UPEI SCHOOL OF MATHEMATICAL AND COMPUTATIONAL SCIENCES

CS2710 FINAL PROJECT – ARDUINO WHEATHER STATION

BY

Samuel Nnam (135228) and Nathan Munn (144212)

December 16th, 2017

GitHub link:
https://github.com/The3Null4Player613310/ArduinoWeatherStation

TABLE OF CONTENTS

1.0 Introduction

The following statement is an introduction to highlight the problem that team has been faced with, as well as an introduction to the end goals the team has set to accomplish. The team's client, Professor Chris Vessey a computer science instructor at the University of Prince Edward Island. The team came with an idea of creating a platform called Arduino Weather Station using Arduino system which will help the individuals living in Charlottetown, Prince Edward Island to stay safe during winter through reading the temperature and humidity updating the weather condition through social media (twitter). The major idea was based on getting two Arduino UNO's using Wi-Fi shield at different locations to collect data using Temperature and Humidity sensor and send the collected data to an Arduino Mega which uses the Ethernet shield to collect and save the data to an SD card then the collected data will be sent out through a social media server (twitter) using a twitter token code. A twitter account was created as a communication means where the collected data is sent out to the society for a weather condition update.
The team was able to carry out the above tax through setting up two Arduino UNO's using a Wi-Fi shield, Temperature and Humidity sensor and CdS light sensor at a different internet location (that is, **not** on the same network-two different locations). The two Arduino UNO's was used as a client which we called **Node.** The node functionality was based on data collection and sending out the collected data to a second system which stores the collected data.
The second system was referred to as the server with a setup using the Arduino Mega which consist of the Ethernet shield, RTC, LCD Display, an Mp3 player as part of the connections. The server functionality was based on getting the data from the **Node** and storing the data and sending out the stored data to the social media server every 5 minutes; that is, Using the twitter account to send out the current temperature, average current temperature, current humidity and average humidity at a given location.

2.0 Problem Statement

The society needs a weather station updater that automatically update the weather situation every 5 minutes which will save the people time from watching television but rather use social media to get update about current weather condition since social media is common in the society and the weather station will help the people to stay safe during the winter season.

3.0 Components

The materials that are required for this project includes:

- 2 X Arduino UNO
- 1 X Arduino Mega
- 1 X Ethernet/SD Shield

- 2 X ESP8266 Wi-Fi Shield
- 1 X Real-Time Clock module (RTC)
- 2 X microSD card
- 2 X LCD Display (module 16x2)
- 1 X LCD Display (module 20x2)
- 1 X IR Remote
- 2 X Cds photocell
- 1 X Mp3 player
- 2 X DHT-11 Sensor
- 1 X Breadboard

4.0 Components Functionality

The functionality for the components are:

- **Arduino UNO** is a microcontroller board which consist of 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHZ quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It also contains everything needed to support the microcontroller [1].

- **Arduino Mega** is a microcontroller board which consist of 54 digital pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHZ crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It also contains everything needed to support the microcontroller [1].

- **The Arduino Ethernet** Shield R3 (assembled) allows an Arduino board to connect to the internet. It consists of micro SD card connector and supports up to four simultaneous socket connections. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top. Arduino uses digital pins 10, 11, 12, and 13 (SPI) to communicate with the W5100 on the ethernet shield. These pins cannot be used for general i/o [2].

- **ESP8266 Wi-Fi Shield** module is industrial-grade chips ESP8266, which is ESP-12E with metal shield with strong anti-interference ability. The shield is pin-compatible with Arduino UNO, Mega2560 and other control board. A voltage converter chip is used to deal with 3.3V (ESP8266) and 5V (Arduino); Dual DIP switches is used for serial ports so that this module shield can be used alone as an Arduino UNO expansion board, also be used as esp8266 expansion board; Serial data is transported to Wi-Fi device transparently [3].

- **Real-Time Clock module (RTC)** is a simple breakout board for the DS1307 real-time clock which accurately keep track of seconds, minutes, hours, days, months, and years for almost a decade.

- **MicroSD card** helps to store the collected date for proper analyzation.

- **LCD Display** is a flat-panel display that uses the light modulating properties of liquid crystals. Light crystals do not emit light directly, instead using a backlight or reflector to produce images in colors [4].

- **IR Remote** Infrared remote is used to wirelessly control the system
- **Cds photocell** is basically a resistor which changes the resistance according to the amount of light falling on its surface.
- **Mp3 player** uses an SD micro card and it serves as an output for audio file.
- **DHT-11 Sensor** is used for reading the temperature and humidity of specific location.
- **Breadboard** helps with the connection of all the components.

5.0 Node A and B functionality

The nodes (A and B) was used as our client which was set up at different locations (that is with different internet connections) to collects the temperature and humidity data at a specific location.
The nodes serve as a simple data gathering system with three data commands and two **commands** as specified below:

- get temp: Gets temperature in Celsius
- get humi: Gets humidity (unknown units)
- get lumi: Gets a relative value of light using Cds photocell for reference purposes
- ack: Acknowledge command that is sent for when a data is received successfully
- nack: Not-Acknowledged command sent when parity has failed or if the get command returns unexpected result/value.

During the coding process, some **difficulties** was encounter as mentioned below;

- The DHT11 sensor that was been used failed to record temperature that was blow 0 because the sensor was designed to have a temperature range of 0 -50 degrees Celsius which was created a huge problem for the group but we tried fixing the temperature range problem by adding Lm35 temperature sensor because is have a sub-zero temperature reading since the LM35 is roughly proportional to the DHT11 sensor but down the road we figured out that LM35 sensor was returning a weird value for us which was way off from the current temperature so we decided to go with just the DHT11 sensor since it returns better value for us except the fact is that it doesn't read negative temperature but we decided to just go with room temperature reading.

- Memory over flow was another big issue that we had during the process; that is, we discovered that the node couldn't handle a very long string and that was fixed through finding a reference material which helped with finding the number of free bits of memory in the Arduino and made a leaky of this so that they wouldn't overflow the memory.

6.0 Connection process / Building for Node A and Node B

The connection process was one of the major steps that was carried out during this project and this was done by connecting all the components needed for the node functionality with a current supply of 5V from the Arduino UNO and Connecting the Ground (GND) pin from the Arduino UNO. The diagram below represents the Breadboard connections and schematics for both Node A and B.

Proper functionality of the system is the most important part of the project and the team observed that testing the system outside when the temperature is really cold is likely to affect the functionality of our system, so the team made a 3D printed case using the solidworks program for the two nodes which was used to reduce the effect of the temperature. The diagram below shows the 3D printed case for the system:

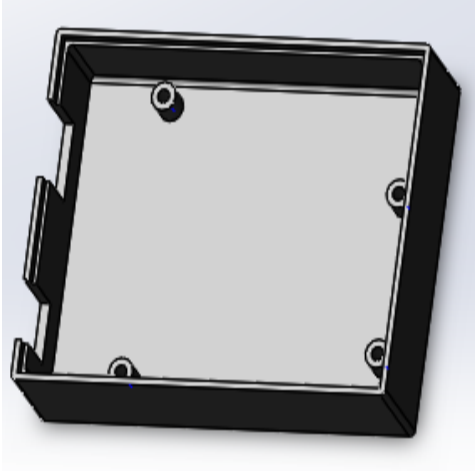

Figure 1: Un-assembled Node Case

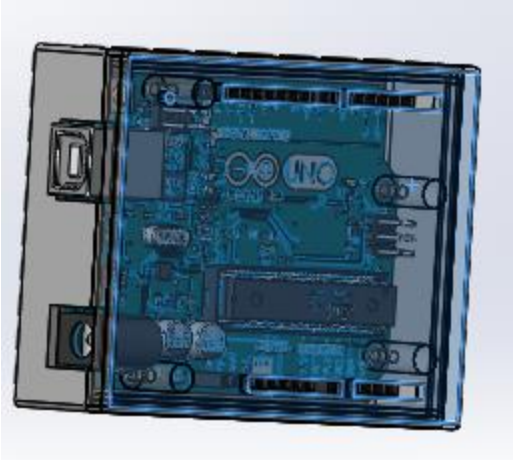

Figure 2: Assembled Node Case

Figure 3: Node Case 3D-Print Inner View

Figure 4: Node Case 3D-Print Outer View

During the building process, we encountered **difficulty** as described below;

- With the aid from the engineering department, the team made a 3D printed case for the two nodes for protection of the system during harsh weather condition, but we discovered that there was an error after attaching the Wi-Fi shield which lead to little room in the case, but we fix the room space problem by drilling holes of the case to allow or give us more space as shown below:

Figure 5: Node System Attachment

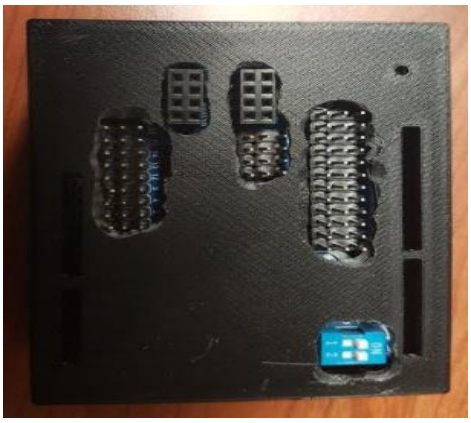

Figure 6: Node Case Modification

The team used fritzing program to display proper connection for the node system and this include both the breadboard connections and schematics of the node system. The breadboard connections and schematics for the node system is represented as shown below:

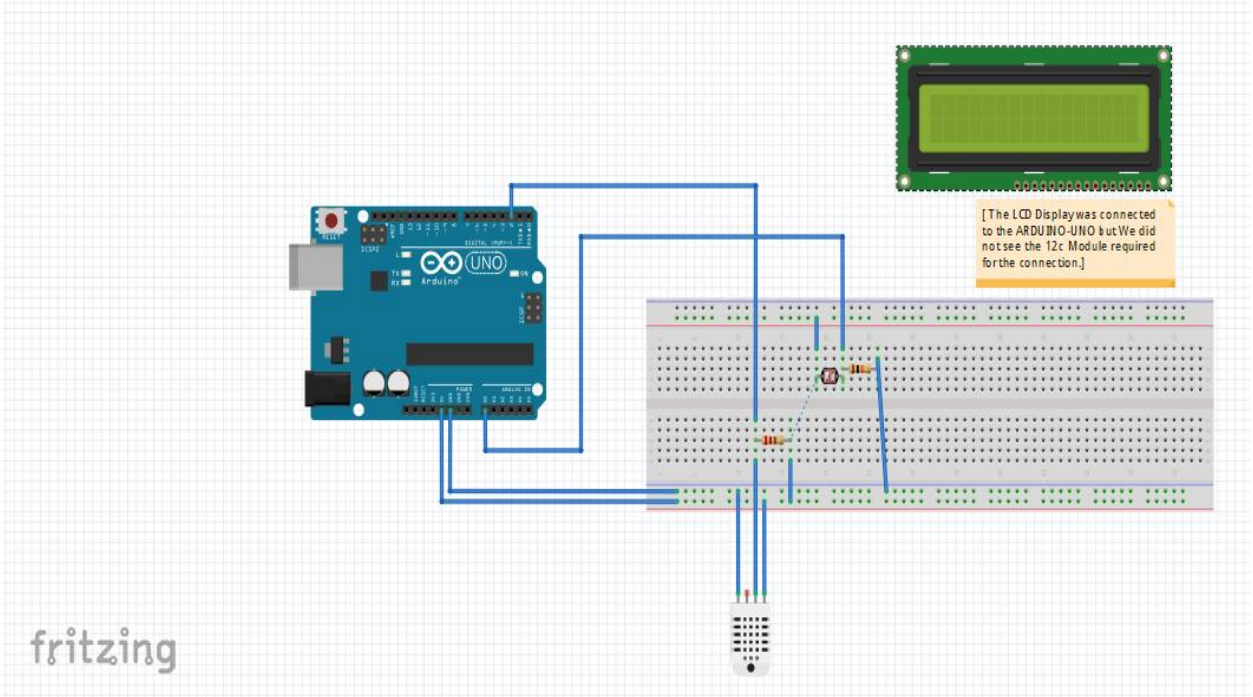

Figure 7: Breadboard connection for Node A and B

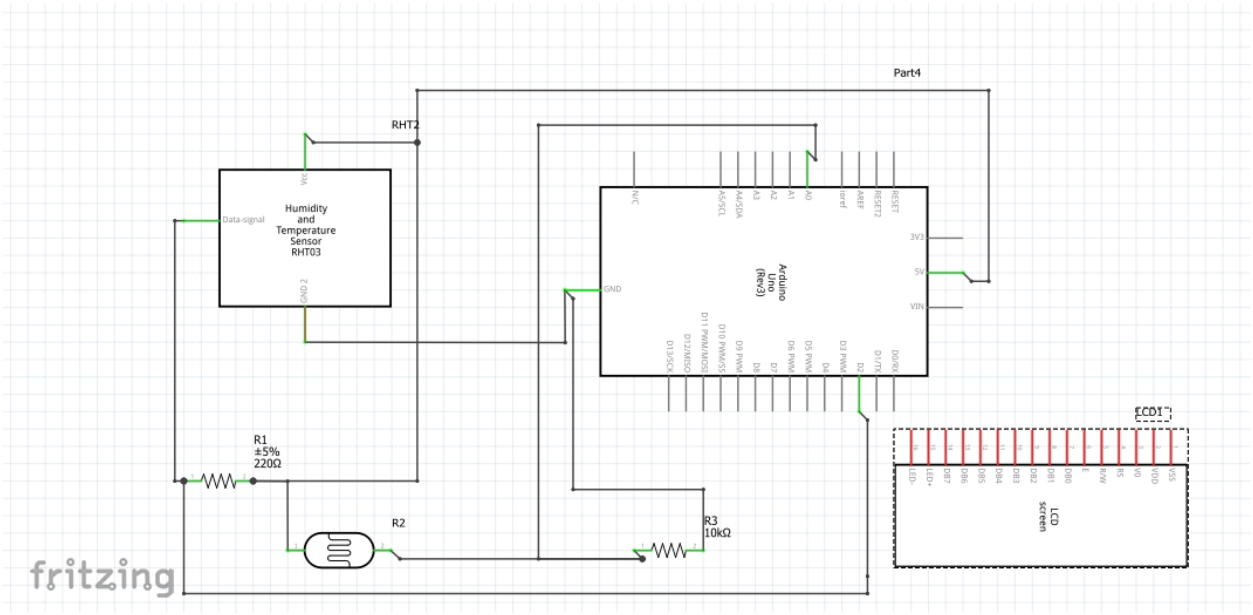

Figure 8: Schematics of Node A and B.

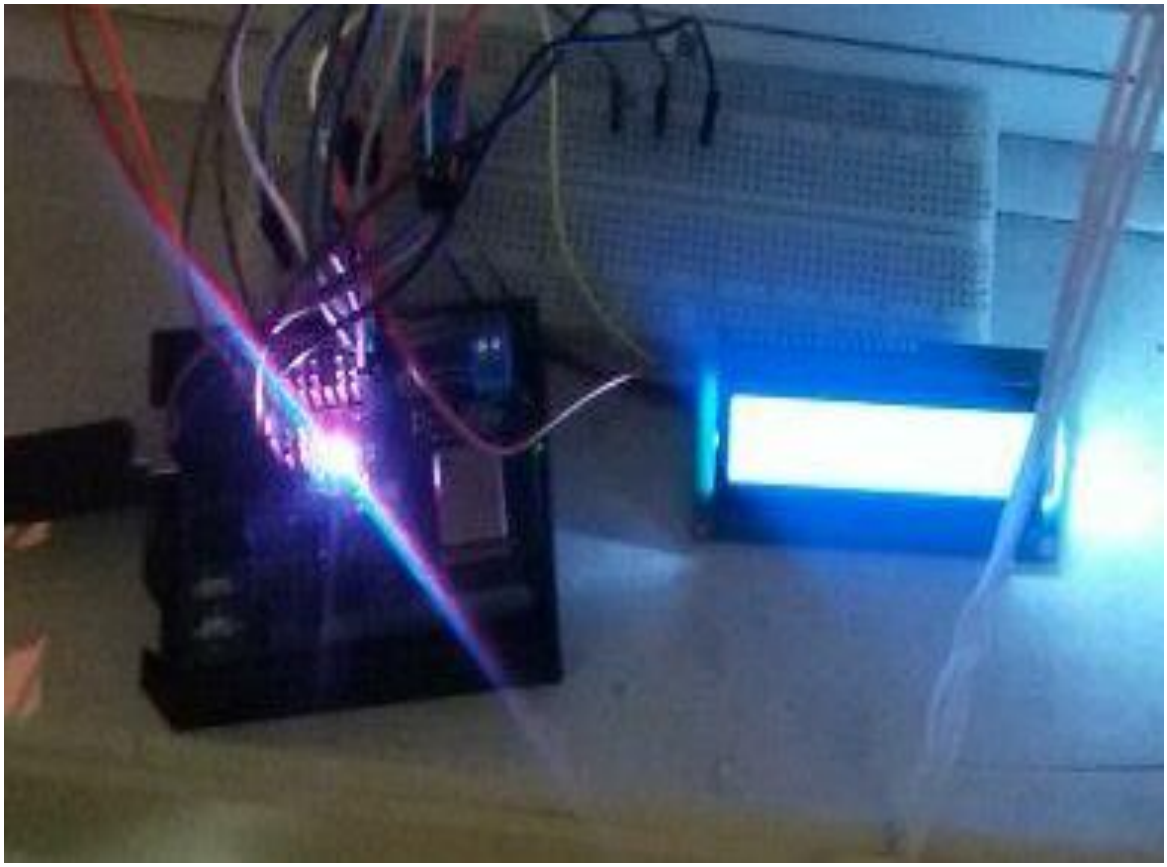

Figure 9: Pictorial Connection Of The Nodes.

7.0 Server Functionality

The server was the major system which we referred to as the coordinator because it requests the data; that is, it requires the temperature and humidity data at two different locations from the two nodes and after it collects the data, it stores the data to an SD Card and it doesn't just stores the data, but it sends out the collected data through twitter using the twitter token. The team created twitter account that helped to successfully send out the weather situation (Temperature and Humidity of two different locations in Charlottetown) to the people living both on PEI and adventurers coming to PEI.

The server also uses MP3 player to say the current temperature, current humidity and current time (using the RTC to keep track of time).

The server gathers the data from the two nodes using three data commands and two **commands** as specified below;

- get temp: Gets temperature in Celsius
- get humi: Gets humidity (unknown units)
- get lumi: Gets a relative value of light using Cds photocell for reference purposes
- ack: Acknowledge command that is sent for when a data is received successfully
- nack: Not-Acknowledged command sent when parity has failed or if the get command returns unexpected result/value.

The server also sends out the minimum and maximum temperature data every 5 minutes. The diagram shown below shows the sample of the server sending out the information through twitter:

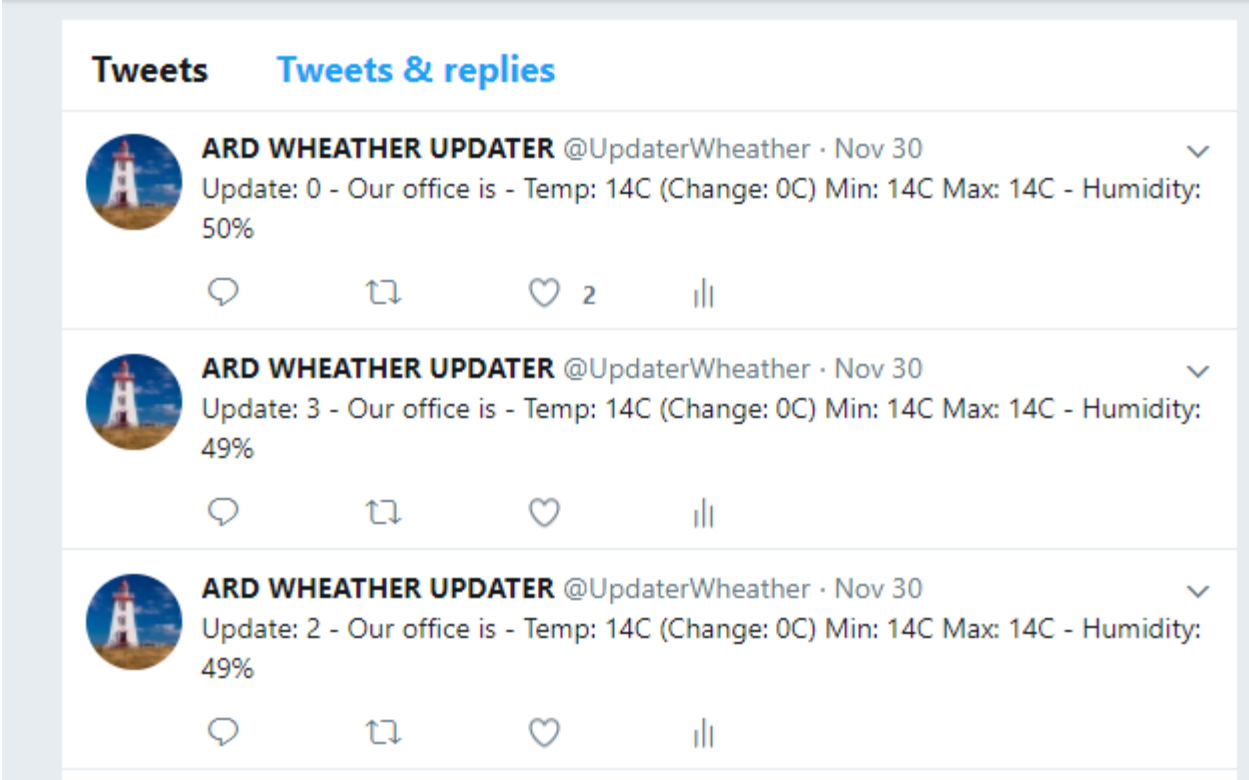

Figure 10: Server Sending Out Data Sample.

During the coding process, some **difficulties** was encounter as mentioned below;

- The team spent so much time trying to fix a major error that occurred which was the RTC unit been used reading values that were not precise but were mostly accurate and this was fixed through creating three variables to make sure that the values of the RTC were not weird values.
- Another difficulty was getting the Arduino Mega to use the SD Card to store the collected data but that wasn't successful, so the team decided to implement the MP3 player SD Card file to solve the memory issue which was successful.
- Another difficulty was getting the MP3 to use a library because in the course MP3 library wasn't provide so we spent so much time to figure out a library that will work and we did find something.

8.0 Connection process / Building for Server

The connection process was one of the major steps that was carried out during this project and this was done by connecting all the components needed for the node functionality

with a current supply of 5V from the Arduino UNO and Connecting the Ground (GND) pin from the Arduino UNO. The diagram below represents the Breadboard connections and schematics for the server.

The team used fritzing program to display proper connection for the server system and this include both the breadboard connections and schematics of the server system. The breadboard connections and schematics for the node system is represented as shown below:

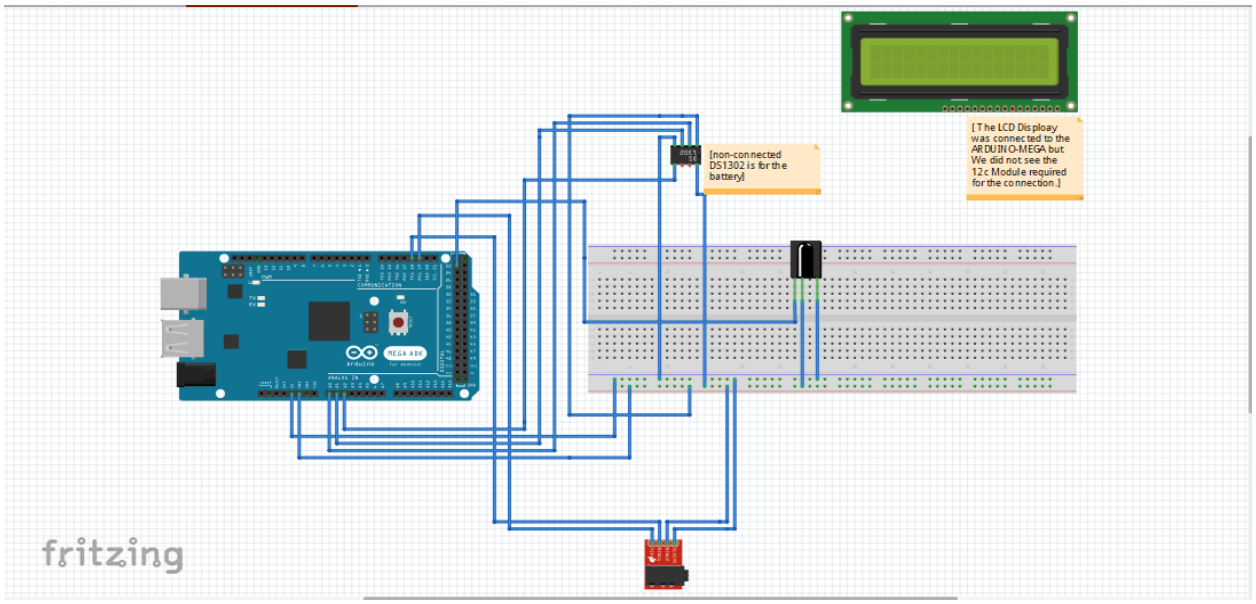

Figure 11: Breadboard connection for Server System

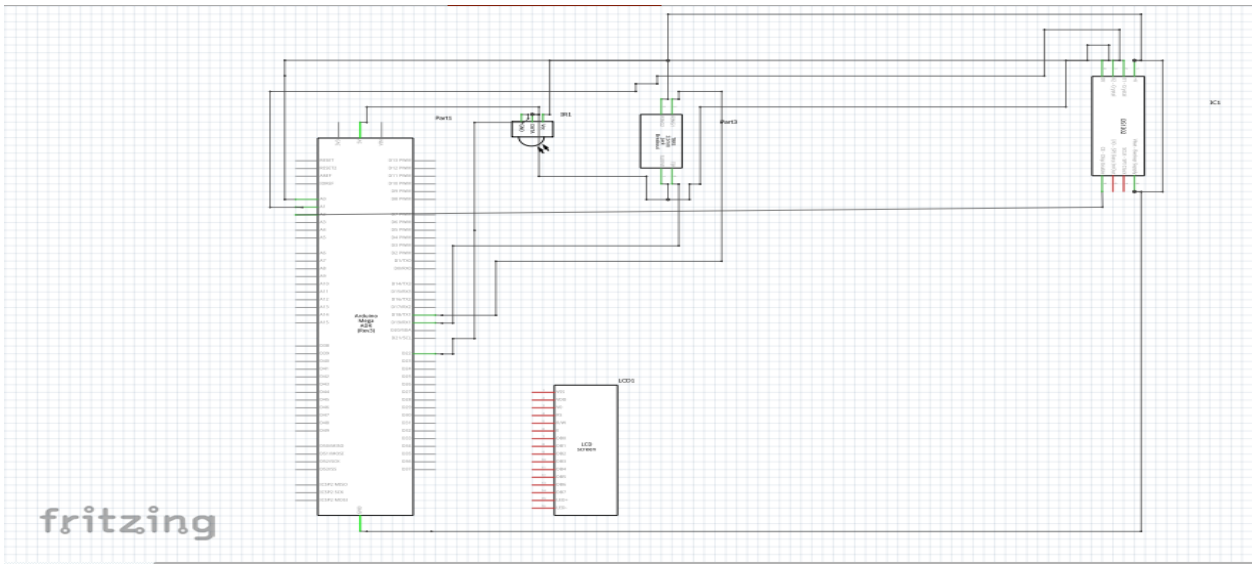

Figure 12: Schematics of Server System.

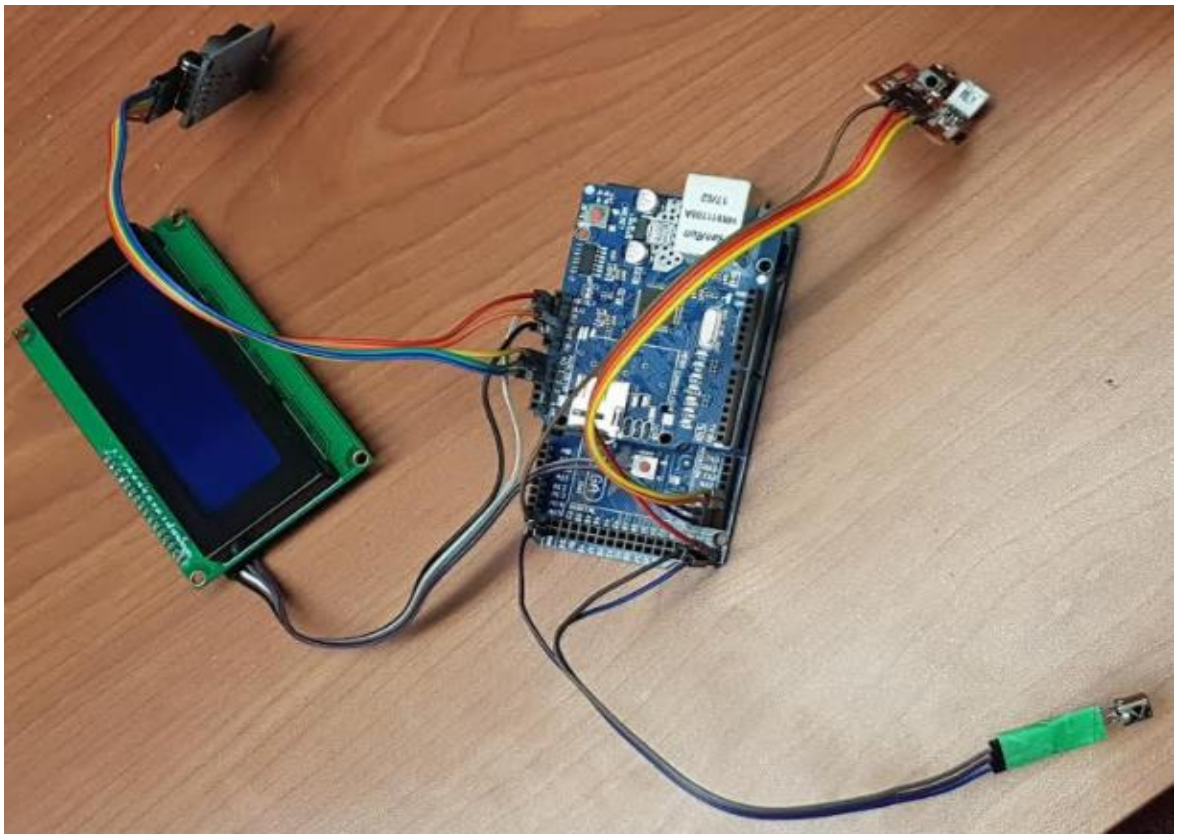

Figure 13: Pictorial Connection Of The Server.

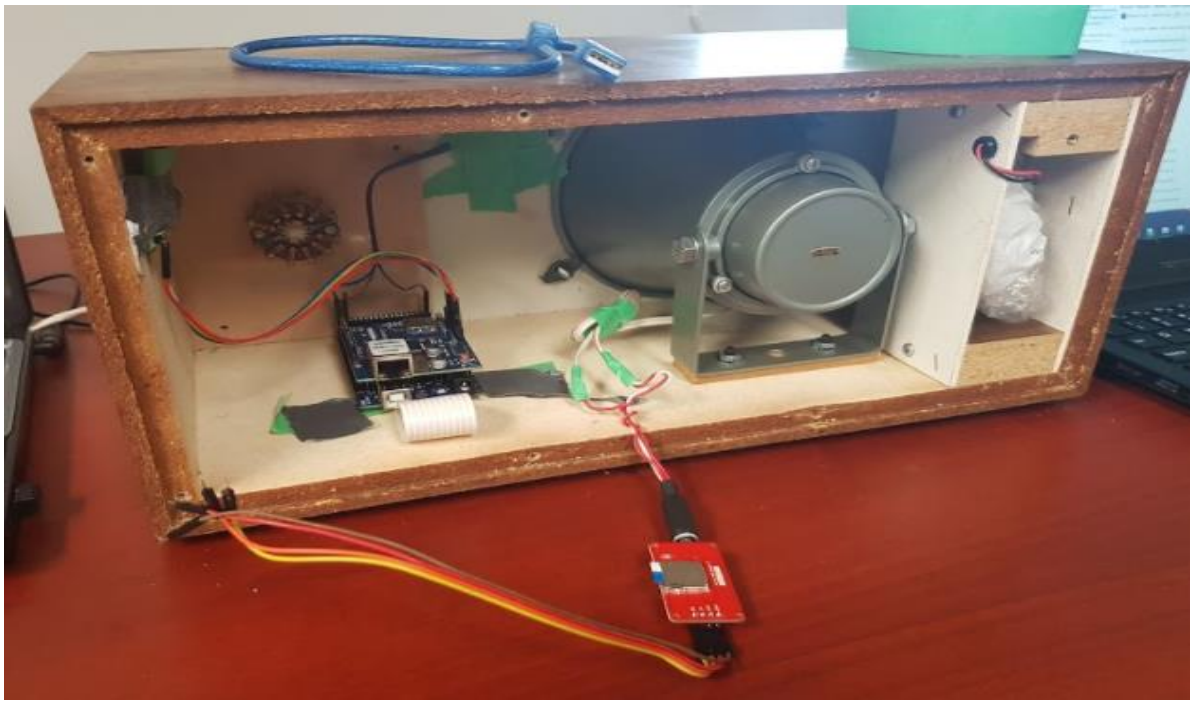

Figure 14: Built Case For The Server

9.0 Code

The attached file below was the written code for the system:

FINAL PROJECT.UNO

NOTE: The code is attached as part of zip file of the document.

10.0 Conclusion

The society safety is the major priority of this project that was successfully carried out even though we had so much difficulties but as shown from the sample above and the attached video we were able to get things working.

11.0 Reference

[1]. “Adafruit”, https://store.arduino.cc/usa/arduino-uno-rev3 [Accessed: 16- Dec-2017]
[2]. “Arduino UNO”, https://www.adafruit.com/product/201 [Accessed: 16- Dec-2017]
[3]. “Tindie”, https://www.tindie.com/products/doit/arduino-uno-r3--esp8266-wireless-wifi-shield [Accessed: 16- Dec-2017]
[4]. “Arduino” https://en.wikipedia.org/wiki/Liquid-crystal_display [Accessed: 16- Dec-2017]