Frequency it detects and it receives a reset signal from the Receiver Arduino module so it can be ready for the next incoming object.

5.9 Module 9/Ahmed El Gabaly



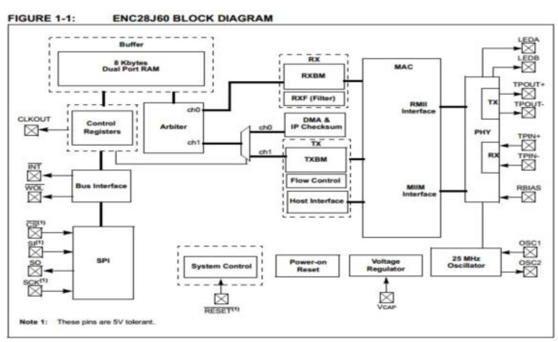


Figure 5.9 Module 9: Ethernet Controller ENC28J60

5.9.1 Module 9: Processing description

The Ethernet module is the ENC28J60 chip. It is needed to connect the Receiver Arduino to the internet to record the object's current location in the hospital

5.9.2 Module 9: Interface Description

The Ethernet module receives the network parameters and the detected Device's number from the Receiver Arduino module as input and outputs the detected device's number to the server module.

5.9.3 Module 9: Processing Details

The Ethernet module communicates with the Receiver Arduino module using the SPI bus. The Ethernet module makes sure that it received the MAC address of the Ethernet Shield from the Receiver Arduino module, once this happens, the Ethernet module becomes initialized, then the Ethernet module sends a request to the DHCP server to ask for a Dynamic unique IP address to identify the Receiver Arduino module on the network, Then the Ethernet module sends a request to the DNS server to convert the server website (www.gabaly.byethost10.com) into its equivalent IP address which uniquely identifies it on the internet. Once the Server IP is resolved, the Ethernet module waits for the detected device's number from the Receiver Arduino module and responds by sending this received number to the server module within a link.

5.10 Module 10/Ahmed El Gabaly

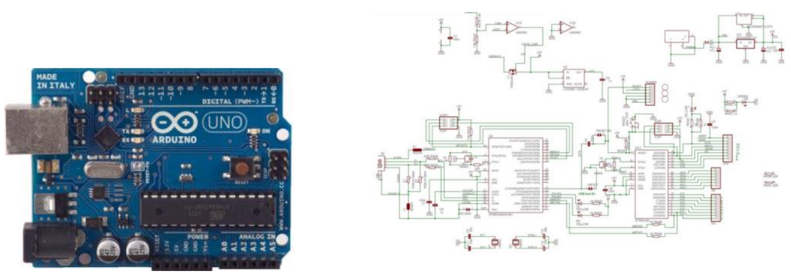


Figure 5.10 Module 10: Receiver Arduino Uno

5.10.1 Module 10: Processing description

The Receiver Arduino module is an Arduino Uno microcontroller. It controls the Ethernet module and the Teensy module.

5.10.2 Module 10: Interface Description

The Receiver Arduino module receives the signals that correspond to the detected Device from the Teensy module as input and outputs the detected device's number to the Ethernet module and the reset signal to the Teensy module.

5.10.3 Module 10: Processing Details

The Receiver Arduino module uses the ether card library to initialize the Ethernet module using the ether.begin function, then uses the ether.dhcpSetup function to request a dynamic IP address for the Receiver Arduino to uniquely identify it on the network, then uses the dnsLoopup function to resolve the server web address (www.gabaly.byethost10.com) into its corresponding IP address. The receiver Arduino waits for a signal from the Teensy module, it sends the device number to the server module based on the pin where the signal is received. Then it sends the reset signal to the Teensy module to prepare for the next device.

5.11 Module 11/Ahmed El Gabaly

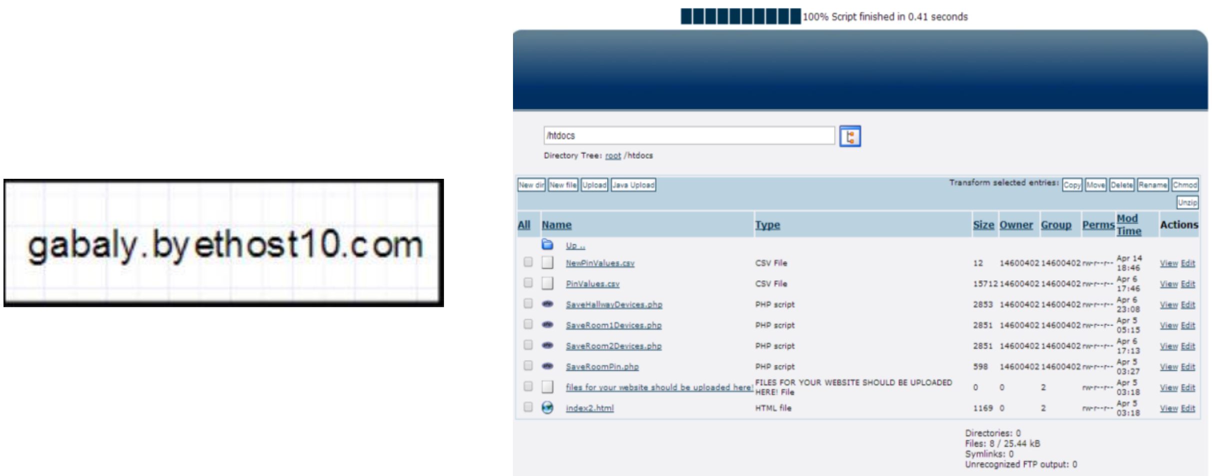


Figure 5.11 Module 11: Server

5.11.1 Module 11: Processing description

The server module contains the real time locations of the devices in the hospital

5.11.2 Module 11: Interface Description

The Server module receives the detected device's number from the Ethernet module through a link as input and outputs a CSV file to the GUI module

5.11.3 Module 11: Processing Details

The Server module contains one CSV file which the database consisting of 2 columns lists, one for the rooms and the other for the devices. The Server waits to receive the detected device's number within a link, runs a PHP algorithm that checks which number is received, the algorithm converts the CSV file into an array, updates it and writes this array to the CSV file. Each receiver has a PHP file that receives the number of the device detected by this receiver at the moment it is detected and updates the location of this device within the CSV file.

5.12 Module 12/Mark Logston

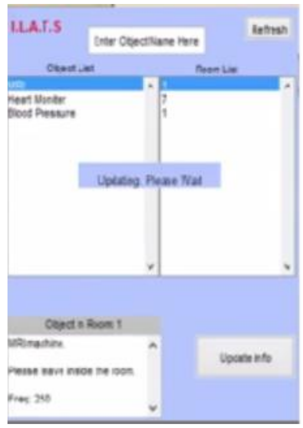


Figure 5.12 Module 12: GUI

5.12.1 Module 12: Processing description

The Matlab GUI, it is a graphical user interface that allows the user to see the latest locations of the devices in the hospital in real time

5.12.2 Module 12: Interface Description

The GUI receives a CSV file from the Server module as input and outputs the latest location of the objects through the screen.

5.12.3 Module 12: Processing Details

The GUI consists of two columns (objects list and room list), 2 buttons (Refresh and update info), when the refresh button is pressed, the Matlab script for the GUI downloads the CSV file from the server module using the urlwrite command and stores this CSV file as a struct where it can access 2 columns containing the devices that are in the hospital and the real time locations of these devices in the hospital, it reads this information from the CSV file and updates the room list column in the above figure to reflect the current location of every device in the hospital. When the Update info Button is pressed, The GUI shows the user information specific to the devices, for example the Infrared light signal frequency corresponding to a certain device or weather this device should be left in the current room or not etc...... . In addition, the GUI allows the user to add new information about the devices in the data base.

6 Administrative and Other Design Issues

6.1 Project Management

Project Task Manager

Group Members: [Moheb Amini, Ahmed El Gabaly, Mark Logston]

Deadline: [April 2014]

Done?	Fall Quarter	Due By	Responsible for Task
Yes	Planning	10/18/13	The three of us
Yes	Preparation	10/25/13	The three of us
Yes	Find best LED wavelength	10/25/13	The three of us
Yes	Order IR LED	11/1/13	Moheb
Yes	GUI research	11/15/13	Mark
Yes	Order Accelerometor	11/1/13	Moheb
Yes	Finish Accelerometor code	11/29/13	Ahmed.pass on to Moheb in 11/5/13
Yes	GUI template	11/29/13	Mark
Yes	Ethernet coding research	12/6/13	Ahmed
Yes	Fall Quarter Presentation	12/6/13	The three of us

Done?	Winter Quarter	Due By	Responsible for Task
Yes	Order Microcontrollers	1/24/14	The three of us
Yes	Order the rest of the parts	1/31/14	The three of us
Yes	Finish FFT	2/21/14	Moheb
Yes	Finish Ethernet and Online Server	2/21/14	Ahmed
Yes	Finish GUI	2/21/14	Mark
Yes	Demo	3/14/14	The three of us

Figure 6.1 Project Task Manager

Figure 6.1 shows the scheduler that was used to manage the project progress for all the members and each member had the ability to update and add to each section of the scheduler.

Moheb: The project was managed in a way so that we were making visible progress every week and in order to do that we divided the tasks so that each of us was able to work on the project as much as possible. Our schedule consisted of meeting twice a week so that we could compare each of our progress with the rest of the team and this was so that if any of us was stuck on a specific part and was unable to solve a problem we would work together through it. Through this I learned that team projects can be beneficial if there is a task I don't understand but also difficult to maintain contact with each other to maximize our efficiency.

Ahmed: We managed the project in an equal and efficient way such that the tasks of the project were divided equally between the team members. In order to get the project done, we met each other at least once a week to work on the project together, if one of the team members had a problem on one of his tasks the other two would help him to get it done. In terms of teamwork, I learned how to work on a small isolated task that would soon show its importance in the big picture when combined with the remaining tasks. I also learned how to communicate with my teammates to make sure I optimize my tasks to work such that they work correctly when combined with their tasks.

Mark: The project was managed so that each member had enough time to work on their separate parts. We meet twice each week so that we were able to know where each others progress was. We kept in communication with each other so that we knew what was going on throughout the whole project. I learned that sometimes you have to take the lead when it comes to assigning parts. Sometimes during the project very little would get done because nothing was assigned to a person during a given week. I also learned that you need to be able to rely on your group members. There were times where it would take me longer to finish a task because I did not ask for help and just figured out the issue on my own.

7 Experiment Design and Test Plan

7.1 Design of Experiments

7.1.1 Test Experiment case 1: Accelerometer Output / Moheb Amini

1. The objective of this experiment is to successfully read the x, y, z values from the Accelerometer. This experiment tests acquiring data from the Accelerometer. This Experiment measures the coordinates of the position of the Accelerometer in space.
2. Setup: Accelerometer - Transmitter Arduino Pinout

MMA7455 Accelerometer	Transmitter Arduino
SDA	SDA
SCL	SCL
VCC	VCC
GND	GND

Table 7.1.1 Accelerometer - Transmitter Arduino Pinout

3. Experimental procedure:
   - A) The Transmitter Arduino is programmed with the Accelerometer code
   - B) The Accelerometer is moved in 3D space

   How to collect data:
   - A) The Arduino Serial monitor is opened.
   - B) The x, y, z values are observed
4. The expected results are seeing positive x, y, z values on the serial monitor as the accelerometer moves and seeing negative or zero x, y, z values when the accelerometer returns to the origin.

7.1.2 Test Experiment case 2: Accelerometer Output Threshold / Moheb Amini

1. The objective of this experiment is to find the threshold for the Accelerometer x and y values so that the PWM signal is sent to the Infrared Led module after this threshold is exceeded. This experiment tests how the Accelerometer and the Infrared led can work simultaneously together. This experiment measures the least x, y coordinate values after which the Infrared Led has to turn on.
2. Setup:

MMA7455 Accelerometer	Transmitter Arduino
SDA	SDA

SCL	SCL
VCC	VCC
GND	GND

Infrared Led	Transmitter Arduino
Long pin	pin 11
Short pin	GND

Table 7.1.2.1 Infrared Led - Transmitter Arduino pinout

Table7.1.2.1 Accelerometer-Transmitter Arduino pinout

3. Experimental procedure:
A) The Transmitter Arduino is programmed with the Accelerometer and Infrared Led code
B) The setup is on a small breadboard on a bread board that is moved in 3D space
How to collect data:
A) The Arduino Serial monitor is opened.
B) The response of the Infrared Led to the movement of the Accelerometer is observed
4. The expected result is being able to choose a threshold such that the PWM signal is sent to the Infrared Led only after this threshold is exceeded.

7.1.3 Test Experiment case 3: FFT Spectrum Observation / Moheb Amini

1. The objective of this experiment is to observe how the FFT code works in real time. This experiment tests the detection of a desired frequency using the FFT algorithm. This experiment measures the highest amplitude in the spectrum generated using the FFT algorithm.
2. Setup: The signal generator is connected to the Teensy 3.0 Analog input pin and the teensy 3.0 Serial port is connected via USB to the laptop.
3. Experimental procedure:
A) The Teensy 3.0 is programmed with the FFT code
B) The Signal generator is setup to output a square wave with a specified frequency
How to collect data:
A) The Processing IDE serial monitor is opened and shows the real time spectrum generated from
the Teensy 3.0 FFT algorithm
B) The frequency that has the highest amplitude is observed.
4. The expected result is that the frequency of the highest amplitude in the spectrum will correspond to the frequency of the signal generated from the signal generator.

7.1.4 Test Experiment case 4: GUI Response to Manual Server Update / Mark Logston

1. The objective of this experiment is to check if the data on the GUI would update correctly when the data in the database in the server is manually changed. This experiment tests the

response of the GUI to changes in the server data base. This experiment measures the ability of the GUI to correctly show the latest data in the data base in the server.
2. Setup: Matlab is opened
3. Experimental procedure:
A CSV file with predetermined values for the rooms and the devices frequencies is uploaded
on the server.
How to collect data:
The following command is executed in the Matlab work space to acquire the latest data from the
Server:
temp = csvread(urlwrite('http://www.gabaly.byethost10.com/NewPinValues.csv','pin.csv'))
4. The expected result is that when the rooms in the uploaded CSV file are manually changed, this change will be reflected in the temp matrix which appears in the Matlab work space.

7.1.5 Test Experiment case 5: Detecting IR Light with Amplifiers & Photodiode / Mark Logston

1. The objective of this experiment is to see of the photodiode is able to detect the transmitted signal from 8 feet. This experiment tests the ability of the photodiode to meet the application requirements. This experiment measures the success of the photodiode to detect the Infrared module light at a distance of 8 feet.
2. Setup: The photodiode is connected to the amplifier circuit which is connected to the oscilloscope.
3. Experimental procedure:
The infrared led is moved up to 8 feet from the photodiode.
How to collect data:
The voltage of the detected signal is observed on the oscilloscope.
4. The expected result is that at 8 feet the photodiode circuit should give us an amplified signal with a good readable voltage on the oscilloscope.

7.1.6 Test Experiment case 6: adding new object to database using GUI / Mark Logston

1. The objective of this experiment is to test adding a new object with the GUI to the GUI and the database. This experiment tests the ability of the GUI to add a new object to the offline database. This experiment measures the success of the GUI in adding a new object to the GUI and to the objects database
2. Setup: Matlab is opened, and the GUI code is executed
3. Experimental procedure:
The Update info button is pressed and a new object is entered in the top tab
How to collect data:
The added object is observed in the GUI object list and the offline database file
4. The expected result is that the GUI object list and the offline database file should both be

updated with the new object.

7.1.7 Test Experiment case 7: Ethernet Initialization / Ahmed El Gabaly

1. The objective of this experiment is to test initializing the Ethernet module and displaying the network parameters on the serial monitor. This experiment tests if the receiver Arduino connects to the internet or not. This experiment measures the success of the Ethernet controller to connect the receiver Arduino to the internet.

2. Setup: ENC28J60 - Receiver Arduino pinout

ENC28J60	Receiver Arduino
VCC	3.3v
GND	GND
SCK	pin 13
SO	pin 12
SI	pin 11
CS	pin 8

Table 7.1.7 ENC28J60 - Receiver Arduino pinout

3. Experimental procedure:

The Ethernet module is connected to an Ethernet port via an Ethernet cable and the serial monitor is opened.

How to collect data:

The data to be collected is the network the success of the initialization of the Ethernet controller, the success of the DHCP request and successfully acquiring the network parameters of both the receiver Arduino Uno and the server, which are; the IP address assigned to the Receiver Arduino Uno by the DHCP server, the netmask address, the Gateway address, the DNS server address and the server IP address.

4. The expected result is that the Network parameters would appear on the Serial monitor.

7.1.8 Test Experiment case 8: Two Arduinos Connected to Server / Ahmed El Gabaly

1. The objective of this experiment is to test connecting 2 receiver Arduinos simultaneously to the server. This experiment measures the success of two receiver Arduinos simultaneously sending analog pin outputs to the server
2. Setup: The two Arduinos are connected to two Ethernet modules respectively using the same setup as experiment 7.
3. Experimental procedure:
   A) Wait for the Ethernet modules to initialize
   B) Wait until both Arduinos send a couple of analog pin outputs to the server

How to collect data:

Download the CSV file from the server and check if the outputted values from the pins exist in the CSV file or not

4. The expected result is that the values outputted from the pins from both Arduinos should be found in the file on the server

7.1.9 Test Experiment case 9: Battery with Accelerometer / Ahmed El Gabaly

1. The objective of this test is to show the battery voltage on the RGB Led without interrupting the IR transmission.
2. Setup: RGB Led - Transmitter Arduino pinout

RGB Led	Transmitter Arduino
pin 1	pin D2
pin 2	GND
pin 3	pin D3
pin 4	pin D4

Table 7.1.9 RGB Led - Transmitter Arduino pinout

3. Experimental procedure:

Move the object in the 3D space and return it back to the origin

How to collect data:

Observe the RGB Led light and the Infrared Led light

4. The expected result is that the RGB Led would light up without interrupting the Infrared Led

7.1.10 Test Experiment case 10: LED Detection Range / Moheb, Ahmed, Mark

1. The objective of this experiment is to know which Led color wave length has the longest range of detection.

2. Setup: The photodiode is fixed in any position in the room and is connected to an oscilloscope to measure the voltage that corresponds to the amount of light detected by it.

3. Experimental procedure:

The Led under test is moved as far as possible in a horizontal line away from the photodiode.

How to collect data:

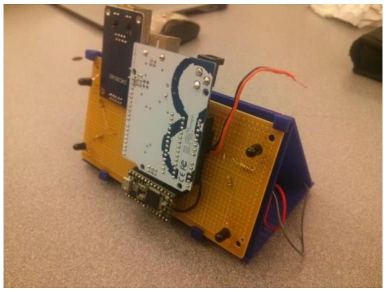
The longest distance at which the highest voltage appears on the oscilloscope for the respective led is noted

4. The expected result is that the Ultra violet led will give the highest voltage at the longest distance from the photodiode.

7.1.11 Test Experiment case 11: Full System Test / Moheb, Ahmed, Mark

1. The objective of this experiment is to test the entire system to make sure that the system functions as required

2. Setup: Mount the receivers to the entrance of the rooms and Hold the Objects



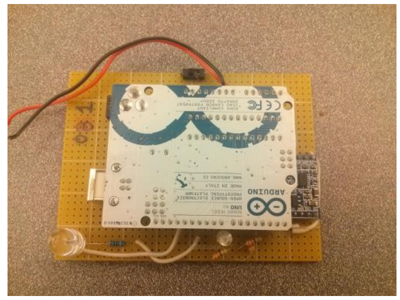


Figure 7.1.11 Left (Receiver) Right (Transmitter)

3. Experimental procedure:

A) Program the transmitter Arduino with the Accelerometer + Battery code
B) Program the Receiver Arduino with the Ethernet code
C) Program the Teensy with the FFT code
D) Open the Receiver Arduino Serial monitor
E) Open the serial monitor for the receiver Arduino
F) Wait for the Ethernet module to initialize
G) Walk with object under the Receiver

How to collect data:

A) Press Refresh on the GUI, the data collected is the latest locations of the objects
B) Check the detection of a device and the server response on the receiver Arduino Serial monitor

4. The expected result is that as the object is passed under the receiver at normal walking speed the GUI will display the latest location of the object and the serial monitor for the receiver Arduino will display the detected device and the response of the server

7.1.12 Test Experiment case 12: Send HTML Script to IP address/ Ahmed Elgabaly

1. The objective of this experiment is to test sending an HTML to the arduino IP address

2. Setup: ENC28J60 - Receiver Arduino pinout

ENC28J60	Receiver Arduino
VCC	3.3v

GND	GND
SCK	pin 13
SO	pin 12
SI	pin 11
CS	pin 8

Figure 7.1.12 ENC28J60 - Receiver Arduino pinout

3. Experimental procedure:

A) Program the Receiver Arduino with the Ethernet HTML code

```
/* HTML Script */
char page[] PROGMEM =
"HTTP/1.0 503 Service Unavailable\r\n"
"Content-Type: text/html\r\n"
"Retry-After: 600\r\n"
"\r\n"
"<html>"
  "<head><title>"
    "Service Temporarily Unavailable"
  "</title></head>"
  "<body>"
    "<h3>This service is currently unavailable</h3>"
    "<p><em>"
      "The main server is currently off-line.<br />"
      "Please try again later."
    "</em></p>"
  "</body>"
"</html>"
;

void setup(){
  Serial.begin(57600);
  /* Ethernet Initialization */
  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
    Serial.println( "Failed to access Ethernet controller");
  /* DHVP request */
  if (!ether.dhcpSetup())
    Serial.println("DHCP failed");
  /* If DHCP request succeeds, Print Network Parameters */
  ether.printIp("IP:  ", ether.myip);
  ether.printIp("GW:  ", ether.gwip);
  ether.printIp("DNS: ", ether.dnsip);
}

void loop(){
  /* Send Script to IP Address */
  if (ether.packetLoop(ether.packetReceive())) {
    memcpy_P(ether.tcpOffset(), page, sizeof page);
    ether.httpServerReply(sizeof page - 1); }}
```

Figure 7.1.12 HTML Script to IP code

B) Open the Receiver Arduino Serial monitor

C) Wait for the ethernet module to initialize and output the Arduino's Assigned IP address

D) Write the Assigned IP address in an internet browser

How to collect data:

The data to be collected is the output of the HTML script in the IP address assigned to the Arduino

4. The expected result is that as the Ethernet controller would initialize successfully and the HTML script written in the code will be executed by the browser and the text written in it would appear in the IP Address

7.1.13 Test Experiment case 7: Send Pin Value via HTML to IP address / Ahmed El Gabaly

1. The objective of this experiment is to test sending the receiver Arduino analog pin value via HTML to the Receiver Arduino IP address

2. Setup: ENC28J60 - Receiver Arduino pinout

ENC28J60	Receiver Arduino
VCC	3.3v
GND	GND
SCK	pin 13
SO	pin 12
SI	pin 11
CS	pin 8

Table 7.1.13 ENC28J60 - Receiver Arduino pinout

3. Experimental procedure:

A) Program the receiver Arduino with the HTML Ethernet code

```
// HTML Script
String HTML =
"HTTP/1.0 503 Service Unavailable\r\n"
"Content-Type: text/html\r\n"
"Retry-After: 600\r\n"
"\r\n"
"<html>"
  "<head><title>"
    "Service Temporarily Unavailable"
  "</title></head>"
  "<body>"
    "<h3>Print Analog Pin Online</h3>"
    "<p><em>"
      "Analog Pin Value =     <br />"
    "</em></p>"
  "</body>"
"</html> \0";
char page[100] = " ";
void setup(){
  Serial.begin(57600);
  Serial.println("\n[backSoon]");
  /* Initilize Ethernet controller */
  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
    Serial.println( "Failed to access Ethernet controller");
  /* DHCP Request */
  if (!ether.dhcpSetup()) Serial.println("DHCP failed");
  /* Print network parameters */
  ether.printIp("IP:  ", ether.myip);
  ether.printIp("GW:  ", ether.gwip);
  ether.printIp("DNS: ", ether.dnsip);
  // Add pin value to HTML Script
  int val = analogRead(3);
  int pos = HTML.indexOf('=');
  for(int i = 0; i < String(val).length(); i++) {
    HTML.setCharAt((pos + 2) + i, String(val).charAt(i)); }
  //Add string to char array
  unsigned int length = HTML.length();
  HTML.toCharArray(page,100); }
void loop()
{ /* Send HTML script to IP Address */
  if (ether.packetLoop(ether.packetReceive())) {
    memcpy_P(ether.tcpOffset(), page, sizeof page);
    ether.httpServerReply(sizeof page - 1); }}
```

Figure 7.1.13 HTML Script to IP code

How to collect data:

The data to be collected is the outptut of the HTML script in the IP Address assigned to the

Receiver Arduino

4. The expected result is that as the Ethernet controller would initialize successfully and the HTML script written in the code will be executed by the browser and the text written in it would appear containing the Receiver Arduino Analog pin in the IP Address.

7.2 Bug Tracking

A database is used to track the defects found while performing the test cases. All defects will be logged as they are discovered and will be assigned to Person A to fix, or to Person B to investigate.

Test Case	Defect	Person to Fix
1	Too Large Accelerometer Readings	Moheb Amini
2	N/A	N/A
3	N/A	N/A
4	The format of the CSV file needs to be changed	Ahmed Elgabaly
5	Inconsistent results	Mark Logston
6	N/A	N/A
7	N/A	N/A
8	Ethernet modules not initialized	Ahmed Elgabaly
9	N/A	N/A
10	N/A	N/A
11	N/A	N/A
12	N/A	N/A
13	HTML is not suitable for sending the Arduino Uno Analog pin value to the IP address	Ahmed Elgabaly

Table 7.2 Bug Tracking

7.3 Quality Control

The completed test cases are reviewed to ensure that all cases were run, completed successfully; and that any deviations from the test cases are noted accordingly. Each step is marked as Passed or Failed. Failed cases should be marked with the date and time of the failure, and the associated test track number. When a failed case is fixed, the date and time of the retest should be noted

Test Case	Result	Date of Failure	Date of Retest
1	Success	N/A	N/A
2	Success	N/A	N/A

3	Success	N/A	N/A
4	Failure	Winter week 4	Winter Week 4
5	Failure	Winter week 5	Winter week 5
6	Success	N/A	N/A
7	Success	N/A	N/A
8	Failure	Winter week 6	Winter week 6
9	Success	N/A	N/A
10	Failure	N/A	N/A
11	Success	N/A	N/A
12	Success	N/A	N/A
13	Failure	Winter week 3	N/A

Table 7.3 Quality Control

7.4 Identification of critical components

Accelerometer: We had to make sure that it is not turned on by noise, when the object is not moving, a filter had to be applied to accurately identify when the object is actually moving.

FFT: We had to make sure the FFT conversion was accurate so we can detect the correct frequency from the transmitter.

Server: We had to make sure the CSV file in the server had the rooms and the objects as columns instead of rows for the GUI code to be able to correctly read it.

Server Response time: We had to make sure that the server is given enough time to respond to the device detection and to update the database as fast as possible before a new transmitter passes under the receiver, the minimum response time needed was 400 ms

The IR Photo transistor: It has to be calibrated to a position where it can accurately detect the incoming signal from the transmitter module.

The Ethernet module: Each Ethernet module in a room must be assigned a unique mac address to avoid overlap of the devices on the network which will cause problems in communication to the server.

7.5 Items Not Tested by the Experiments

Test experiment case 3

Does not test how close the frequencies generated in the spectrum are to each other because this is not needed as the frequencies chosen for the moving objects are at least 250 Hz apart so the chance of overlap in the spectrum is minimal.

Test experiment case 7

Does not test static IP setup of the Ethernet connection to the server because 95% of the time the DHCP server responds to the DHCP request and assigns an IP address to the newly activated device on the network. In addition, the code will resend the DHCP request in case it fails.

Test experiment case 10

Does not test the detection of the leds with photo diodes with different angles of detection because the one we have had an approximate angle of detection of 30 degrees which was suitable for our application.

Test experiment case 11

Does not test the system in an actual hospital setting where the system has to detect hundreds of devices simultaneously moving between rooms. We only tested the system with a maximum of three receivers and two transmitters for demoing purposes and as a proof of concept.

8 Experimental Results and Test Report

8.1 Test Iterations

8.1.1 Experiment 1 – Iteration 1

Moheb Amini

1. The x, y, z values are successfully read from the Accelerometer.
2. These results meet the expected results for the experiment.
3. The x, y, z values are acquired from the Accelerometer but are too large for our application.
4. Action: Find a divisible number that make the range more readable and accurate.

8.1.1.1 Experiment 1 – Iteration 2

Moheb Amini

1. The x, y, z values are successfully read from the Accelerometer and are at a readable more accurate range
2. These results meet the expected results for the experiment
3. The x, y, z values acquired from the accelerometer are small enough to meet the requirements of the application
4. No corrective action taken

8.1.2 Experiment 2

Moheb Amini

1. The x, y threshold value is successfully chosen and is 0.13 units.
2. This Result meets the expected results.
3. The threshold is accurate and stable; when the object moves, the values read from the Accelerometer are greater than the threshold and therefore the PWM signal is sent to the Infrared led.
4. No corrective action needed.

8.1.3 Experiment 3

Moheb Amini

1. The Results are that the amplitude of the desired frequency is the highest in the spectrum

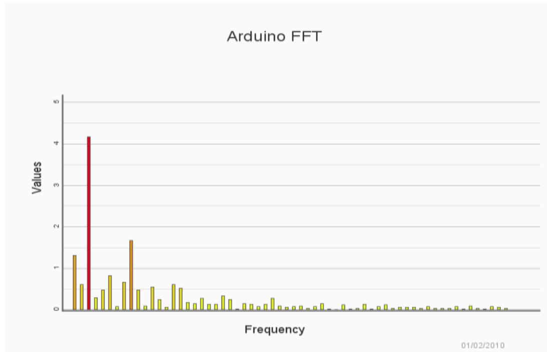


Figure 8.1.3 FFT Spectrum

2. This Result meets the expected results
3. The frequency used in the experiment was 500 Hz and as the show in the figure by the red line, the FFT does work because the highest amplitude corresponds to a frequency close to 500 Hz.
4. No corrective action needed.

8.1.4 Experiment 4 – Iteration 1

Mark Logston

1. The wrong data is retrieved from the server.
2. This Result is contrary to the expected results
3. When the CSV read command is executed in the GUI code, it results in an error
4. Action: Reverse the order of the columns in the CSV file on the database.

8.1.4.1 Experiment 4 – Iteration 2

Mark Logston

1. The latest new data on the server is successfully retrieved from the server.

Figure 8.1.4.1 Server Data

2. This Result is meets the expected the results

3. When the CSV read command is executed in the GUI code, it shows the latest uploaded the CSV file on the server which proves that the GUI can always see the latest new object locations updated in the server.

4. No corrective action needed

8.1.5 Experiment 5 – Iteration 1

Mark Logston

1. The voltage gained from the amplifier circuit is either not amplified at all and also, different every time.
2. This Result is contrary to the expected results
3. The voltage read on the oscilloscope should be within a stable range every time it is tested and varies a lot with height.
4. Action: Change to Infrared phototransistor circuit.

8.1.5.1 Experiment 5 - Iteration 2

Mark Logston

1. The Results obtained are consistent when the transmitter is within around 8 feet from the receiver
2. These results meet the expected results for the experiment
3. The voltage received from the phototransistor is always stable and consistent at around 5 volts and form a distance at around 8 feet from the transmitter which fits the application requirements
4. No corrective action taken

8.1.6 Experiment 6

Mark Logston

1. The new added object appears in both the GUI object list and the database object list.
2. This Result is meets the expected result.
3. The new object added in with the update info button in within the GUI is immediately added to the GUI object list and the database object list which proves that adding a new object this way works.
4. No corrective action needed.

8.1.7 Experiment 7

Ahmed Elgabaly

1. The Ethernet module is successfully initialized

```
FFT pins Initialized
Dislpay Room 1 Devices

Accessed Room 1 Ethernet controller
Sent DHCP Request
DHCP Request Succeeded
Please Wait, Resolving Server IP....
Server IP resolved

Room 1 Arduino IP: 169.235.29.227
Netmask:         255.255.254.0
Gateway:         169.235.28.1
DNS:             138.23.169.10
Server IP:       185.27.134.227
```

Figure 8.1.7 Ethernet Initialization

2. This Result is meets the expected result.
3. The DHCP request is a success, the receiver Arduino is assigned a unique IP address on the server and the receiver Arduino is ready to send the incoming devices to the server.
4. No corrective action needed.

8.1.8 Experiment 8 – Iteration 1

Ahmed Elgabaly

1. The Ethernet modules failed to initialize.
2. This Result contradicts the expected result.
3. The Ethernet modules failed to initialize and therefore the Arduinos are not connected to the internet
4. Action: Give each Ethernet module a unique MAC Address

8.1.8.1 Experiment 8 – Iteration 2

Ahmed Elgabaly

1. The pin values from both Arduinos to the server.


PinValues (34) - Notepad
File Edit Format View Help
389
388
389
388
634
389
387
387
387
389
389
388
388
385
390

Figure 8.1.8.1 Server Data from 2 Arduinos

2. This Result meets the expected result.
3. The Ethernet modules were successfully initialized and therefore the Arduinos successfully sent pin values to the server.
4. No corrective action needed.

8.1.9 Experiment 9

Ahmed Elgabaly

1. The Infrared Led was not interrupted when the RGB Led was turned on.
2. This Result meets the expected result.
3. The RGB led only turns on when the object is stationary and the Accelerometer x,y values are below the threshold which means that the Infrared Led does not turn on and therefore is not uninterrupted.

4. No corrective action is needed.

8.1.10 Experiment 10

Moheb Amini, Mark Logston, Ahmed Elgabaly

1. The results for all tested Leds are as follows

LED Color	Wavelengths (nm)	Bias (mV)	Voltage Read (mV)	Gain (mV)	Distance (in)	Horizontal Distance (in)
Infered	850	1.25	5.12	3.87	18	
		1.35	1.62	0.27	67	
		1.36	1.37	0.01	291	146.5
Red	610-760	1.14	2.42	1.28	17	
		1.16	1.24	0.08	58	
		1.33	1.36	0.03	86	
		1.34	1.35	0.01	147	74.03
Orange	590-610	1.17	1.54	0.37	17	
		1.28	1.32	0.04	47	
		1.23	1.24	0.01	92	46.33
Yellow	570-590	1.22	1.48	0.26	19	
		1.3	1.32	0.02	58	
		1.24	1.25	0.01	86	43.3
Green	500-570	1.19	1.56	0.37	18	
		1.34	1.36	0.02	51	
		1.32	1.33	0.01	96	48.34
Blue	450-500	1.35	2.2	0.85	18	
		1.46	1.5	0.04	58	
		1.55	1.56	0.01	101	54.89
UV/Purple	375	1	1.13	0.13	18	
		1.18	1.2	0.02	58	
		1.23	1.24	0.01	88	44.32
Pink	Multiple Types	1.07	1.78	0.71	18	
		1.18	1.2	0.02	58	
		1.1	1.11	0.01	101	50.86
White	Broad Spectrum	1.29	2.24	0.95	16	
		1.45	1.5	0.05	57	
		1.4	1.41	0.01	126	63.45

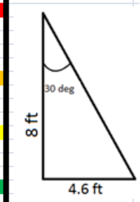


Figure 8.1.10 Led Detection range results

2. This Result contradicts the expected result
3. The Infrared Led gives the highest voltage gain and the longest distance at which this gain is achieved.
4. The Led that would be used in the project will be the Infrared led.

8.1.11 Experiment 11

Moheb Amini, Mark Logston, Ahmed Elgabaly

1. The Results for the full system are as follows

```
No Device Detected
No Device Detected
No Device Detected
500 Hz Device Detected
500 Hz Device Detected
sent to website... No response
 Resending Device: sent to website... OK
No Device Detected
No Device Detected
No Device Detected
```

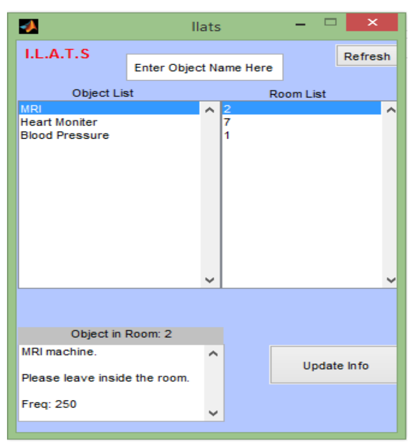


Figure 8.1.11 Left (Server Response) Right (GUI Output)

2. This Result meets the expected result.
3. When the object moves under the receiver at normal walking speed, the device detection is shown by the GUI and by the receiver Arduino serial monitor
4. No corrective action is needed.

8.1.12 Experiment 12

1. The Script was successfully executed on the browser

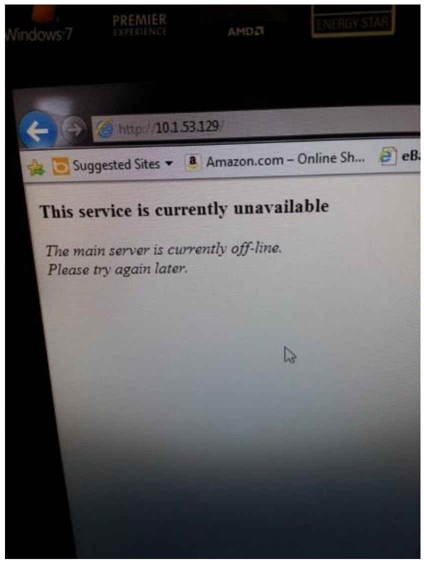


Figure 8.1.12 Receiver Arduino IP Address Output

2. This result meets the expected result
3. When the IP address assigned to the Receiver Arduino by the DHCP server is entered in the internet browser, the HTML script that is sent to the IP Address is executed and the String written in it appears in the IP address page.
4. No corrective action is needed.

8.1.13 Experiment 13

1. The Script was unsuccessfully executed on the browser

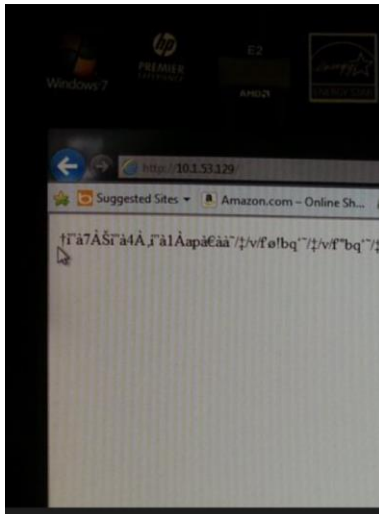


Figure 8.1.13 Receiver Arduino IP Address Output

2. This result contradicts the expected result
3. When the IP address assigned to the Receiver Arduino by the DHCP server is entered in the internet browser, the HTML script that is sent to the IP Address is executed and the String written in it does not appear in the browser and the analog pin value is not sent to the IP Address
4. Action: instead of using Arduino as the server programmed with HTML, a remote server will be used and programmed with PHP

9 Conclusion and Future Work

9.1 Conclusion

In conclusion the project was a success. For all of the issues that were brought out over the course of this project's lifetime, I.L.A.T.S functions correctly and well. The project ended up meeting all the requirements that we set for this project. Moheb was able to read the values from the accelerometer so that almost any movement would cause the Infrared LED to start transmitting a signal. Mark was able to create a circuit that gave a distance of over 8 feet from the transmitter to the receiver. Moheb was able to create an FFT function that would extract the frequency that was being received from the receiver and send what object was being sensed. Ahmed was able to create a server that would be able to respond underneath the 400 milliseconds so that the server would be updating in real time. Mark was able to create a GUI that could update based on the server changing. Because of limitations of MATLAB, the GUI was not able to update in real time, but a button was placed that will update the GUI with the correct information when needed. The circuits created ended up having a power consumption of 0.506 μW.

Ahmed El Gabaly:

This project taught me that it is not about learning and being limited to the field of electrical engineering but that I should always be flexible to learn new things and expand my knowledge. I used my embedded systems knowledge to learn about programming an embedded system to communicate with an online server and I also used the knowledge I learned about C++ and C to easily learn PHP to program the server, I also learned how to Read the voltage of a battery of a microcontroller within the software which was something new to me.

Moheb Amini:

This project taught me how to compute the Fourier transform within a microcontroller rather than computing it by hand. I was able to further my knowledge of i2c and execute it in order to communicate with an accelerometer and manipulate its values in order to make use of it for the project. I learned the art of design and implementation of both PCBs and 3D models for the purposes of this project. I learned the benefits and also the hardships of working with a team for a technical project.

Mark Logston:

From this project I learned how to analyze data and make small changes to get the correct response needed. I also learned how to control the environments that we worked in so that the project would move forward. I learned how to use MATLAB to create a GUI and how to use the GUI to communicate with a server. I learned how to step up to the plate when parts of the project started to slip so that our project would move towards completion.

9.2 Acknowledgement

Professor Liang:

For giving us the idea of the project and giving us weekly advice based on our progress

Professor Chen:

For leading us in the right direction of switching UV light for IR light and for ideas on amplification of the photodiode received signal

Professor Chomko:

For helping us with determining methods of building circuit oscillators

Andrew Olguin:

For helping with the understanding of the Cooley-Tukey FFT method

Russell Perry:

For giving us access to a room for our testing and demoing of the project

Dr. Hossny El-Sherief:

For editing and giving feedback on the Rough Draft of the Design Report.

10 References

1. http://www.arduino.cc/
2. https://github.com/jcw/ethercard/blob/master/README.md
3. http://nathanhein.com/2013/02/getting-arduino-online-with-an-enc28j60/
4. http://www.lucadentella.it/en/category/enc28j60-arduino/
5. http://dlnmh9ip6v2uc.cloudfront.net/datasheets/BreakoutBoards/39662b.pdf
6. http://www.php.net/manual/en/index.php
7. http://www.byethost.com/
8. http://forum.arduino.cc/index.php/topic,38119.0.html
9. http://jeelabs.org/2012/05/04/measuring-vcc-via-the-bandgap/#comments
10. http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA745xL
11. http://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm
12. Matlab GUI tutorials found within Matlab help functions
13. http://www.suntekstore.co.uk/product-14001427-20pcs_200mw_10mm_infrared_ir_1-chip_led_night_vision_850nm.html
14. http://www.taydaelectronics.com/sensors-transducer/optical-sensor/tops-050-photo-transistor-900nm-5mm-radial-tops-050tb2.html

11 Appendices

Appendix A: Parts List

Item Bought	Cost of Item ($)	Amount bought
Arduino Uno R3	**34.95**	**6**
Zitades ENC29J60 Ethernet LAN Network Module	**11.99**	**4**
Teensy 3.0 USB Development Board	**19**	**3**
Teensy 3.0 USB Development Board with Pins	**22**	**2**
10mm Infrared IR 1 Chip LED Night Vision 850nm	**7.87**	**1**
Wire, Headers, and Pins	**15.36**	**1**
SainSmart MMA7455 Angle Sensor Accelerometer Module AVR ARM MCU	**11.79**	**2**
Pins (header), 14 pins, 0.1 inch (2.54 mm) spacing	**6**	**1**
Ethernet Cables	**16.99**	**2**
Grand Total:	**466.81**	

Appendix B: Equipment List

Equipment List
Laptop
Server

Appendix C: Software List

Software List
Microsoft Windows (XP or above)
Matlab
Arduino IDE
Processing