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 3

Executive Summary

Development Standards & Practices Used

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

Summary of Requirements

- Sensors programed to work with MCU and achieve measurement at low frequency.
- Long-range wireless communication module programed to work with MCU and can transmit data at least 1 mile away.
- Self-sustainable
- Lower power consumption
- Powered by solar cells
- 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

1 Introduction 4

1.1 Acknowledgement 4

1.2 Problem and Project Statement 4

1.3 Operational Environment 4

1.4 Requirements 4

1.5 Intended Users and Uses 4

1.6 Assumptions and Limitations 5

1.7 Expected End Product and Deliverables 5

2. Specifications and Analysis 5

2.1 Proposed Design 5

2.2 Design Analysis 6

2.3 Development Process 6

2.4 Design Plan 6

3. Statement of Work 6

3.1 Previous Work And Literature 6

3.2 Technology Considerations 7

3.3 Task Decomposition 7

3.4 Possible Risks And Risk Management 7

3.5 Project Proposed Milestones and Evaluation Criteria 7

3.6 Project Tracking Procedures 7

3.7 Expected Results and Validation 7

4. Project Timeline, Estimated Resources, and Challenges 8

4.1 Project Timeline 8

4.2 Feasibility Assessment 8

4.3 Personnel Effort Requirements 8

4.4 Other Resource Requirements 8
4.5 Financial Requirements 9
5. Testing and Implementation 9
5.1 Interface Specifications 9
5.2 Hardware and software 9
5.3 Functional Testing 9
5.4 Non-Functional Testing 9
5.5 Process 10
5.6 Results 10
6. Closing Material 10
6.1 Conclusion 10
6.2 References 10
6.3 Appendices 10

List of figures/tables/symbols/definitions

Figure 1: *Design plan* 7

Figure 2: *Light sensor schematic* 8

Figure 3: *Light sensor schematic* 9

Figure 4: *Complete schematic* 10

Figure 5: *Project flow diagram* 11

Figure 6: *Project timeline* 13

Figure 7: *BOM* 17

Figure 8: *Transmitter code(1/5)* 17

Figure 9: *Transmitter code(2/5)* 18

Figure 10: *Transmitter code(3/5)* 19

Figure 11: *Transmitter code(4/5)* 20

Figure 12: *Transmitter code(5/5)* 21

Figure 13: *Receiver code* 22

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

1.3Operational 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 in Iowa. It would have to be waterproof and be able to withstand temperature ranges of 10-90 F. However, later on in the project it was thought more efficient to have the prototype be tested in a controlled environment.

1.4Requirements

- 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 be waterproof.
- The end product needs to work in a wide range of temperatures.

1.5Intended 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 in soils.

1.6Assumptions 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 is comprised 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, self-sustainable, and work under specific weather conditions.

2. Specifications and Analysis

2.1 Proposed Design

The original proposed design was to build the device around an Arduino Uno R3. However, this plan will make many of our limitations difficult to meet as the Uno alone is pocket sized and many of the Uno components will go unused. We decided to instead use only the microcontroller from the Uno, the ATmega328p. The figure below shows the original project outline submitted by Dr. Huang. Dr. Huang also recommended using breakout boards for the sensors in the prototype stage.

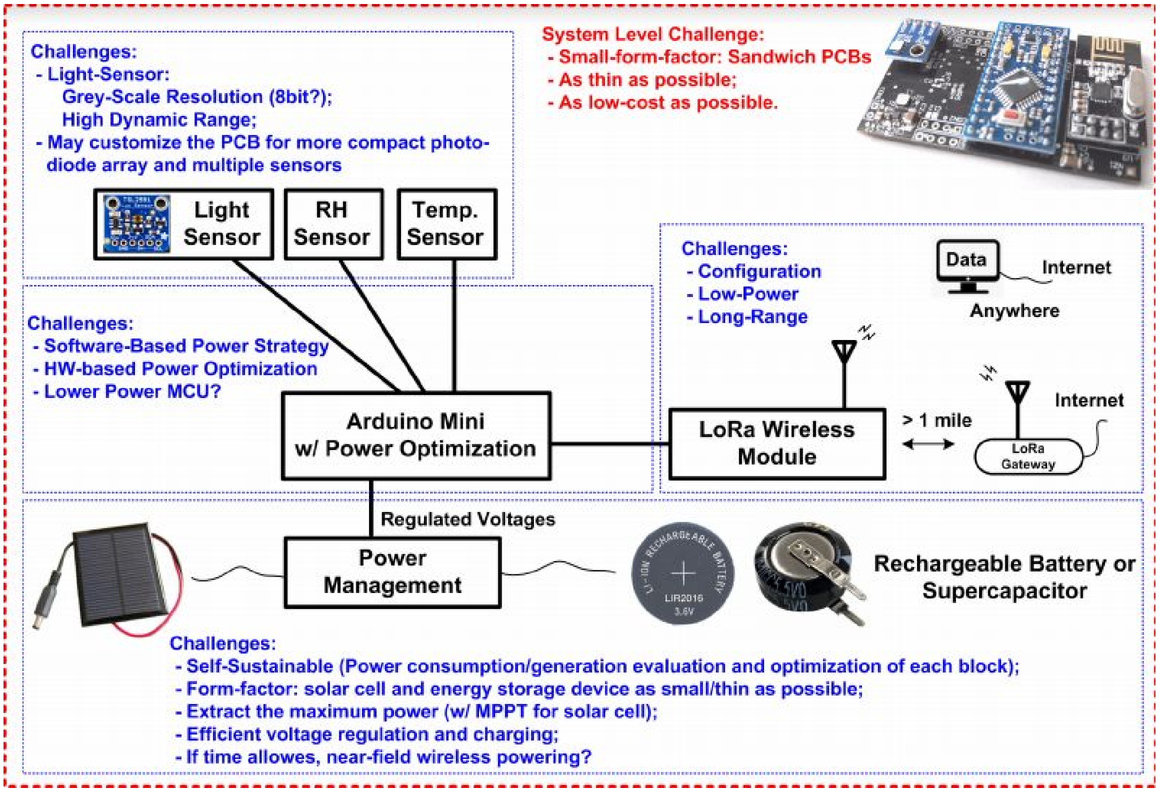

Figure 1: *Original design plan submitted by Dr. Huang. The figure outlines the key elements for the project and the desired limitations.*

Dr. Lu also gave us a research paper that would help us with understanding and designing our project. The title of the paper is "An ingestible bacterial-electronic system to monitor gastrointestinal health" (cited in references). 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, but soils.

2.2 Design Analysis

From the proposed design document submitted by Dr. Huang, we were able to come up with a design plan that followed our advisors guidance. Our goal for the first semester was going to be building the sensors and LoRa module around the Arduino and save the power management, PCB design, and gateway for the second semester.

Our first step was to research the sensors and LoRa module. Both had to be compatible with the Arduino and low cost. 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. The sensor can sense 0-100% RH and -40-80 Celsius. It operates by being sent a start signal and will respond with a 40-bit data signal that reflects the relative humidity and temperature from its internal MCU. The sensing element is a polymer capacitor.

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 and research and they recommended purchasing two different sensors and testing them in the lab with one of Dr. Lu's graduate students. The first sensor 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.

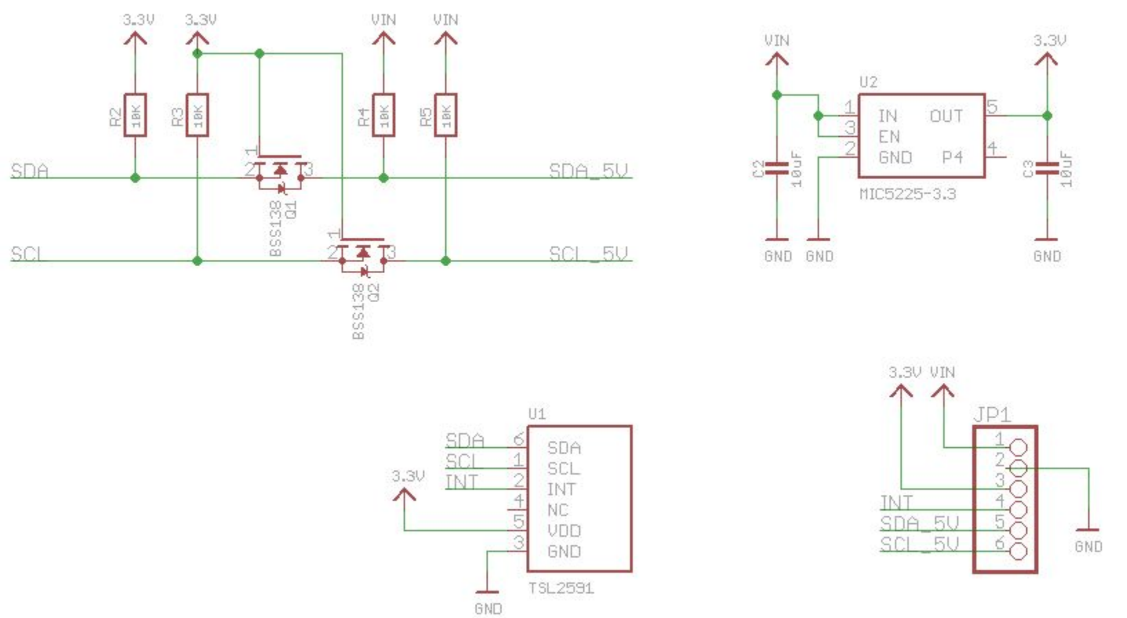

Figure 2: *TSL2591 light sensor schematic from Adafruit Industries.*

The second sensor is 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 to the TSL2591. Our first choice is the ALS-PT19, because it is similar to what was used in the MIT project. If we can not get readings from the ALS-PT19, we will use the TSL2591.

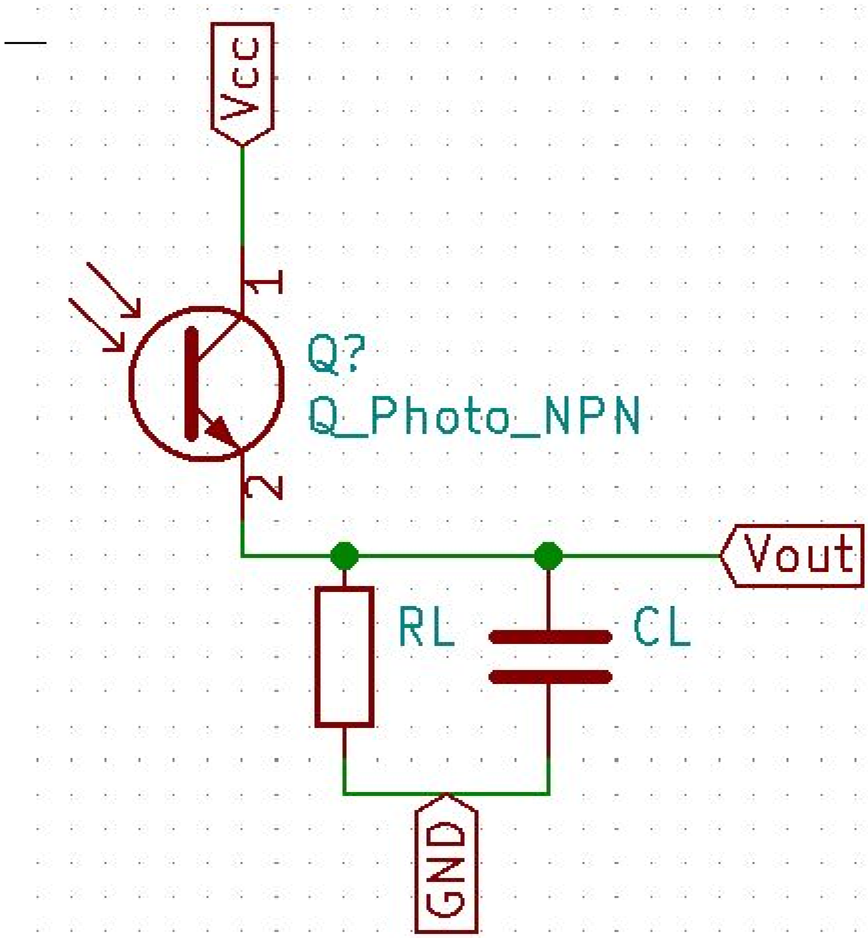

Figure 3: *ALS-PT19 schematic. Uses a NPN phototransistor to produce a photocurrent.*

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 is 16 mA and maximum operating current is 100 mA. 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.

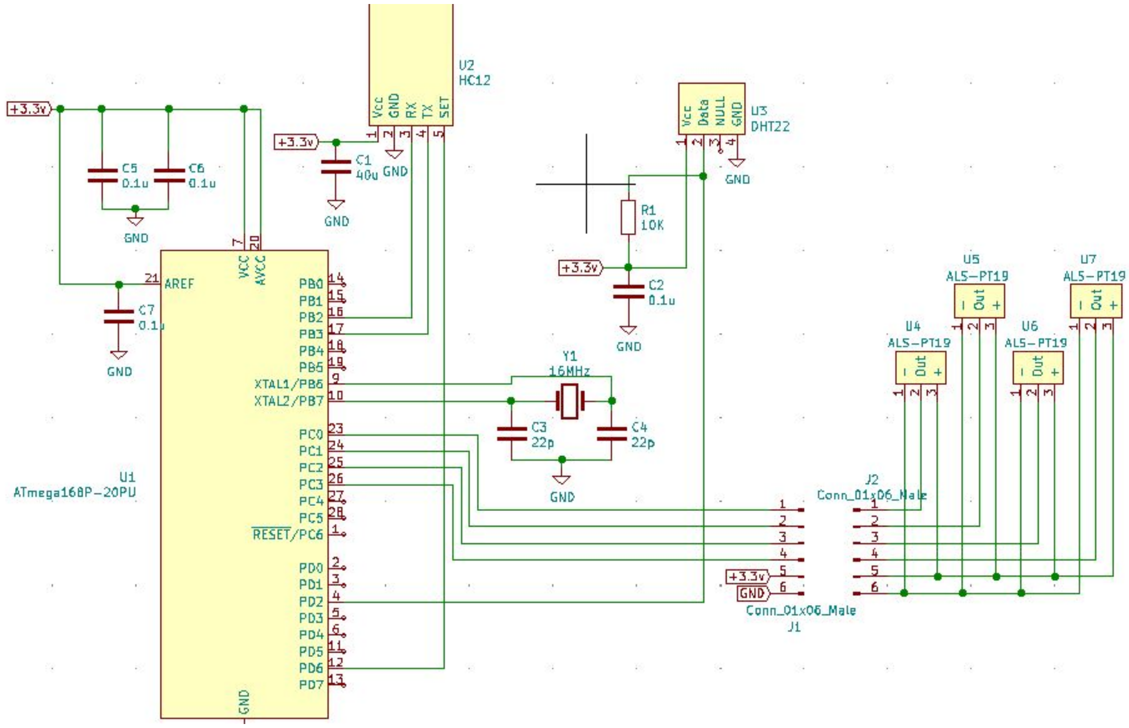

Figure 4: *Full schematic of IoT sensor. Sensors and LoRa module are connected to a ATmega328p, the microcontroller used on an Arduino Uno. HC12 is the LoRa module, DHT22 is the humidity/temperature sensor, and the ALS-PT19 are the light sensors.*

2.3 Development Process

Our team is following the waterfall model. We have the requirements from our client and have designed what we think would be the best approach in solving the problem. We split into teams in order to divvy up the workload and will combine the work that each team has completed at the end.

2.4 Design Plan

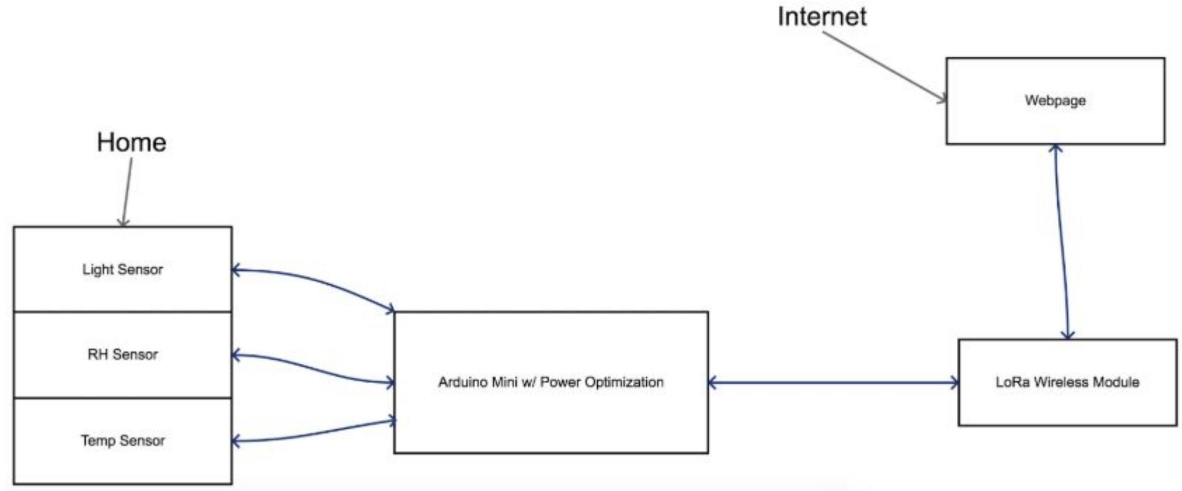

Figure 5: Semester 1 rough design plan.

3. Statement of Work

3.1 Previous Work And Literature

There has been no previous work done for his project so we will have to start from scratch. We have one research paper (Mimee, listed in references section) we will be using as a model for our project.

3.2 Technology Considerations

The three main characteristics of this project are low cost, long range, and self sustainable. We want to buy market components that can be used in the prototype stage of this project. The idea is that once it is built it can be taken to the next step and down scaled in size and price. To start, however we purchase sensors and a LoRa module that require minimal setup to function. Additionally, they must work with low power consumption in order to maintain self-sustainability.

By the end of the project, we will have a self-sustainable system with a schematic of all the components used in it. From there, our client can look into the next step of the IoT sensor.

3.3 Possible Risks And Risk Management

A risk that we have considered is the reliability of transmitting data. Our LoRa module specifications are at the limit of our desired range, so there could be times when data is not received due to being out of range. We have planned to combat this by investing in a different antenna than what came with the LoRa module. This will hopefully amplify the range enough to where it will work consistently.

Another risk is the light sensor ability to sense low levels of light. The light sensor will be sensing light generated from bacteria in soil. Since we are unable to test our sensors in this environment, we are left to replicate the environment with what we know.

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
  - 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 test 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

We have laid out scheduling plan where we have a specific day and time we meet weekly. Additional times are set up if needed. We have also set up a time table on google drive to keep track of members work hours. Using these methods we plan to clear communication and productive work times.

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 should be able to stay in an open field as long as possible. It is not yet clear how long this should be, but as long as there is a source of light for our solar cells to harvest power, it should stay functional.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project TimelinE

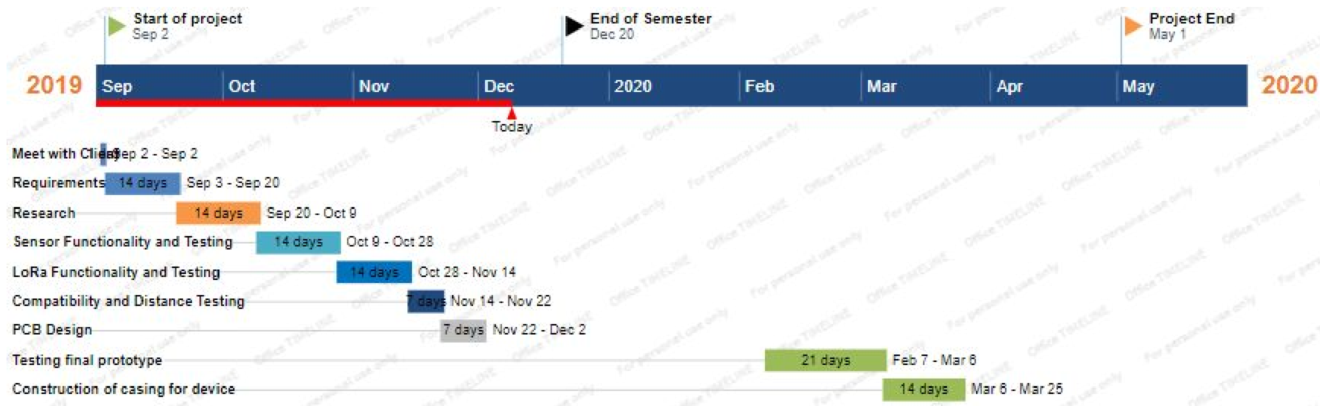

Figure 6: *Gantt chart showing our projects timeline.*

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. Also, measuring the illumination of bacteria will be difficult for we do not have the means currently to test it regularly. We have setup a time with Dr. Lu and his graduate student to test it and will have to be set on changes we make at this time.

4.3 Personnel Effort Requirements

Table 1: *Rough teams efforts table in respects to each component of the semester.*

Task	Personnel effort	Time
Parts research	We have to do a lot of research to select the right parts for our project. Our parts need to be low power consumption to meet our requirements.	Calvin: 10 h Chuxin:6 h Qin: 6 h

		Lun: 6 h Yuchen: 8h Luke:8 h
Sensor testing	We check the data sheet to make sure how to connect with the breadboard, and then,set up the Arduino and write the code to function every sensor. We need to make sure all the data read from Arduino are correct.	Calvin:8 h Chuxin:6 h Qin:8 h Lun:8 h Yuchen:8 h Luke:6 h
Lora module testing	Lora module should be connected correctly and can transmit and receive the message properly. We first tried to transmit some simple words, after that, we tried to transmit the sensor data to the receiver port.	Calvin:10 h Chuxin:7 h Qin:6 h Lun: 8 h Yuchen:5 h Luke:5 h
Arduino coding	We have to write a clear and concise Arduino code of all the functions. All the code should be have proper functionality.	Calvin:8 h Chuxin: 10 h Qin:6 h Lun:4 h Yuchen:6 h Luke:5 h

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. For example, none of our team members have designed a PCB before, so ETG has supplied us with different resources to understand and design our own PCB.

4.5 Financial Requirements

The total will be less than $70, this will include all the sensors, communication, MCU, and customized PCB board. In the next semester, we also need to buy a solar energy panel and a case to protect our final prototype.

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 not much interfacing necessary.

5.2 Hardware and software

We use the Arduino IDE to help code on the Arduino Uno. 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.

Hardware we tested were all of the sensors and the LoRa module. We did this by finding online examples and by doing them learned how to wire and code them. We built our test circuits using a breadboard. Each test was similar in some way to what we want to do for the project and we will expand on them in section 5.5.

5.3 Functional Testing

For functional tests, we had the humidity-temperature sensor, the LoRa module, and the two light sensors. The sensors were the easiest to test since they only have one mode and function. 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 communication between the two modules. The second test was to send a continuous value from one module to the other. After testing its functionality we also had to test the limitations on the LoRa module, specifically its range.

5.4 Non-Functional Testing

The only non-functional testing was compiling the code. There was a point when we were waiting for our parts to arrive in the mail, so at that time we wrote the code for the test cases and the project. Once the code was written and compiled we had to wait for functional testing.

5.5 Process

The sensor testing was done by following the instructions from the respective data sheets. The humidity-temperature sensor has four connection pins: Vcc, data, ground, and unconnected. We connect 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). The datasheet also instructs to place a 10 KOhm resistor between data and Vcc to act as a pull up on the data line. With the connections made and a few lines of code (all code is included in the appendix) we are able to get humidity, temperature, and heat index readings. The same approach was taken for the light sensors.

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 setup 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 a little debugging, we were able to get our expected results. We set the sensors to read temperature and humidity every three seconds. We tested it inside and out and were able to see accurate readings when compared to weather data and thermostats. The light sensor is still being tested for accuracy, but we are able to see a change in readings when we bring the light source closer to the sensor.

The LoRa sensor was tricky at first but we were able to get it to function correctly. Sometimes there are skips in data transmission, but we are looking into a way to solve it. Our current idea is to send the same data multiple times to ensure it is received.

Once both components functioned, we put them to work together. This was simple since the potentiometer test was the same except with a few more inputs. At first we were not getting all the readings to the receiver. The reason was that there was not enough time for the LoRa to send all the data, and we fixed it by creating delays between data chunks.

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 soils. In the first semester of the project, our team will work on finding and making functional the light sensor, humidity sensor, temperature sensor, and LoRa module. By the end of the second semester, we will incorporate the LoRa gateway and self-sustainable portion of our project 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

Component	Specification	Quantity	Cost (ea)
Microcontroller	Arduino Uno R3 (Atmega328p)	2	$15.98
LoRa module	HC-12 433 SI4463 Wireless serial module	2	$12.90
Light sensor	Adafruit ALS-PT19 Analog Light Sensor	4	$2.50
RH + Temp. sensor	DHT22 temperature-humidity sensor	1	$9.95
Antenna	Pulse Larsen Antenna 430-435 MHz	2	$5.90
Adapter	U.FL (UMCC) to SMA	2	$3.95
			$51.18

Figure 7: *Initial design bill of materials.*

```
/* 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 8: *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 9: *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 10: *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 11: *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 12: *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 13: *Receiver code.*