Small-Form-Factor Solar-Powered Self-Sustainable IoT Sensor with long-Range Wireless Communication

Design Document

Team Number: sdmay20-07

Client and advisor: Meng Lu and Cheng Hunag

Team Members and Roles:

Calvin Condo - Team Leader

Qin Xia - Sensors

Chuxin Chen - Arduino

Lun Zhang - LoRa module & Arduino

Yuchen Zhao - LoRa module & Arduino

Luke Healy - Arduino & Sensors

Team Email: sdmay20-07@iastate.edu

Team Website: http://sdmay20-07.sd.ece.iastate.edu/

Revised Version 4

Executive Summary

Development Standards & Practices Used

- C code Standards
- Well-Documented & Labelled circuit boards
- Software testing standards
- Clean PCB layout

Summary of Requirements

- Sensors programmed to work with MCU and achieve measurement at low frequency.
- Long-range wireless communication module programmed to work with MCU and can transmit data at least 1 mile away.
- Self-sustainable
- Lower power consumption
- Small design (pocket sized)

Applicable Courses from Iowa State University Curriculum

- EE 201 - Electronic circuits
- EE 230 - Electronic circuits and systems
- EE 330 - Integrated electronics
- EE 321 - Communication system
- EE 333 - Electronic System Design
- CprE 288 - Embedded Systems

New Skills/Knowledge acquired that was not taught in courses

- Long Range Wireless Communication
- Arduino Usage and Programming
- PCB design

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

We would like to thank Dr. Meng Lu and Dr. Cheng Huang for helping us with this project. Their expertise and assistance from the start of the project has helped us understand various aspects of the design. We would also like to thank Lee Harker and ETG for helping us with questions during the time of our research. Thank you.

1.2 Problem and Project Statement

Our project is Small-Form-Factor Solar-Powered Self-Sustainable IoT Sensors with Long-Range (LoRa) wireless communication. The device needs to measure light and temperature in a specified environment. Using the LoRa wireless communication module, we must transmit data to a seperate LoRa module that acts as a gateway to an online source. After that, this data can be monitored by the users from a web browser or a smartphone app. The system is fully self-sustainable using solar energy and power optimization.

The purpose of this project is to create a prototype for our client that can detect light emitted from organisms (bacteria) and measure the temperature from the environment around it. The use of the device will be purely for research purposes.

1.3 Operational Environment

The original idea was for the device to be placed in an open field. In this case, the device would have to be able to operate in outdoor climates, most likely in Iowa. It would have to be waterproof and be able to withstand temperature ranges of 10-90 F. However, since our project will be a prototype it is being built to be used in a lab. Many of the requirements will remain the same, but the environment will be controlled to ensure the prototype functions correctly and so changes can be made as Dr. Lu desires.

1.4 Requirements

- The sensors can collect data correctly.
- The data can be transmitted completely and correctly.
- The device will be pocket sized.
- The entire system needs to be fully self-sustainable.
- The end product needs to work in a wide range of temperatures.

1.5 Intended Users and Uses

Our intended user is our client, Dr. Meng Lu who intends to use it for research purposes on bacteria growth and living conditions.

1.6 Assumptions and Limitations

Assumptions:

- The environment it is placed in will have direct sunlight in order for our device to stay powered using solar cells.
- The device is not expected to be in obscene conditions (weather/hurricanes/tornadoes).

Limitations:

- The final prototype should be as small as possible, at least pocket size.
- Low power consumption to ensure self-sustainability.

1.7 Expected End Product and Deliverables

At the end of semester one, we will produce a preliminary prototype that consists of all the sensors and the LoRa wireless communication module. We will then evaluate the power consumption of our device as well as begin custom PCB designs for the final product.

In the second semester, we will be working on the power optimization of the entire system and the PCB design for our final prototype to be built on. If we have time we will also look into making our own gateway for the receiver device.

At the end, prototyping hardware with all the sensors, LoRa wireless module, and power management circuits in the PCB level. The final product will be pocket size and self-sustainable.

2. Specifications and Analysis

2.1 Proposed Design

The original proposed design was to build the biosensor around an Arduino Uno R3. The figure below shows the original project outline submitted by Dr. Huang. The device would consist of three separate parts: the main board, the sensor board, and the receiver. The main board would consist of the power management, MCU, and communication. The sensor board would be connected to the main board and house an array of light sensors (2x2), the temperature and the humidity sensor. The receiver would be plugged into a gateway that would upload the data to be securely viewed online. Dr. Huang recommended using breakout boards for most of the components in the prototype stage.

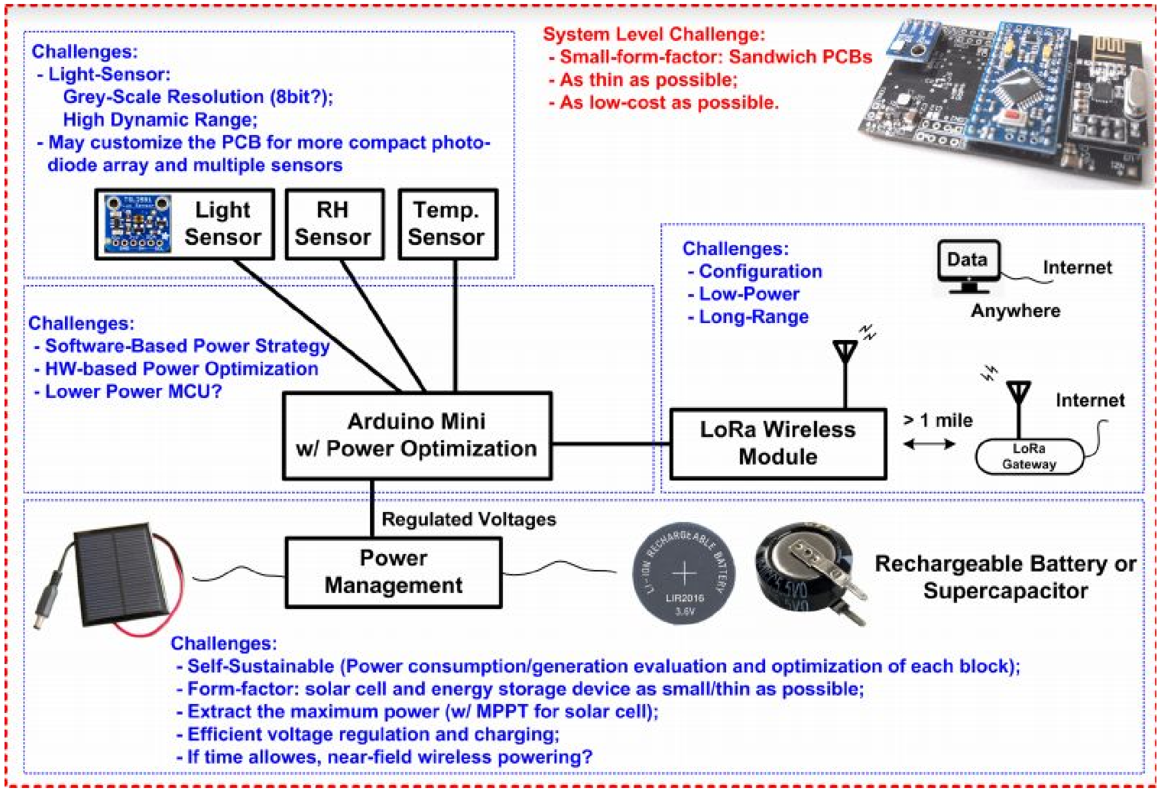

Figure 1: *Original design plan submitted by Dr. Huang. The figure outlines the key elements for the project and the desired limitations.*

Dr. Lu gave us a research paper, "An ingestible bacterial-electronic system to monitor gastrointestinal health" (cited in references), that would help us with understanding and designing our project. It was a project done at MIT to monitor the intestinal health of pigs. The electronic pill was outfitted with photodetectors, a luminescence chip, microcontroller, battery, and antenna. Dr. Lu explained this was what he was looking for a final product. The only main difference is that it would not be used in living creatures.

2.2 Design Analysis

From the proposed design document submitted by Dr. Huang, we created a conceptual sketch of what we envisioned the final product to be (figure 2) and were able to come up with a design plan that followed our advisors' guidance. We separated our tasks into the two semesters. Our goal for the first semester was going to be building the sensors and LoRa module around the MCU and working on the power management, PCB design, and gateway for the second semester. Our plan was to build the project around the Arduino Uno R3, but throughout the project we went through a few changes on the MCU. Since many of the Uno's components would go unused, we decided to downsize and just use the ATmega328p MCU that is on the Uno. Later on we went to an Arduino Mini Pro (3.3V/8MHz), since it could be more easily reprogrammed to fit Dr. Lu's needs.

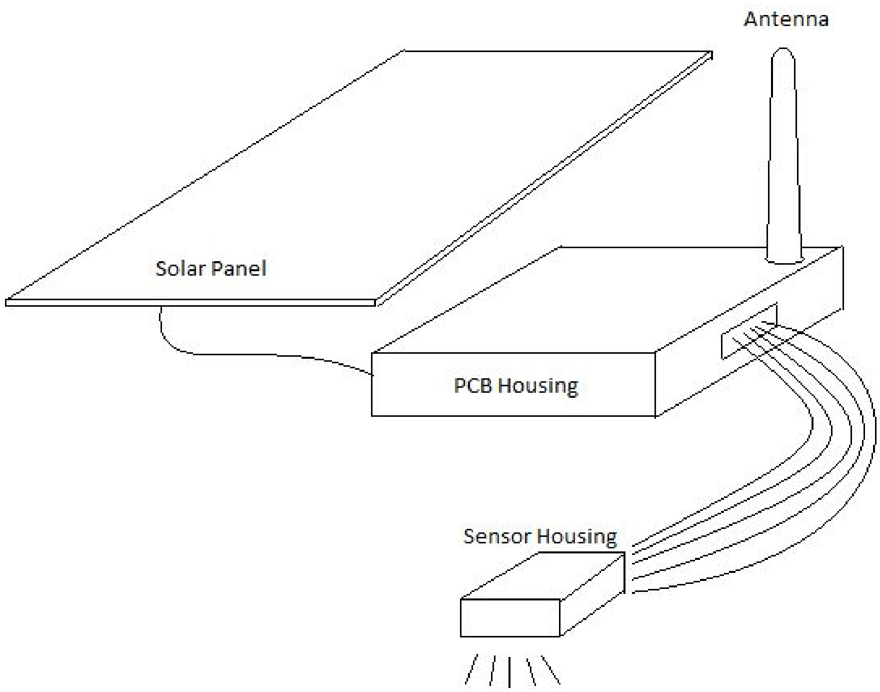

Figure 2: *Conceptual sketch of final product.*

We started by doing research on the sensors and LoRa module. Both had to be compatible with the Arduino and our goals. Our team separated into two groups; one to research the light sensors and the other the LoRa module. After some research, we found a combined humidity and temperature sensor, the DHT22 from Aosong Electronics. However, Towards the start of the second semester, Dr. Lu decided he no longer had a need for the humidity sensor on the device, so we found a new temperature sensor. The TMP36 temperature sensor operates at low voltages with $\pm$ 1 C accuracy.

Researching the light sensor became more challenging than expected. The difficulty was that there is no clear information on the intensity of the light the bacteria will produce. We approached our advisors with our concern and they recommended purchasing two different sensors and testing them in the lab with one of Dr. Lu's graduate students. In the experiment, we placed a showbox on a table with an opening on the top for the light sensor. To generate the light, we used a Pierce ECL Western Blotting Substrate, which consists of Horseradish peroxidase and enhanced chemiluminescence. After combining the two, it would produce a brief moment of light. We would mix the chemicals and quickly place it under the shoebox. The reaction was very quick, so it took a few attempts to get it right.

The first sensor we tested was the TSL2591 high dynamic range digital light sensor from Adafruit (figure 2). It features a dual photodiode system, with one broadband photodiode and one infrared-responding photodiode on a single CMOS IC. The sensor outputs a digital signal to a microprocessor to derive lux from illuminance. The sensitivity ranges from 188 uLux to 88,000 Lux and wavelengths of 400 nm to 1100 nm.

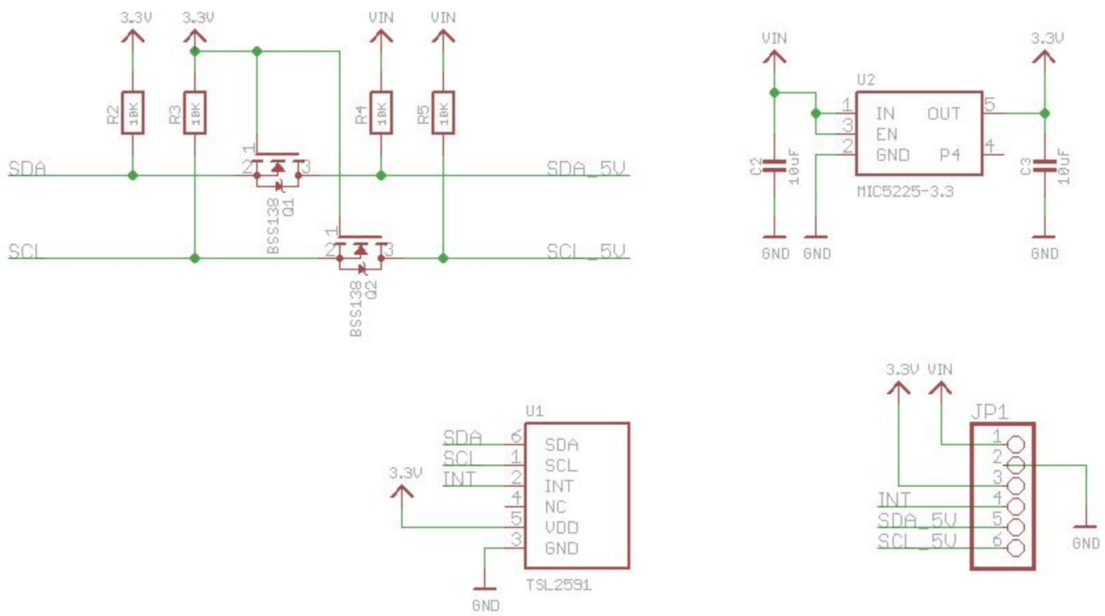

Figure 3: *TSL2591 light sensor schematic from Adafruit Industries.*

The second sensor was the ALS-PT19 analog light sensor from Adafruit (Figure 3). It uses a phototransistor which outputs a voltage based on the photocurrent and load resistor. It is a much smaller and simpler light sensor compared to the TSL2591.

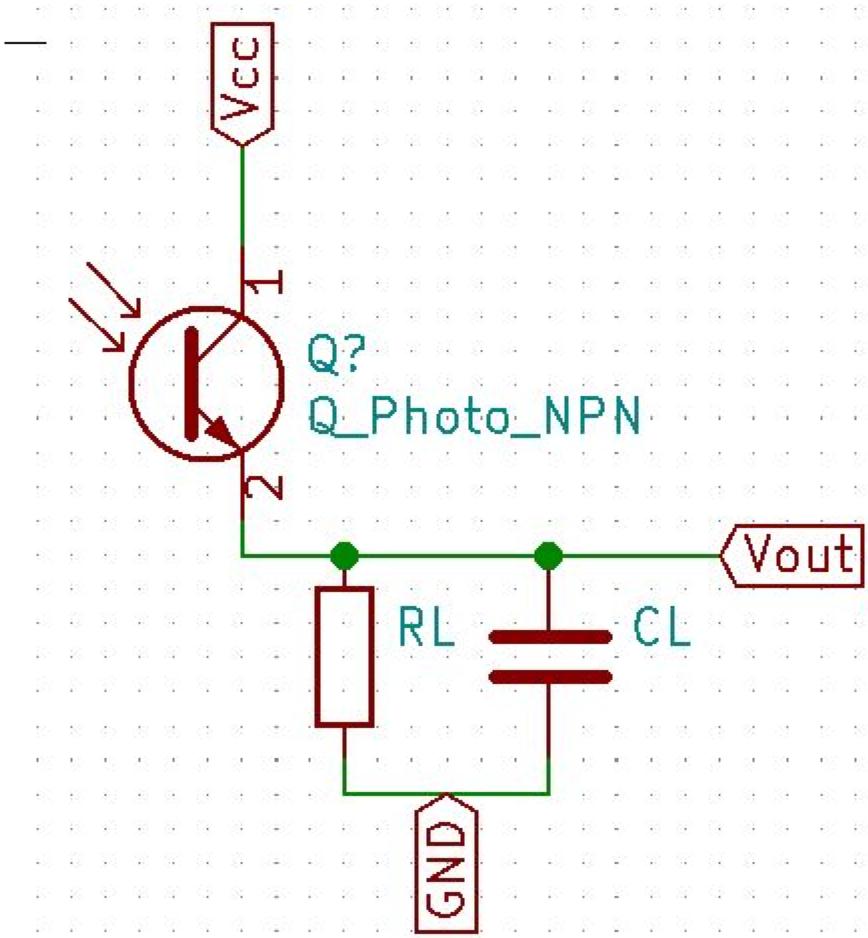

Figure 4: *ALS-PT19 schematic. Uses a NPN phototransistor to produce a photocurrent.*

The ALS-PT19 was not sensitive enough to give readings but the TSL2591 was, so we decided to use it for our device. Since the TSL2591 is a larger and more complex light sensor, we agreed to use just one, instead of an array for this project.

Next we had to figure out a LoRa module. We wanted something that was low cost, low power, and up to a mile range. We landed on the HC-12 wireless serial port communication module. It operates in the 433.4 MHz to 473 MHz frequency range with a maximum transmitting power of 100 mW. Some of the settings had to be manually changed to meet our requirements. This includes, the baud rate set to 2,400 bps to increase the communication range and the mode to automatically adjust the air baud rate according to the serial port baud rate. The idle current of the LoRa module in transmitting mode is 16 mA and maximum current is 100 mA. To limit the amount of current drawn by the LoRa module, we set it to low power mode (idle current 80 uA) while it is not transmitting. We also purchased an alternate antenna (PulseLarsen 430-435MHz stick antenna) to have more reliable data transmission than the default spring antenna that came with the LoRa module.

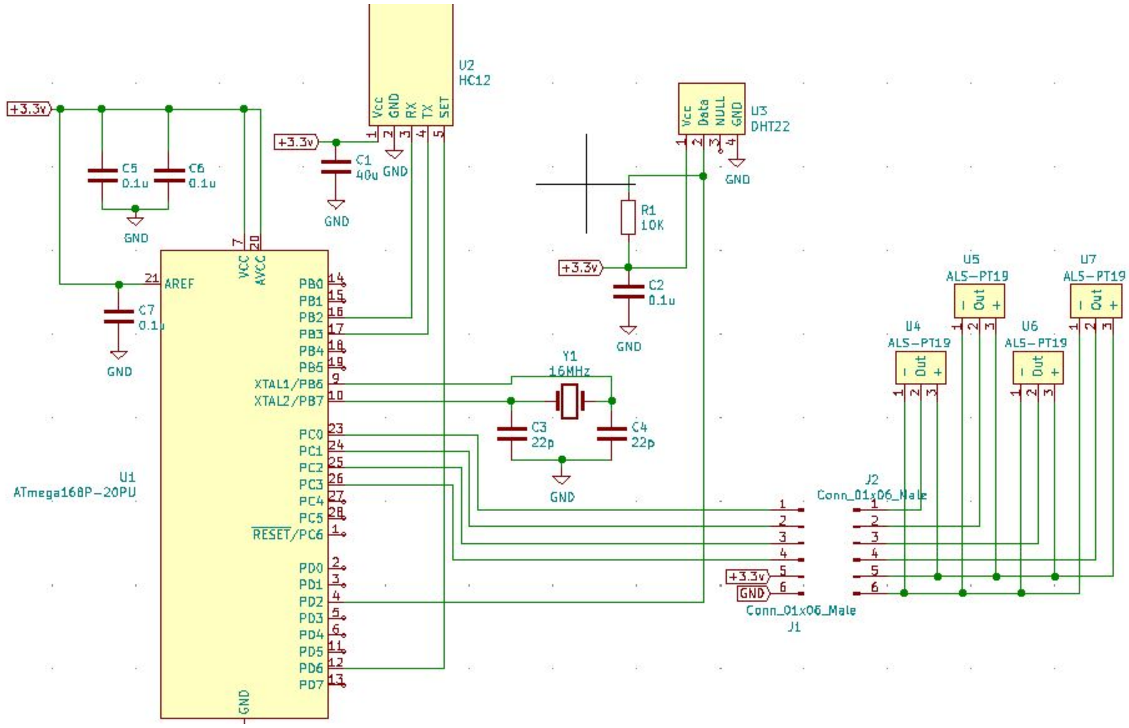

Figure 5: *Full schematic of IoT sensor after first semester. Sensors and LoRa module are connected to a ATmega328p, the microcontroller used on an Arduino Uno. HC12 is the LoRa module, DHT22 is the humidity/temperature sensor, and the ALS-PT19 are the light sensors.*

After we had all our components, we had to figure out how to power it all. We were given that it had to be powered by a battery or super capacitor and they would be charged using solar power. We found a solar panel that fits our specifications in operation and size. The solar panel outputs 6V at 180 mA and dimensions are 89x113 mm. For researching the rest of the power systems, we started by assuming that the longest the device would go without sun was 24 hours. Additionally, the frequency of the device taking measurements would be 6 times an hour (once every 10 minutes). With these assumptions, we could do calculations with different battery options. We decided to use a lithium ion battery and battery charger. The final design consists of two 2032 Li-Ion batteries that will be charged using a battery charger based on the MCP73833 charge management controller. The device is calculated to operate without sun for up to 27 hours.

Next we had to regulate 3.3V to power the MCU and sensors. We used a MCP1700 LDO regulator. We chose this regulator mainly for its low operating current. To improve the range of our LoRa module, we decided to have it operate with 5V so we needed to find another regulator to fit the task. We found the MCP1801 LDO regulator that is the same model as the 1700, which matches our needs.

With all our components figured out, we could create a custom PCB design. We decided that the PCB should match the solar panel in size, so we could easily mount the two together using the mounting holes on the solar panel. Other additions included a power switch and decoupling capacitors. For the sensor PCB, we designed it in KiCAD (figure 6) but realized due to its simplicity it would be easier to fabricate it on a perf board.

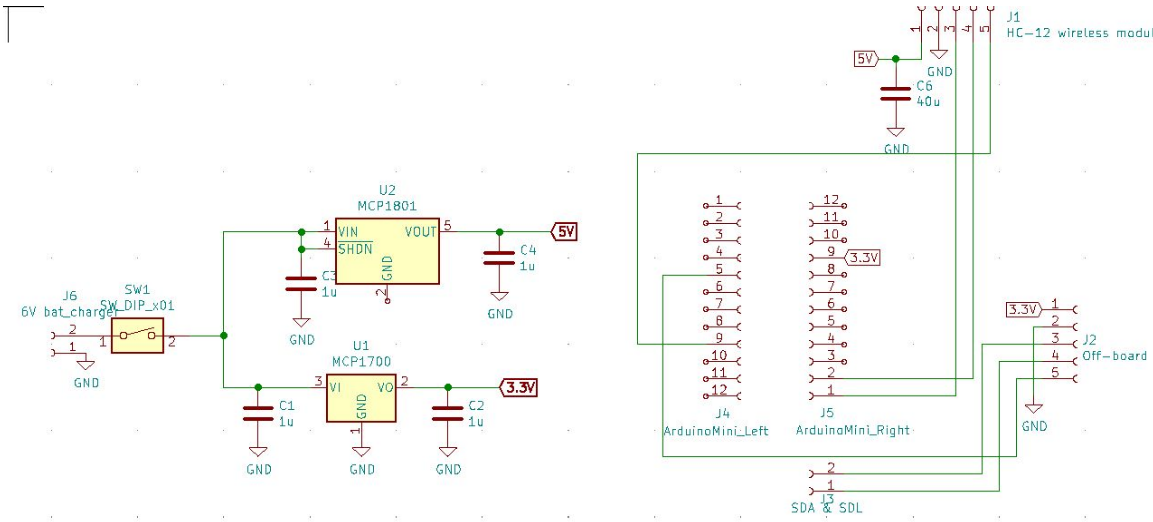

Figure 6: Schematic view of main board.

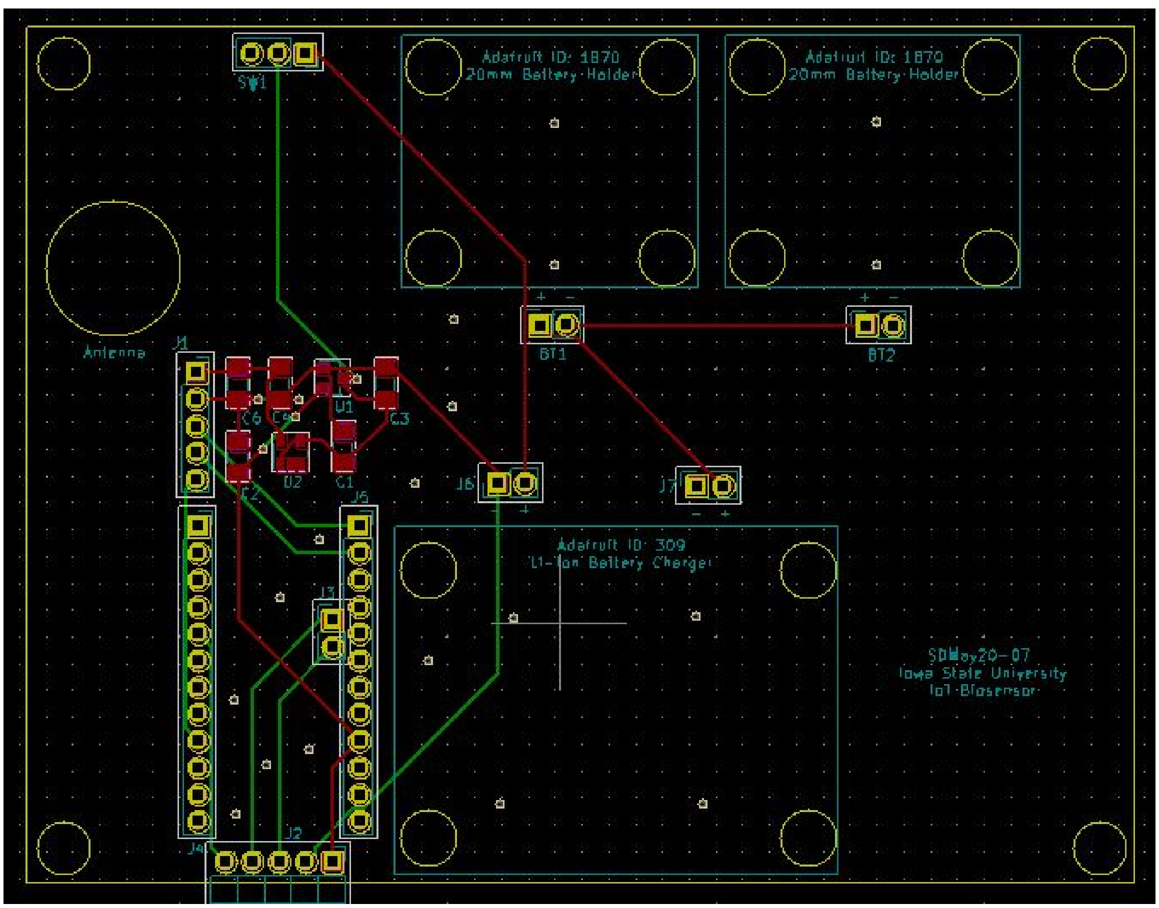

Figure 7: *KiCAD's Pcbnew view of the custom PCB. Bottom left 5 pins (J2) connect to the sensor board, Middle left 5 pins (J1) are for the HC-12 LoRa module, and between them (j3-5) are for the Arduino Mini Pro.*

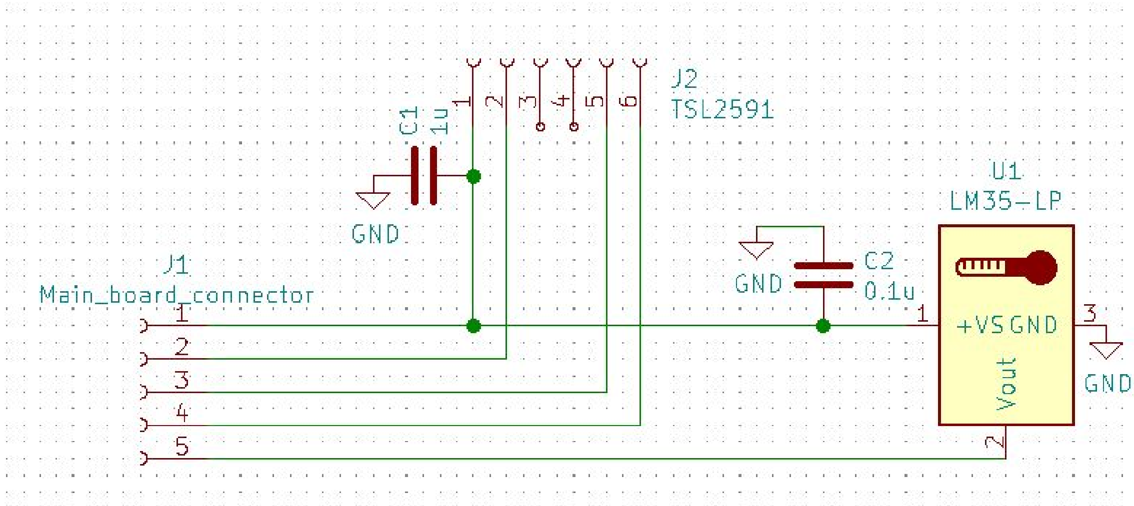

Figure 8: *Schematic view of sensor board.*

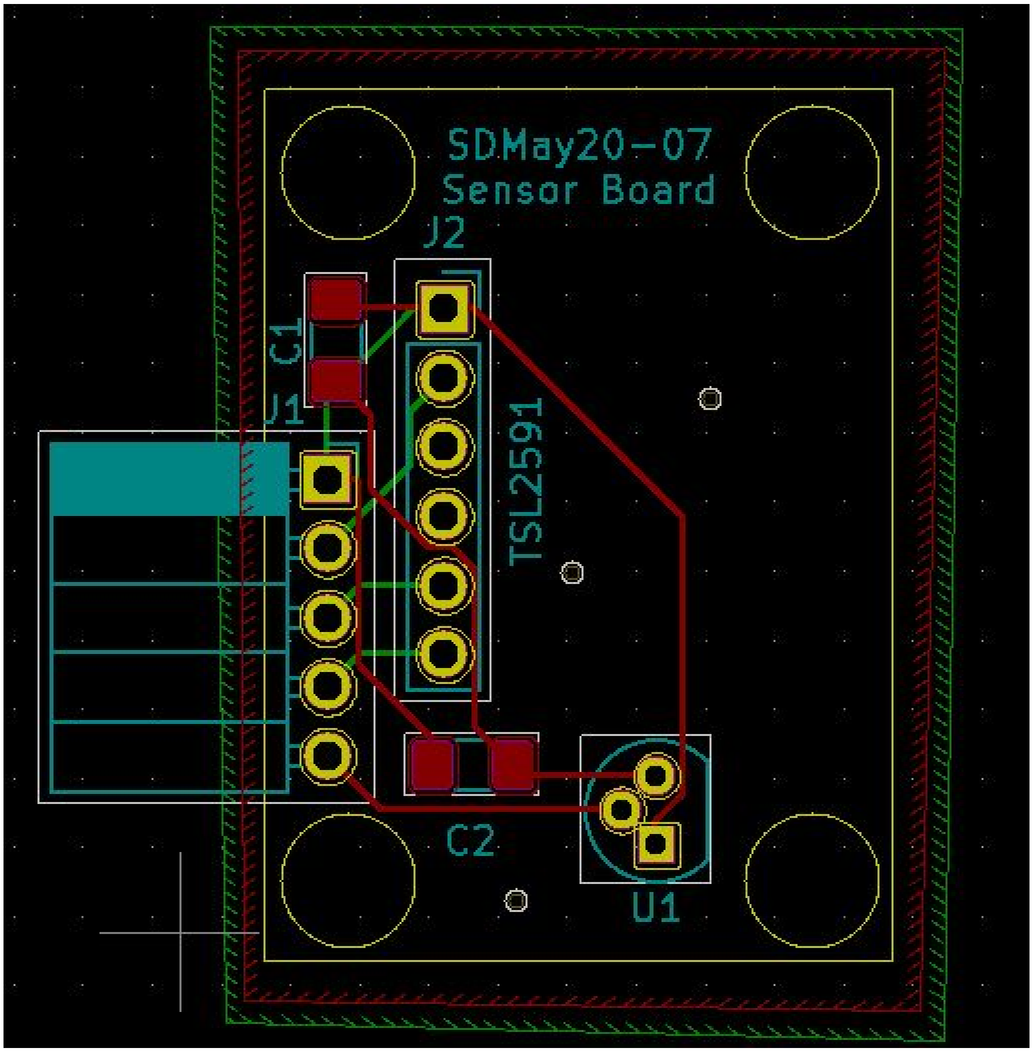

Figure 9: *PCB view of sensor board. Consists of connection pins to main PCB, sensors, and mounting holes for housing.*

2.3 Development Process

Our team's development process will follow the waterfall model, which follows 4 basic steps: research, design, testing, and development. We start every phase of our project with research to understand the area we're going to be working on and familiarize ourselves with the possibilities in that area. For example, there are many different types of light sensors (transistors, diodes, resistors, etc.) so wwe want to find what is best suited for our project. After researching and finding a possible component, we then design a circuit and, if required, the software. We then can move on to testing the design. During testing there can be minor changes to the design in order to optimize the system. If the system does not work as we planned, we move back to the design phase and redesign the system. If there is no way to redesign the component, we move back to the research stage where we select another component we found during our research. If testing goes as planned, we move on to the development. In development, we integrate the components we worked on into the entire system. This is the process we followed to complete our project.

2.4 Design Plan

As mentioned previously, our plan is to build a device around an Arduino based MCU. In our first semester, we will split into two groups to find appropriate sensors. One group will work on finding a light sensor and the other group will find a temperature and humidity sensor. Each group will follow the development process for their respective components. Following the sensors we moved onto finding a LoRa device and then a combination of the LoRa and sensors. In the second semester, we worked on power and creating a custom PCB.

3. Statement of Work

3.1 Previous Work And Literature

There has been no previous work done for his project. Dr. Lu submitted a research paper relating to this project as guidance. The paper is cited in the references under M. Mimee.

3.2 Technology Considerations

For this project, we want to carefully consider the self sustainability and accuracy of the device. Since our device is going to be a prototype, we have opted to buy breakout boards when possible. After our work is done, the next step will be to down size the device from the breakout boards and optimize the device. For now though, we want to create a device that will make the next transition as easy as possible.

To create a self sustainable device, we must consider a battery for the solar energy to charge, power management to optimize the power we are producing, regulators to regulate the correct amount of voltage and current to our components, and the solar panel itself. There are two aspects to accuracy of the device. The first is the accuracy of the LoRa module. The LoRa module will need to transmit data at the correct frequency and transmit and receive complete data. The second is the accuracy of the sensors. This is obtained by researching quality sensors and optimizing the software they operate on.

3.3 Possible Risks And Risk Management

Possible risks for this project include, incomplete transmission of data, the device dying from not enough power, and creating a housing for the device and still get accurate readings. We know the distance our LoRa module is able to transmit data and the bit rate. To ensure we complete transmission, we have installed delays in the software, giving the LoRa module enough time to send all the data.

Another risk we considered was the device losing power. For this, we had to create assumptions in order to approach the problem. We started by assuming the longest the device will go without light would be 24 hours, so our device could rely on battery storage only for at least 24 hours. With this assumption, we know the amount of current our system draws and at what voltage and can then calculate the battery capacity we need.

Accuracy of the sensors is mainly upto the components we buy, making sure they have low error ratios. Once we receive them, we can also modify them in our software. By testing our temperature sensor and having a market thermometer handy, we can see the accuracy of our sensor and change it accordingly. For the light sensor, we use a lux meter to tests its accuracy and adjust it accordingly.

3.4 Project Proposed Milestones and Evaluation Criteria

Our main milestones are as follows:

- Semester 1
    - Complete component research and place order
    - Functional sensors
    - Functional LoRa module
    - Functional LoRa module with sensors
    - Begin PCB design
- Semester 2
    - Complete component research and place order
    - Functional self-sustainability component
    - Update PCB design
    - Finalize and order PCB
    - Build and test full prototype
    - Build final design

For component research, we will have to do extensive research to be sure that components do what we need and are compatible with our other components. To evaluate the functionality of the components is to test they work in our desired way. Once we have confirmed they do, we can start adjusting them to work in the way our project desires. Testing them beforehand also gives a better understanding of the device.

3.5 Project Tracking Procedures

The team set up weekly meetings to work on our project and discuss any changes or problems within the groups. For communication, everyone is a part of a Groupme chat where we can discuss with each other and agree on additional meeting times. Project documents could all be found on a shared google drive folder. Update reports were also completed every other week and posted on our team website.

3.6 Expected Results and Validation

By the end of the first semester, we expect to have the LoRa module and sensors functioning together, as well as a schematic without any of the power aspects. After the second semester, we expect to have a fully self-sustainable functioning prototype.

We expect the sensors to function correctly, meaning they will read accurate data. The sensors are currently set to take measurements every ten minutes. When the measurements are taken, the data should be sent to a gateway to be displayed online. This data should be complete and accurate. The device is expected to remain powered until it requires a change of batteries.

However, due to the COVID-19 pandemic we are unable to create a physical prototype in time. We have completed all the design aspects and ordered all the components, but they will not arrive in time because of mailing services being slowed down. Additionally, all campus labs are closed so we do not have the resources to complete the project.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project TimelinE

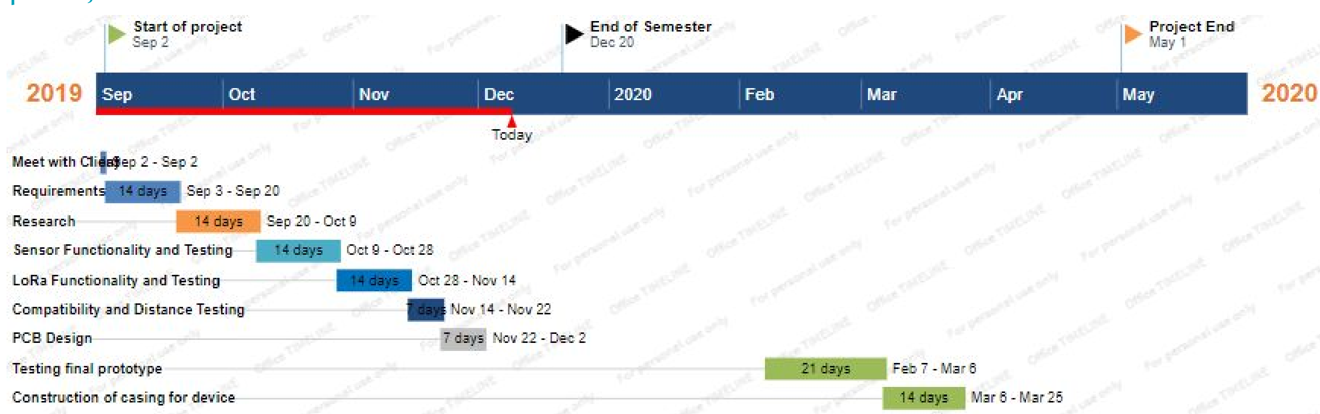

Figure 10: *Gantt chart showing our projects timeline.*

In order to complete our project in 2 semesters, we have to complete a lot of the initial design steps in the first semester and begin constructing the device right away so that it leaves us time to test and make changes in the second semester.

The Gantt chart provided above shows our proposed timeline. Most of the deliverables are in the first Semester by design. We want to get as much done as possible in the first Semester, because we believe there will be a lot of work to do in constructing the final version of our device as we have not taken power consumption into consideration yet. We have found it useful to assume ordering parts from the ETG will take around 2 weeks to deliver.

4.2 Feasibility Assessment

There are no feasibility restrictions for our project. We only have to consider the different conditions we will be testing our device in versus where its application will be. We will be testing in a controlled environment, so certain weather conditions won't be present.

4.3 Personnel Effort Requirements

Table 1: *Teams efforts table in respect to each component of the semester.*

Team Member	Personnel efforts
Calvin Condo	Process development with: light sensor, temperature sensor, LoRa module, and power management. Also worked on the PCB design and

	documentation.
Yuchen Zhao	Process development: light sensor, temperature sensor, LoRa module testing.
Lun Zhang	Light and Temperature sensor testing. LoRa module testing. Final poster.
Qin Xia	Light sensor, temperature sensor, LoRa module testing. Arduino functionality.
Luke Healy	Light and Temperature sensor development and testing. Arduino functionality. Documentation

4.4 Other Resource Requirements

One of our main resources has been Electronics Technology Group (ETG) here at Iowa State. They have not only helped us by supplying components, but also assisting with our understanding of certain design aspects of our project. We also require Iowa State lab equipment (soldering irons, power supplies, etc) for testing and fabricating our project.

4.5 Financial Requirements

This project will require us to find market components that will fit the needs of our project. We are focusing on breakout boards instead of designing our own circuits. We are supplied with the Arduino pro mini but everything else we must order online. The components are expected to total less than $50 and the PCB will be around $20.

5. Testing and Implementation

5.1 Interface Specifications

All of our tests are being done on the Arduino IDE and the code is being mounted directly onto an Arduino so there is no interfacing necessary for this project.

5.2 Hardware and software

We are using the Arduino IDE to code the Arduino. This software is meant for writing C code on Arduinos especially. It includes necessary libraries for most Arduino hardware and makes it simple to import libraries for commonly used hardware modules.

Every circuit and component we construct will be tested for functionality and power consumption. Hardware components include, light sensor (TSL2591), temperature sensor (TMP36), LoRa module (HC-12), and power charger. Hardware circuits include the power regulators (MCP1700 and MCP1801).

5.3 Functional Testing

For functional tests, we had the temperature sensor, the light sensor, and the LoRa module. For the temperature sensor, we tested it by comparing it to a market thermometer. If our sensor did not match the thermometer, we would modify it in our software to output accurate readings. This was done at a range of temperatures.

For the light sensor, we had to create an environment that would mimic that of the bacteria we want to measure. To do this, we had to go to a lab where we could create a reaction to mimic the light produced by bacteria. We set up the test by placing the light sensor in an opening on top of a shoebox. We would then create the reaction and place it inside the dark show box and get the readings. The reaction was created using a Pierce ECL Western Blotting Substrate (figure , which consists of Horseradish peroxidase and enhanced chemiluminescence. After combining the two, it would produce a brief moment of light. The reaction was very quick, so it took a few attempts to get it right.

Figure 11: *Pierce ECL western blotting substrate used to create an imitation of light produced by*

bacteria.

For the LoRa module we did two tests before testing the project code. We did this to become more familiar with the device as none of us had any experience with this type of device. Our first test was to establish basic communication between the two modules by following a tutorial we found online. The second test was to send continuous values from one module to the other, as our project will do. During these two tests, we found that the key to getting good transmissions is incorporating delays within the software to give the module enough time to transmit all the data. Once we successfully completed the two preliminary tests, we wanted to test the range of the module. We set up the transmitter and receiver to send values every 3 seconds. We then went out to a field and kept the receiver stationary, and slowly moved the transmitter further away. We kept track of the distance using a navigation app on our phones.

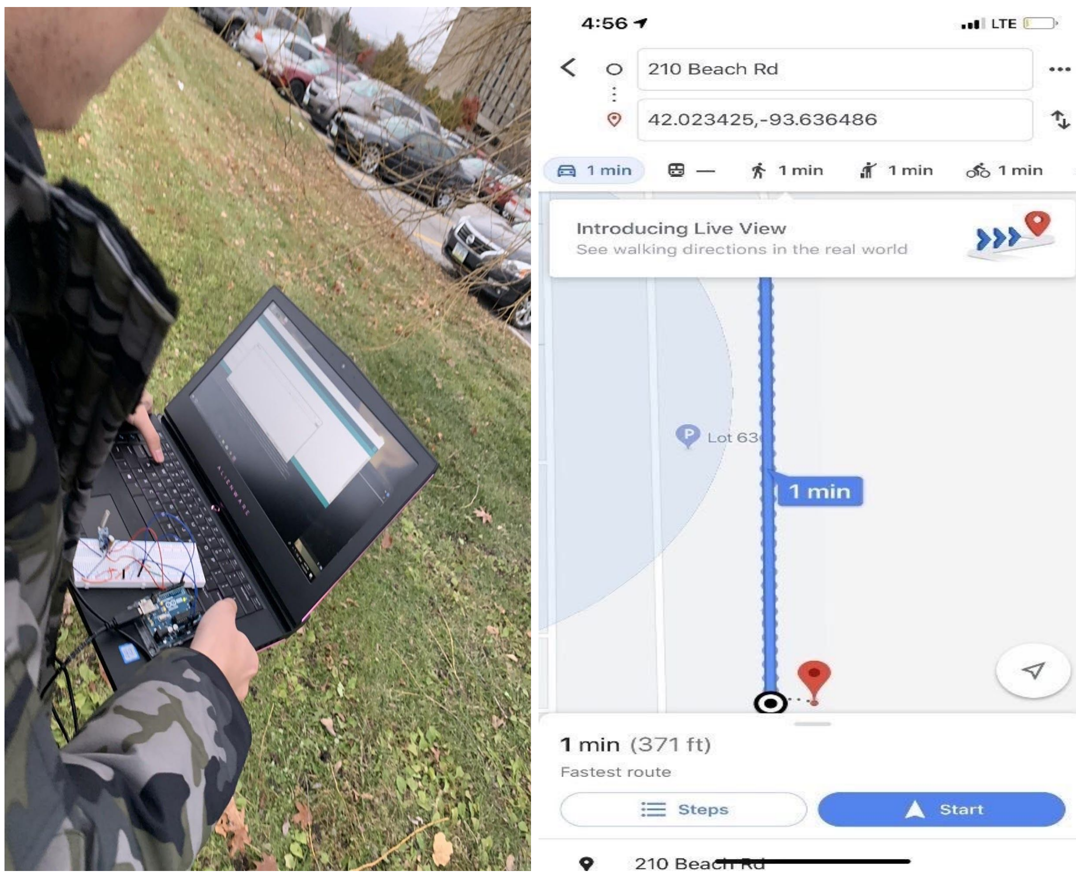

Figure 12: *Testing the LoRa module range in the field by University Towers (left). Example of range tracking using navigation app (right).*

5.4 Non-Functional Testing

Non-functional testing consisted of measuring current drain by each component in order to calculate the necessary battery needed for the device. We calculated for idle and functioning parameters. We would start by referring to the datasheet and calculating using the maximum ratings. From there, we used measured readings to get a new value and have something to compare to in order to ensure accuracy.

5.5 Process

The sensor testing was done by following the instructions from the respective data sheets. The temperature sensor has three connection pins: Vcc, data, and ground. We connected the Vcc and ground to the 3.3 V and ground pin on the Arduino, respectively, and the data pin was connected to one of the digital pins (D2). With the connections made and a few lines of code (all code is included in the appendix) we are able to get temperature readings that resembled that of our market thermometer.

The light sensor test was done in Dr. Lu's biosensor lab in Coover hall. We set up the light sensor on the shoe box and turned the lights off. We had small pill shaped capsules to generate our reaction. We started by placing the peroxide solution in the capsule and then added the catalyst. We then quickly sealed the capsule and placed it under the shoebox.

The LoRa module has five connection pins: Vcc, ground, RXD (receive), TXD (transmit), and set. The set pin is the parameter setting control pin. Our first test was a simple communication test, to understand and see the functionality of the LoRa module. The two modules were set up to send and receive data, so when something was typed, it would be sent to the others serial monitor. The second test was to send analog readings from a potentiometer to adjust the brightness of an LED attached to the receiver. The third test was to test the range limitations of the LoRa module. We set up the transmitter in place, and walked away with the receiver and sent data. If the data was received, we went further and further, etc.

5.6 Results

With the testing process we conducted, we were able to get our expected results. Our light sensor, temperature sensor, and LoRa module all operate accurately. With a little debugging, we were able to get our expected results for all the sensors. The LoRa sensor was tricky at first, but we were able to get it to function correctly. We had a problem where the data was not being fully transmitted, but we solved it by sending the same data 3 times. This way, if there was a gap in the first transmission it can be filled in by the second or third. The receiver code was edited to display the one message.

Due to the COVID-19 pandemic, some of our power tests in the second semester were cut short. We did not, however, let this deter us. We used what we could find using the datasheets and other similar projects we found online. From this, we were able to get mostly accurate values to do calculations and find the remaining components and PCB. We were not able to compile the entire project due to our components not arriving.

Despite not being able to produce a physical functioning prototype, we have completed the full design of Dr. Lu's desired biosensor. We hope that once the parts do arrive we can compile it and see our device function.

6. Closing Material

6.1 Conclusion

Our goal for this project is to create a prototype of a self-sustainable IoT biosensor, with long range communication capabilities. The device will be used by our client, Dr. Meng Lu, for research purposes in bacteria concentrations. In the first semester of the project, our team worked on finding and making functional the light sensor, temperature sensor, and LoRa module. In the second semester, we completed the power and power management of our device making it self-sustainable. By the end of the second semester, we will incorporate both semesters work to have a functioning prototype.

6.2 References

M. Mimee, P. Nadeau, A. Hayward, S. Carim, S. Flanagan, L. Jerger, J. Collins, S. Mcdonnell, R. Swartwout, R. J. Citorik, V. Bulović, R. Langer, G. Traverso, A. P. Chandrakasan, and T. K. Lu, “An ingestible bacterial-electronic system to monitor gastrointestinal health,” *Science*, vol. 360, no. 6391, pp. 915–918, May 2018.

6.3 Appendices

```
/* Senior Design Spring 2020 - team 07
 * Title: Transmitter code
 *
 */
#define DHTTYPE DHT22   //DHT 22  (AM2302), AM2321 - ssensor we are using
#define DHTPIN 2        //Digital pin connected to the DHT sensor
#define ALSPT19a 0      //Analog pin connected to light sensor 1
//#define ALSPT19b 1      //Analog pin connected to light sensor 2
//#define ALSPT19c 2      //Analog pin connected to light sensor 3
//#define ALSPT19d 3      //Analog pin connected to light sensor 4
#define HC12RxdPin 10   //Recieve pin on hc12
#define HC12TxdPin 11   //Transmit pin on hc12
#define HC12SetPin 6    //Set pin on hc12

#include "DHT.h"
#include <SoftwareSerial.h>

SoftwareSerial HC12(HC12TxdPin,HC12RxdPin);   //Create software serial port
DHT dht(DHTPIN, DHTTYPE);

int seth=0;
int setm=0;
int sets=0;
int realh=0;
int realm=0;
int reals=0;

void setup() {
  Serial.begin(9600);
```

Figure 13: *Transmitter code (1/5).*

```
  HC12.begin(9600);
  dht.begin();
  pinMode(HC12SetPin, OUTPUT);
  //digitalWrite(HC12SetPin, LOW);   //hc12 command mode
  //delay(100);
  //HC12.print("AT+B2400");          //set baud rate to 2400bps
  //delay(200);
  digitalWrite(HC12SetPin, HIGH);   //hc12 normal mode

  //Set time increment between each measurment
  seth = 0;    //set hours
  setm = 0;    //set minutes
  sets = 3;    //set seconds
}

void loop() {

  //timer
  while((seth != realh) || (setm != realm) || (sets != reals)){
    //cycle clock
    reals = reals + 1;
    delay(1000);
    //handle overflow conditions for seconds, minutes, and hours
    if(reals > 59){
      reals = 0;
      realm = realm + 1;
    }
    if(realm > 59){
      realm = 0;
```

Figure 14: *Transmitter code (2/5).*

```
      realm = 0;
      realh = realh + 1;
    }
    if(realh > 23){
      realh = 0;
    }
  }
  //reset timer
  realh=0;
  realm=0;
  reals=0;

  // Reading temperature or humidity takes about 250 milliseconds!
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  //Read light sensors
  int Lsensor1 = analogRead(ALSPT19a);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
```

Figure 15: *Transmitter code (3/5).*

```
// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("%  Temperature: "));
Serial.print(t);
Serial.print(F("°C "));
Serial.print(f);
Serial.print(F("°F  Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.print(F("°F"));
Serial.print(F("Light"));
Serial.print(Lsensor1);

//Convert all float values to char in order to be transmitted
//HUMIDITY CONVERSION
float HumidityFloatPost;
char HumidityCharPre[3];
char HumidityCharPost[3];
//convert value before and after decimal place to char
utoa(h, HumidityCharPre, 10);
HumidityFloatPost = h - (int)h;
HumidityFloatPost = HumidityFloatPost*100;
utoa(HumidityFloatPost, HumidityCharPost, 10);
```

Figure 16: *Transmitter code (4/5).*

```
float TemperatureFloatPost;
char TemperatureCharPre[3];
char TemperatureCharPost[3];
//convert value before and after decimal place to char
utoa(f, TemperatureCharPre, 10);
TemperatureFloatPost = f - (int)f;
TemperatureFloatPost = TemperatureFloatPost*100;
utoa(TemperatureFloatPost, TemperatureCharPost, 10);

//LIGHT CONVERSION
char light1[4];
utoa(Lsensor1, light1, 10);

//send data
HC12.write("RH:");
HC12.write(HumidityCharPre);
delay(100);
HC12.write(".");
HC12.write(HumidityCharPost);
delay(100);
HC12.write("% Temp:");
HC12.write(TemperatureCharPre);
delay(100);
HC12.write(".");
HC12.write(TemperatureCharPost);
HC12.write("F ");
delay(100);
HC12.write("Light:");
HC12.write(light1);
```

Figure 17: *Transmitter code (5/5).*

```
#include <SoftwareSerial.h>

#define HC12RxdPin 10   //Recieve pin on hc12
#define HC12TxdPin 11   //Transmit pin on hc12
#define HC12SetPin 6    //Set pin on hc12

SoftwareSerial HC12(HC12TxdPin,HC12RxdPin); //Create software serial port

void setup() {

  Serial.begin(9600);           //open serial port to computer
  HC12.begin(9600);             //open serial port to hc12
  pinMode(HC12SetPin, OUTPUT);
  digitalWrite(HC12SetPin, LOW);   //hc12 command mode
  delay(100);
  HC12.print("AT+B2400");          //set baud rate to 2400bps
  delay(200);
  digitalWrite(HC12SetPin, HIGH);   //hc12 normal mode
}

void loop() {

  if(HC12.available()){         //if arduino's hc12 rx has data
    Serial.write(HC12.read());  //send the data to the computer
  }
  if(Serial.available()){       //if arduino's computer rx buffer has data
    HC12.write(Serial.read());   //send that data to serial
  }
}
```

Figure 18: *Receiver code.*