Energy Harvesting Hydrometer

Design Document

Team Number:

SDDEC20-06

Client and Advisor:

Prof. Henry Duwe

Team Members:

Griffin Orr – Hardware Design

Tilden Chen – Embedded Systems Design

Chris McGrory – Base Station Design

Chris Pedersen – Embedded Systems Design

Jensen Mayes - Mechanical Design

Josh Hall – Base Station Design

sddec20-06@iastate.edu

https://sddec20-06.sd.ece.iastate.edu

Executive Summary

Development Standards & Practices Used

- Kicad for schematic layout and PCB design
- Advanced Design Systems for RF design
- FreeCAD for mechanical design

Summary of Requirements

- Wirelessly powered
- Inert
- Wirelessly transmit readings with a minimum frequency of hourly
- Measure the specific gravity of the fluid
- Measure the temperature of the fluid
- Waterproof

Applicable Courses from Iowa State University Curriculum

- EE285
- EE303
- EE452
- EE333
- EE201
- EE230
- IE305
- EE311
- PHYS221
- CPRE 281
- CPRE288

New Skills/Knowledge acquired that was not taught in courses

- Wireless power transfer using matched antennas
- Bluetooth communications on embedded systems
- Tilt hydrometer design and implementation
- Electrical integration into mechanical systems

Table of Contents

1 Introduction

1.1 Acknowledgement

We would like to acknowledge Assistant Professor Henry Duwe for his contributions towards our technical design. Professor Duwe works at Iowa State University and focuses on computing and networking systems for secure and reliable computing. One of his current research areas concentrates on wireless energy harvesting and limitations associated with onboard functionality. His research in this area has provided us with the groundwork information that is fundamental in the implementation of our design. He has also overseen all the individual research that our team has compiled throughout this project.

1.2 Problem and Project Statement

In the past decade researchers have been exploring the many facets of wireless energy harvesting. In this process an RF signal is used to wirelessly transmit power to a designated electronic platform. This platform uses either a re-chargeable battery or capacitor to store and distribute the signal energy. Many embedded computational systems are beginning to take advantage of this developing technology but there are many issues surrounding its implementation. Such issues include limited power transfer, signal reception quality, signal strength, etc.

Our project expands the capabilities of these embedded computational systems and reduces some of the limitations associated with wireless power transfer. To achieve this goal, we have developed a device that can measure the specific gravity within an enclosed container with one hundredth precision. The device is powered wirelessly by a commercially produced PowerCast transmitter (tx91501b) at 915Mhz. It remains unpowered until the onboard capacitor is charged to a voltage release point. The stored power is distributed using an appropriate hysteresis model to power onboard instrumentation. The device also wirelessly transmits the sampled data to a collection device.

The commercial intent of this system is to supply home brewing hobbyists/enthusiasts with a better way to monitor the quality and control of their fermentation process. There are various hydrometer models on the market right now, but their main disadvantage is that they are all battery powered. This means that the user would have to expose the inside of the fermentation tank to change/charge the battery. Our system would allow for the fermentation process to be completely enclosed and free from outside bacteria.

1.3 Operational Environment

The intended operational environment for our design is a liquid filled enclosure. This environment is inclusive of high temperatures and liquid exposure. Other suitable environments for our platform include aquariums, ponds, or any other enclosed liquid environment.

1.4 Requirements

The following requirements have been assigned to this project:

1. The transmission signal must charge the hydrometer within a days' time without losing signal quality.
2. The transmission signal must provide enough power for consistent data collection and transmission.
3. The data signal must be able to transmit when submerged in liquid.
4. The design must be waterproof.

1.4.1 Engineering Constraints and Non-functional Requirements

Engineering Constraints:

- The device will transmit at least once per activation
- The transmission will utilize Bluetooth iBeacon
- Code will be optimized to increase boot speed and decrease power consumption

Non-functional Requirements:

- PCB dimensions of 3 inches by 1.5 inches
- The device will transmit at minimum once per hour

1.5 Intended Users and Uses

Our design is just one of the many applications that this technology could be used for. With our design one could easily modify the base level instrumentation to fit their specific application. Other applications could include, oil & gas monitoring, wastewater treatment, chemical plants, etc. These applications are all inclusive of physically restrictive environments that require regular instrumentation monitoring. Overall, our design looks to improve upon the typical limitations associated with wireless energy transfer and inspire future wireless-power applications.

1.6 Assumptions and Limitations

Assumptions:

- Most calculations shall be done on the base station to maximize expandability.
- The hydrometer will operate within a temperature range of 50°F to 80°F.
- The specific gravity measurement should be within 1% margin of error.
- The hydrometer will only need to transmit data once per hour.
- The less space the antenna occupies, the better.
- The hydrometer will be in an environment where the 915 MHz band is unused.

Limitations:

- The system will not use a battery.
- The hydrometer will not be transmitting from a metal container.
- The hydrometer will not be farther than 2 ft from the PowerCast.

1.7 End-Product and Deliverables

The deliverable is a prototype, fully encapsulated, hydrometer that can be powered by a functional energy harvesting sub-system. Included with this is the code for our desired base station to communicate with the hydrometer and the code on the hydrometer that transmits the data.

2. Specifications and Analysis

2.1 Proposed Approach

A hydrometer is an instrument used for measuring the relative density of liquids based on the concept of buoyancy. Our design is an electronic implementation of this instrument that is charged wirelessly by an external power source. As shown in figure 1, the design is broken up into two main components, the hydrometer module itself and the base station module. The hydrometer module is composed of a power harvesting receiver, an MCU, a temp sensor and an accelerometer. The receiver is used to collect the 915Mhz signal that powers the onboard hydrometer electronics. The MCU is used to collect and transmit relevant data from the hydrometer module. The accelerometer is used to collect the hydrometer's axis tilt. The temp sensor is used to acquire the surrounding hydrometer temperature. The base station module is compromised of the power harvesting transmitter and a raspberry pi. The transmitter is used to supply a 915MHz power signal to the hydrometer module. The Raspberry Pi is used to collect, process, and display the hydrometer data. This information is posted to a local python web server for user access.

2.2 Design Analysis

The design incorporates several subsystems: The base station software, the energy harvesting circuitry, the computational circuitry, and the embedded firmware. All these subsystems were designed in parallel and implemented in steps. This proved to be an effective method of design because it was both time efficient and done in enough steps to improve the overall system design by the end of the project.

Unfortunately, due to shipping delays, the final assembled system will not be available in time for the conclusion of this project. However, due to the design process we chose, we have a fully operational system that includes every subsystem working at once.

2.3 Development Process

For the mechanical design, we used FreeCAD. For the PCB design, we used KiCad. For the Firmware and software, we used an SDK from Silicon Labs (Simplicity Studio 4) and the IDE on the Raspberry PI. For the RF design, Advanced Design Systems was used.

2.4 Conceptual Sketch

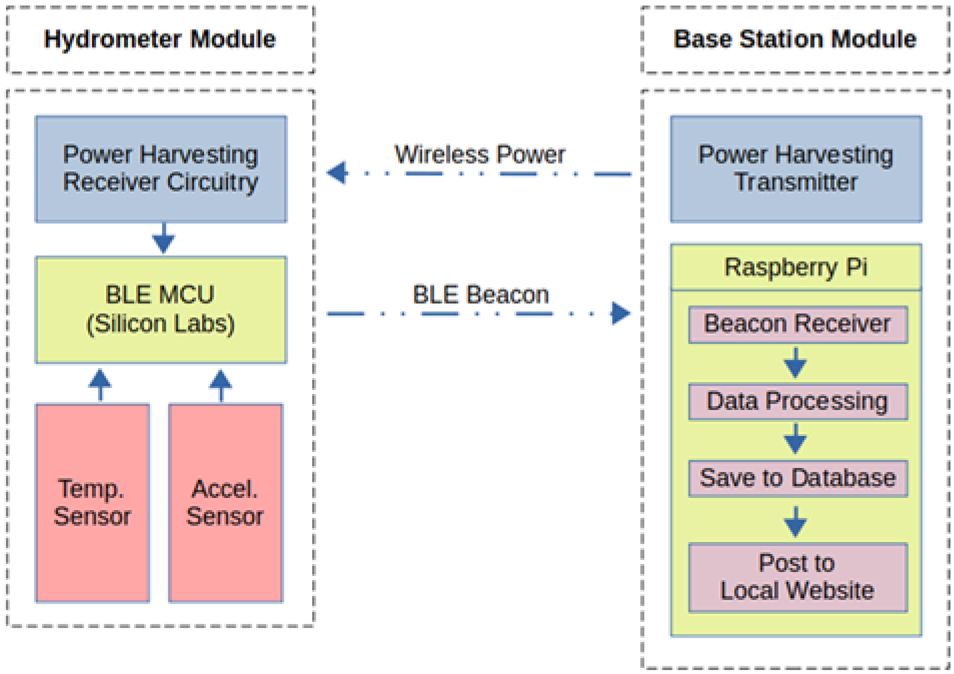


Figure 2.4.1 - System Diagram

3. Statement of Work

3.1 Previous Work and Literature

There is a product called Tilt Digital Hydrometer and thermometer that has the same functionality we are attempting to achieve. The device allows users to wirelessly read specific gravity and temperature. Specific gravity is measured using an accelerometer on this device. The device is powered with a battery.

The hydrometer we have worked on developing does not use a battery. It can take the readings that the Tilt hydrometer can take and has wireless communication with a user interface. Because the hydrometer we have designed is battery less, it implements an energy harvester to wirelessly receive energy which means less user maintenance.

3.2 Technology Considerations

In comparing our design to the alternative market offerings, the major difference is that our design does not need a battery. In the "Tilt" brand hydrometer their unit uses a lithium CR123A battery with an expected lifetime of 3 months in the first-generation units and 12 months in the second generation and later units. In comparison, our system does not use batteries, so it has a lifetime measured in years rather than months. Our battery less hydrometer will not require the user to have routine maintenance (like changing the battery or tracking battery life).

3.3 Task Decomposition

- Layout basic system diagram
- Determine sensors needed
- Select main MCU
- Determine mechanics of tilt system
- Select sensor components
- Design base station layout
- Determine base station hardware
- Write and test some code on 'Gecko' development board
- Gather data on power harvesting capabilities
- Design system schematic
- Design and order prototype PCB
- Determine how to fully encapsulate system
- Design the encapsulating housing
- Order prototype hardware
- Write prototype code
- Assemble prototype
- Test prototype
- Lessons learned and next steps

3.4 Possible Risks and Risk Management

The antenna that will be used for energy harvesting is 6 inches and we are not sure if it will work when it is submerged in water. 95% of the Tilt hydrometer is submerged in water so this may lead to some issues for our implementation. The cost of the energy harvesting circuit is also a concern. The price of our device needs to be around $140 and we are concerned with this extra functionality driving up our cost. If the antenna cannot operate in water, we are concerned with how accurate our device will be with 6 inches of it above water.

3.5 Project Proposed Milestones and Evaluation Criteria

Milestone 1 (Wireless Power Harvesting): Determine the environment that the device is capable of harvesting energy in (water, air, plastic shield, etc.).

Tasks:

- Confirm functionality of power harvesting circuitry in different environments
- Design power harvesting circuitry based on the designs provided by client
- Transmit 915 MHz signal through empty plastic container (amount of time for LED to turn on)
- Transmit 915 MHz signal through plastic container with water (amount of time for LED to turn on)
- Based on the results of the above tests determine an optimal design for the power harvesting circuit

Milestone 2 (Power system function): Analyze the power supply circuitry of our circuit and determine the functional time our microcontroller will have to collect data, perform the necessary calculations, and transmit the data back to the user interface over Bluetooth. Determine the optimal voltages to switch the circuit on and off.

Tasks:

- Determine an optimal capacitor based on cycle life, charging time, and discharge rate
- Measure the discharge rate of the capacitor
- Measure capacitor charging time

Milestone 3 (Microcontroller Function): After determining how much time the microcontroller will have to perform calculations, we will need to confirm that the hardware can complete it in the given allowed time. Additionally, we need to consider the memory we need for the code.

Tasks:

- Determine microcontroller on/off functionality

- Determine the size of written code for calculations and sensor libraries
- Cycle the microcontroller on and off and test functionality

Milestone 4 (Sensor Accuracy): To provide a device that is accurate the sensors may need to be calibrated. For example, if the temperature of the accelerometer increases this may affect the way that calculations must be done to determine an accurate position reading. The accuracy will also be heavily influenced by the mechanical design of the battery less hydrometer so before calibration can be finalized, we need to finalize the mechanical design.

Tasks:

- Calibrate Sensors
- Test operation of the accelerometer in different temperature environments
- Test operation of the accelerometer in mechanical design

Milestone 5 (PCB Layout/Chip pinouts): Decide on IC's and associated circuitry that will be used for all functional areas of battery less hydrometer. Design circuitry and layout PCB of battery less hydrometer based on chosen IC pinouts and associated circuitry as well as desired mechanical functionality.

Tasks:

- Choose IC's
- Design circuitry (Simulate and test everything that is possible to)
- Layout PCB
- Test individual IC's that are possible to test without PCB
- Test PCB functionality

Milestone 6 (Mechanical): The hydrometer must tilt based on the density of the liquid that it is in. This seemingly simple task will be quite challenging. We must consider weight, buoyancy, and the shape of the mechanical design.

Tasks:

- Design container
- Test operation of the accelerometer in mechanical design

3.6 Project Tracking Procedures

We will use the milestones discussed in the 3.5 as well as the Gant chart from 4.1.

3.7 Expected Results and Validation

The desired outcome is an accurate battery less hydrometer capable of energy harvesting that communicates data to the user using Bluetooth communication.

Measurements displayed on the user interface will correspond to the temperature and specific gravity of the environment being used to test the battery less hydrometer.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

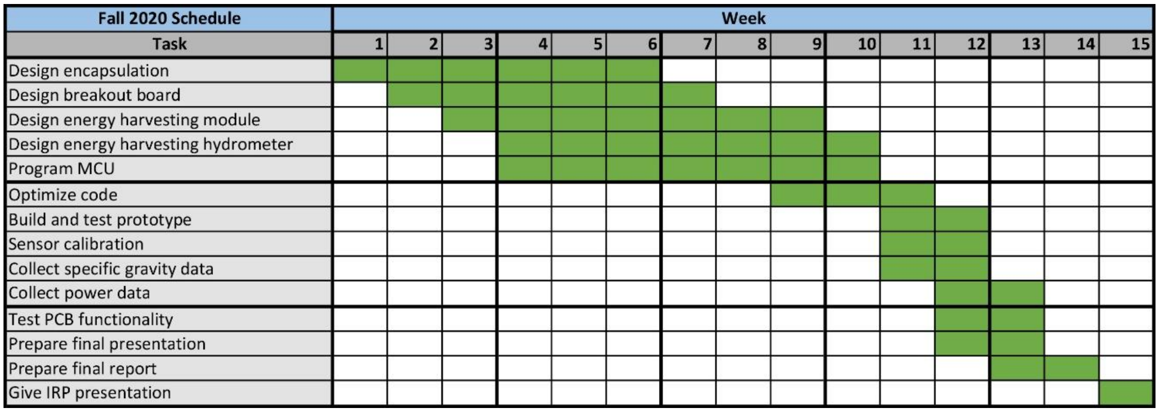


Figure 4.1.1 - Project Gantt Chart

In this timeline you can see that we focused on making sure that we developed sound solutions for our project before purchasing components and trying to build a prototype. Also, within this Gantt chart you can see that we progressed methodically through the project and planned out design and build timelines for different aspects to try to ensure that we could be successful. This timeline differs from our original expectations due to the lingering effects of being remote for most of last semester along with limited lab and equipment access this fall. Due to those issues we were not able to make it as far as originally hoped but were still able to accomplish many aspects of our project.

Additionally, we tried to stay specific in the tasks listed on the Gantt chart to keep us focused on the end goal. We were able to focus on those items and get them accomplished methodically.

4.2 Feasibility Assessment

We found this project to be challenging but feasible. There were a few key challenges in attempting to complete this project. The main challenge was harvesting enough power from the transmitter to consistently and reliably transmit data. The other challenge was properly designing the mechanics of the tilt to ensure that the sensor tilts enough for precise specific gravity measurements. Additionally, a new challenge we had was needing to adjust the schedule and tasks to deal with the pandemic as being unable to work in the labs had greatly limited our hardware development.

4.3 Personnel Effort Requirements

Table 4.3.1 - Personnel Efforts

Task	Member Responsible	Effort Required
Research Workings of Similar Products	All	Fair; but a very important task to make sure we understand the functionality
Layout basic system diagram	Jensen	Significant; but brief to get the layout done.
Determine sensors needed	Chris M.	Fair; takes some time to reconcile the diagram with available sensor types.
Research Tilt Mechanism Physics	Jensen	Significant; it was challenging to find information on utilizing tilt in such an application.
Select the main MCU	Griffin	Significant; due to the abundance of options.
Determine mechanics of the tilt system	Tilden	Very challenging; as there is little information available about it.
Select sensor components	Griffin	Fair; to determine the proper packages and specifications.
Determine the base station hardware	Jensen	Fair; required finding hardware capable of our requirements.
Write and test some code on 'Gecko' dev board	Josh	Significant; required writing custom Bluetooth beacon code.
Gather data on power harvesting capabilities	Chris P.	Straightforward; but time consuming to collect data.
Design prototype test board schematic	Griffin	Challenging; needed to take all parts and properly layout circuitry.
Design and order prototype PCB	Griffin	Very challenging; required very specific design choices.
Determine how to fully encapsulate the system	Tilden	Challenging; required looking at novel manufacturing methods.
Mockup tilt test prototype	Jensen	Significant; required finding creative solutions for a manual measurement method
Order prototype hardware	Chris P.	Straightforward; just a matter of ordering the BOM from Kicad.
Write the prototype board code	Josh	Challenging; required highly efficient and fast code for the sensor.

Final Presentation	All	Significant; the final presentation will be important to demonstrate our progress.
Final Report	All	Fair; this document builds upon the other versions with added updates.
Review lessons learned and next steps	All	Significant; this will be important for our overall success in the fall.

4.4 Other Resource Requirements

Apart from financial resources we will need the following resources:

- Lab Access to the power harvesting development kits.
  - It is necessary to gather data on the power harvesting capabilities of these modules.
  - It will be extremely useful to have hands on experience with these modules before building them into our sensor.
- Equipment
  - Our prototype and final designs will require the use of a reflow oven to properly construct.
  - To encapsulate the entire system, we will need a 3D printer to make a functionally accurate prototype.
- Time
  - We will need to make time available outside of our work schedules to contribute to this project.
- Software Access
  - We will be using Kicad for PCB design.
  - We will be using FreeCAD for the mechanical design.
  - Advanced Design Systems for RF design

4.5 Financial Requirements

For our project we have the following cost estimate of what it will take to go from the initial design to a final product.

Table 4.5.1 - Financial Analysis

Development Hardware	$30
Power Harvesting Hydrometer • PCB - $15 • Electrical Components - $100 • Housing - $15 • Energy harvesting Antenna - $40	$170

Base Station with Power Transmitter	$150
Miscellaneous	$50
Project Total (Estimate)	**$400**

5. Testing and Implementation

5.1 Interface Specifications

The major consideration with interfaces in this product is concerned with power consumption. In order to limit the power consumed, the communication between devices will need to be fast. A model of total time required to complete the communication in each cycle should be made.

There will be serial communication via an I2C bus between the microcontroller and the accelerometer and temperature sensor. The functionality of this serial bus will need to be evaluated and modeled.

There will be wireless communication with the base station via a connectionless Bluetooth transmission. The current consumption and time required to transmit should be measured. Additionally, the environment it will be transmitting from is near fluids which can limit signal integrity. The ideal transmission power (in dBm) should be selected based on test results.

5.2 Hardware and Software

The hardware for this product will exist on a printed circuit board (PCB) that hosts the embedded system. This will include the energy harvesting subsystem, the power management circuitry, the microcontroller and sensors, and the wireless communication subsystem. The software will include sensor data collection, calibration, computation, wireless communication, and an exit flag.

Hardware:

1. *Energy Harvesting:*

   The product will utilize a PowerCast system that operates in the 915 MHz frequency band. The energy harvesting dipole antenna is designed by the PowerCast Corporation along with the P2110B PowerHarvester Receiver.

2. *Power Management:*

   The power management circuitry exists on the P2110B module. Once the voltage across the capacitor reaches a selected value, the voltage regulator will be activated until either the voltage drops too low or the shutdown flag is activated.

3. *Microcontroller and Sensing:*

   The microcontroller (MCU) selected for this product is the EFR32BG13. This product offers low power consumption along with fast boot times. Additionally, it has an integrated Bluetooth 5 physical layer (PHY) and 2.4 GHz transceiver. The sensors that will be selected will utilize an I^2C serial bus for communication with the MCU. There is a 6-axis accelerometer and temperature sensor on the device which will be utilized in the calculation of the specific gravity of the current fluid to within ±0.001 [g/cm^3] accuracy.

4. *Wireless Communication:*

   The wireless communication will utilize a connectionless Bluetooth session called a Bluetooth Beacon. In this configuration, the product will transmit the data at 2.4 GHz under the Bluetooth 5 protocol without connecting to the base station.

Unfortunately, due to the coronavirus and inaccurate shipping times we did not receive the custom PCB that we designed for the hydrometer module. Because of this inconvenience we were forced to run the energy harvesting circuitry separate from the rest of the hydrometer module.

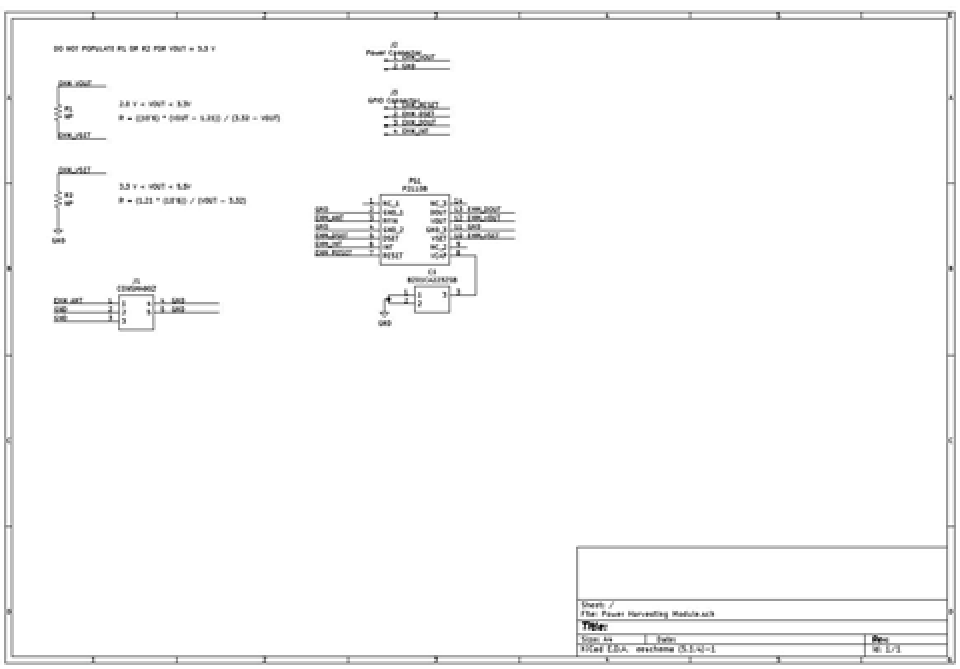


Figure 5.2.2 - Energy Harvesting Module Schematic

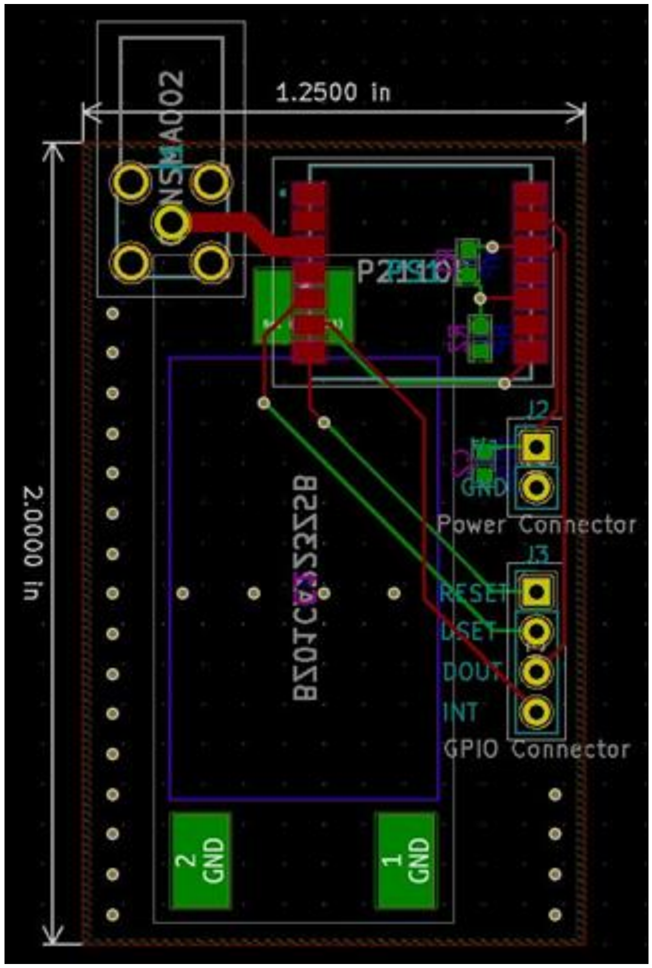


Figure 5.2.3 - Energy Harvesting Module Layout

The energy harvesting module (EHM) incorporates the super capacitor that holds the charge used to power the microcontroller, sensors, and RF circuitry and the PowerCast's antenna and module. The measurements and results presented in this document utilize the printed circuit board (PCB) shown in figures 3b and 3c and the Silicon Lab Thunderboard Sense 2 to harvest the energy, sample the orientation data, and transmit the Bluetooth iBeacon. The Thunderboard Sense 2 has the same microcontroller and accelerometer as is present on the final PCB designed for this project. These two components, the EHM and the Thunderboard Sense 2, are a comparable model of our complete system.

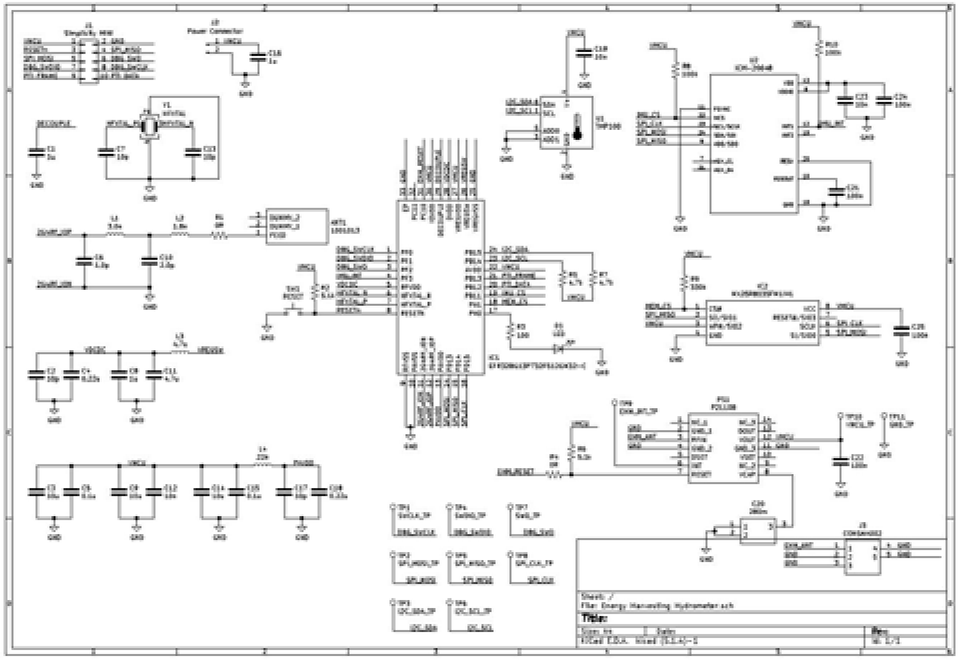


Figure 5.2.4 - Energy Harvesting Hydrometer PCB Layout

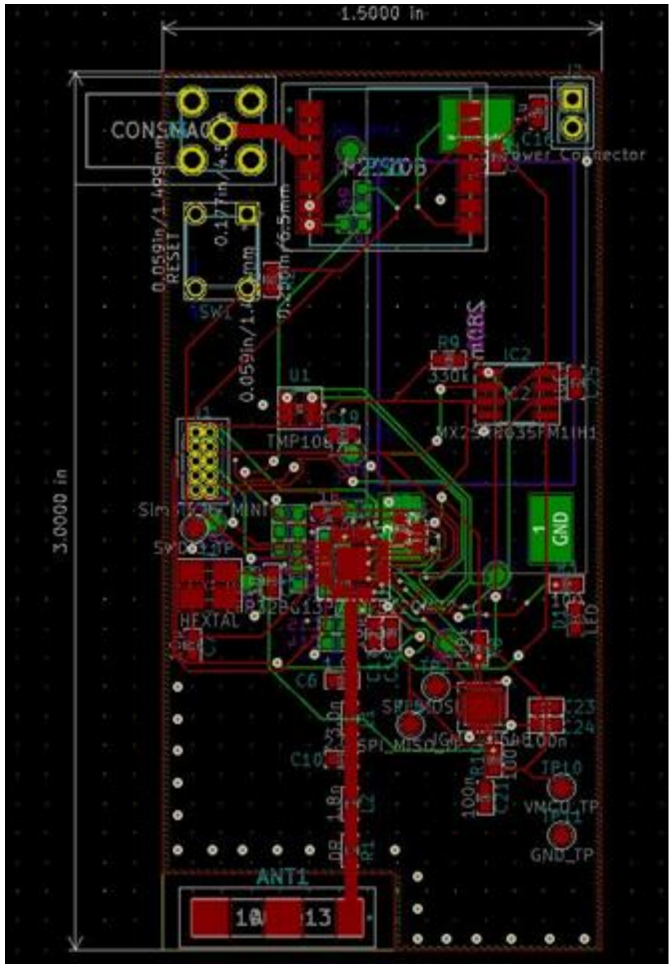


Figure 5.2.5 - Energy Harvesting Hydrometer PCB Layout

The completed board will utilize PowerCast's 915 MHz dipole antenna to harvest the transmitted power. This will be harvested and regulated by PowerCast's module that is incorporated into our complete system. This module will charge a capacitor until the voltage across the capacitor reaches 1.25 V and maintains power until either the voltage across the capacitor reaches 1.02 V or the interrupt pin is driven high. While power is active in the main circuitry, the device will collect orientation data from the ICM-20648 and transmit it via Bluetooth iBeacon. Once the data has been transmitted, the interrupt pin will be driven high and the EHM will begin to charge the capacitor again.

The super capacitor was sized appropriately for this device. Per the P2110B PowerCast Module's datasheet, the capacitance required can be calculated with the following equation:

$$C = 15 * V_{out} * I_{out} * t_{on}$$

Using the standard configuration of the P2110B, the output voltage is 3.3 V. Data collected from testing allowed us to determine that the average current consumption is about 3.9

mA and the time required to wake up, perform the measurements, and transmit the date is about 2.5 s. This results in a required capacitance of 480 mF. To allow for a margin of error, the next size up in the chosen capacitor family – 560 mF – was chosen for this project.

Firmware:

1. *Sensor Data Collection:*

The software collects the orientation and temperature data using the I²C protocol and existing libraries. The libraries used to access the sensor functions were modified versions of existing code to minimize memory utilization on the device. The DCM (Direct Cosine Matrix) is defined by data from the gyroscope and the code for this from SI Labs can be seen below (Note: the DCM is initialized to zero in initialization).

```
void IMU_dcmRotate(float dcm[3][3], float angle[3])
{
  int x, y;
  float um[3][3];
  float tm[3][3];

  um[0][0] =  0.0f;
  um[0][1] = -angle[2];
  um[0][2] =  angle[1];

  um[1][0] =  angle[2];
  um[1][1] =  0.0f;
  um[1][2] = -angle[0];

  um[2][0] = -angle[1];
  um[2][1] =  angle[0];
  um[2][2] =  0.0f;

  IMU_matrixMultiply(tm, dcm, um);

  for ( y = 0; y < 3; y++ ) {
    for ( x = 0; x < 3; x++ ) {
      dcm[y][x] += tm[y][x];
    }
  }

  return;
}
```

```
void IMU_matrixMultiply(float c[3][3], float a[3][3], float b[3][3])
{
  int x, y, w;
  float op[3];

  for ( x = 0; x < 3; x++ ) {
    for ( y = 0; y < 3; y++ ) {
      for ( w = 0; w < 3; w++ ) {
        op[w] = a[x][w] * b[w][y];
      }
      c[x][y] = op[0] + op[1] + op[2];
    }
  }

  return;
}
```

2. *Computation*:

The computation of the specific gravity is performed on the base station. As a result, the hydrometer transmits the x-axis (roll) angle and the temperature data for calculation and calibration to be performed by the end-system. The calculation of the roll, pitch, and yaw from the DCM (Direct Cosine Matrix) obtained from the gyroscope is shown in the code below.

```
void IMU_dcmGetAngles(float dcm[3][3], float angle[3])
{
  /* Roll */
  angle[0] =  atan2f(dcm[2][1], dcm[2][2]);

  /* Pitch */
  angle[1] = -asinf(dcm[2][0]);

  /* Yaw */
  angle[2] =  atan2f(dcm[1][0], dcm[0][0]);
  //printf("IMU: ANGLES: %f,%f,%f\r\n", angle[0], angle[1], angle[2] );
  return;
}
```

The angles are converted into degrees and multiplied by 100 before preparing them for transmission as seen in the code below. These angles range from negative to positive one-hundred and eighty degrees. There are also two decimal places in each value.

```
void IMU_orientationGet(int16_t ovec[3])
{
  ovec[0] = (int16_t) (100.0f * IMU_RAD_TO_DEG_FACTOR * fuseObj.orientation[0]);//x-axis
  ovec[1] = (int16_t) (100.0f * IMU_RAD_TO_DEG_FACTOR * fuseObj.orientation[1]);//y-axis
  ovec[2] = (int16_t) (100.0f * IMU_RAD_TO_DEG_FACTOR * fuseObj.orientation[2]);//z-axis

  return;
}
```

3. *Wireless Communication:*

The wireless communication is performed over a Bluetooth 5 Beacon. We only had room to transmit 16 bits of orientation data, so we decided to design the mechanical mechanism to tilt on the x-axis and we only transmitted the roll data from ovec[0]. After collecting the orientation and temperature data we converted it from on 16-bit value to two 8-bit values then passed them into the beacon function. Then the code transmits it to the base station through the major (orientation) and minor (temperature) numbers of the Bluetooth Low-energy Beacon signal. We used the iBeacon data format which can be seen in the below code and figure.

```
bcnBeaconAdvData
  = {
  /* Flag bits - See Bluetooth 4.0 Core Specification , Volume 3, Appendix C, 18.1 for more details on flags. */
  2,  /* length  */
  0x01, /* type */
  0x04 | 0x02, /* Flags: LE General Discoverable Mode, BR/EDR is disabled. */

  /* Manufacturer specific data */
  26,  /* length of field*/
  0xFF, /* type of field */

  /* The first two data octets shall contain a company identifier code from
   * the Assigned Numbers - Company Identifiers document */
  /* 0x004C = Apple */
  { UINT16_TO_BYTES(0x004C) },

  /* Beacon type */
  /* 0x0215 is iBeacon */
  { UINT16_TO_BYTE1(0x0215), UINT16_TO_BYTE0(0x0215) },

  /* 128 bit / 16 byte UUID */
  { 0xE2, 0xC5, 0x6D, 0xB5, 0xDF, 0xFB, 0x48, 0xD2, \
    0xB0, 0x60, 0xD0, 0xF5, 0xA7, 0x10, 0x96, 0xE0 },//byte[5],byte[4]},

  /* Beacon major number */
  /* Set to Orientation Data*/
  { byte[1] , byte[0] },

  /* Beacon minor number */
  /* Set to Temperature Data*/
  {buffer1[1], buffer1[0]},

  /* The Beacon's measured RSSI at 1 meter distance in dBm */
  /* 0xD7 is -41dBm */
  0xD7
  };
```

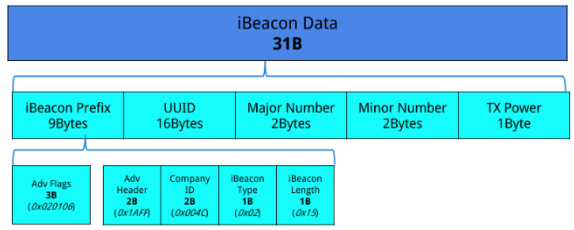


Figure 5.2.6 - iBeacon Data Structure

Software:

1. *BLE Beacon Scan*

   The beacontools library was the primary python library used to collect the BLE beacon data. Accelerometer and temperature output are received in the iBeacon major/minor numbers. This raw data is used to calculate the specific gravity of the surrounding fluid using a python script.

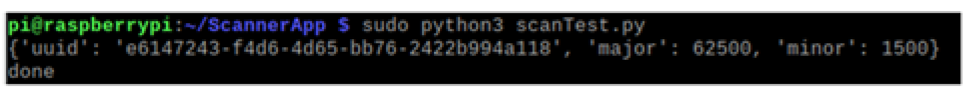


Figure 5.2.6 – Bluetooth Transmission

2. *Post to Web Server*

   The specific gravity is posted to the python web server using a flow-based development tool known as Node-Red. Node-Red automatically runs on boot up from the pi and immediately calls the ScanTest program. This program continuously scans for any ibeacon advertisements with the designated UUID. Once the ScanTest program has output a result, a timestamp is added and appended in JSON format to a file on the raspberry pi. This is done so the user would be able to look at and results before the 7-day window that is shown on the graph.

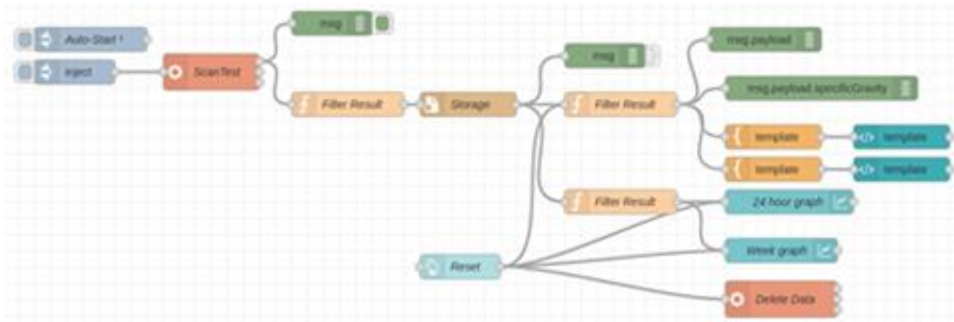


Figure 5.2.7 - Node Red Schematic

Figure 5.2.8 shows the local website that the user would see and interact with. The website shows the last measured specific gravity and the time it received the data. It also shows a graph of the last 24 hours, as well as the last 7 days. The user can click the RESET DATA button that would clear all the graphs and delete any previously recorded data from the logs file.

Figure 5.2.8 – Server Output

5.3 Functional Testing

Mechanical:

- Testing the housing for waterproofing
- Testing how the tilt mechanism reacts to the change in specific gravity

Hardware:

- Test the Thunderboard power consumption
- Test the system for cycling
- Test the energy harvesting circuitry for power delivery
- Test the programming circuitry
- Ensure that the sensors are operational and able to be sampled
- Test that the antenna can be transmitted through and received by the base station

Firmware:

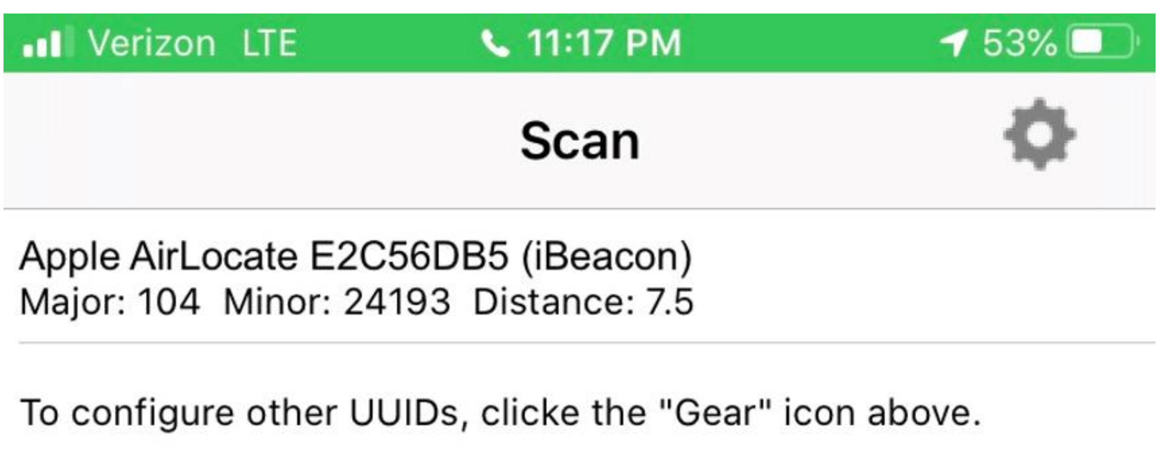
- Sample the sensors
- Transmit the data via Bluetooth iBeacon

Software:

- Receive the Bluetooth iBeacon
- Convert sensor data into the specific gravity
- Upload the data to the web server and display in a real-time graph

We used the retargetserial.h library to read print statements from the serial port to the Tera Term terminal. An example of this data can be seen below. We verified that the temperature and angle measurements were working by doing this. When the board ran for a long time the temperature began to increase on the board, so the measurement was not accurate to the environment. This should not be a concern because the board will cool down while the capacitor is charging up.

To test the iBeacon signal, we used our phones to detect the signal. For the roll positive numbers were between 0.00 and 180.00 and showed up as 0 through 18000. Negative numbers ranged from -0.01 to -180.00 and showed up as 65535 through 47536.



We found the orientation data precision by reading sixty values of the orientation data when the board was powered by through the programming cable and when it was powered by a battery. We averaged these values and subtracted each measurement from the average to find the above deviations. We then averaged the deviations to find the precision. The precision of the computer powered orientation data was 0.008633 degrees, and the precision of the battery powered orientation data was 0.108336 degrees. This is something that we would like to explore further.

5.4 Non-Functional Testing

Performance:

- The capacitors were tested by charging them with the energy harvesting module and using them to power the Thunderboard Sense 2. The time to charge and time to discharge were recorded. Each value was analyzed for its viability in the system.

Compatibility:

- Bluetooth 5 1M PHYs compatible with Bluetooth 4
- The base station is Bluetooth 4.2
- The code developed on the Thunderboard Sense 2 (EFR32MG12) is compatible with EFR32BG13

5.5 Process

Mechanical:

- *Waterproof Housing Tests:*

  The housing was filled with enough weight to sink to the bottom of a tank of water and a paper towel to determine if moisture makes it inside the housing.

- *Tilt Mechanism Tests:*

  The housing was filled with weight to where it floats at a slight angle from vertical in tap water. It has the Thunderboard placed inside the housing and programmed to transmit the angles over iBeacon. For power utilizes a CR2032 coin cell. The fluid in the tank is then brought up to a specific gravity of 1.11 and reduced from there in steps of 0.01 down to a specific gravity of 1.00 here the angles will be recorded. The specific gravity of the tank is changed by diluting a sugar water solution.

Hardware:

- *Power Consumption Tests:*

A 50Ω resistor was placed in series with a 3.3V power source to supply the Thunderboard module with power. The Thunderboard was flashed with the iBeacon code and the voltage drop across the resistor was measured from the time when the Thunderboard was given power to when it completed the first transmission. This voltage drop was then used to calculate the current draw by the Thunderboard.

- *Power Harvesting Testing:*

  The power harvesting capabilities were measured by testing the power harvesting modules with different capacitors at different distances from the power transmitter. We tested the power harvesting modules with 22mF, 50mF, and 100mF supercapacitors as well as a 1F electrolytic capacitor. The voltages across the capacitors were measured as they charged up and recorded for analysis.

- *System Cycling Testing:*

  The power harvesting module with the 1F capacitor was connected to the Thunderboard with the iBeacon code flashed on it. The transmitter was then placed 6 inches away from the harvester module and turned on. The voltage across the capacitor as well as the 3.3V regulated output from the harvester module were both measured and recorded.

5.6 Results

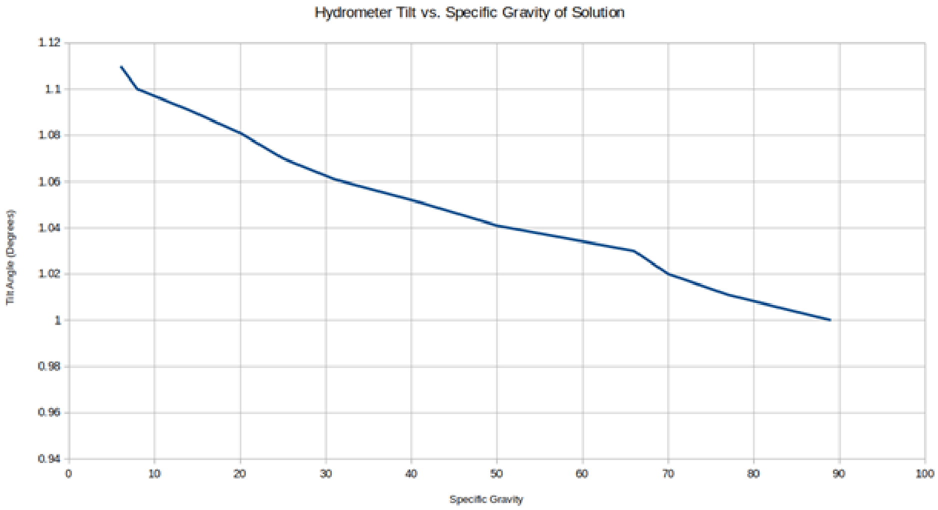
Mechanical:

Figure 5.3.1 – 3D Housing Progression

As we gained familiarity with CAD the encapsulation design progressed and increased in complexity. The result is a sleek three-piece threaded structure that makes testing and calibration a lot easier. The final structure shown has room to put weights (quarters) in both the top and bottom sections to vary the center of gravity and overall mass of the hydrometer.

Figure 5.3.2 – Final Housing

Figure 5.3.3– Tilt Mechanics



Angle to Specific Gravity Relationship Equation:

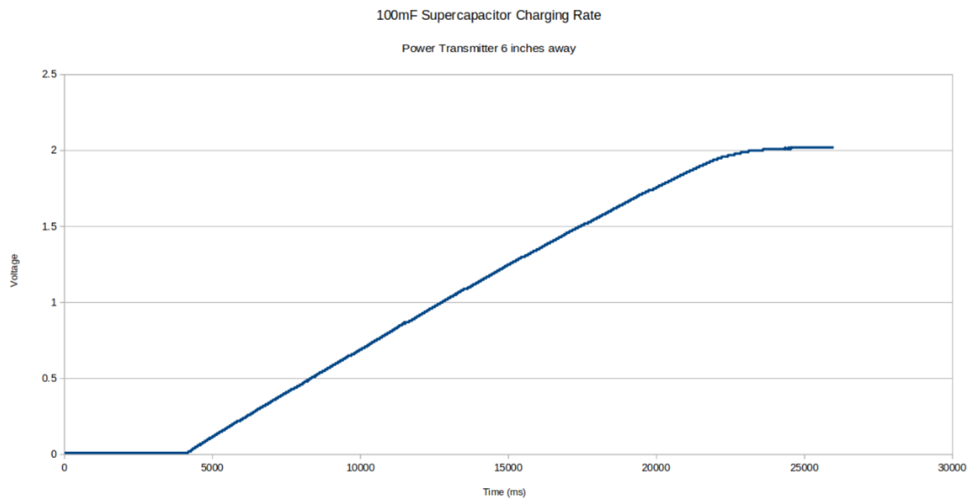
SP = 6.52331044E-06α^2 – 0.00184859α + 1.11611406

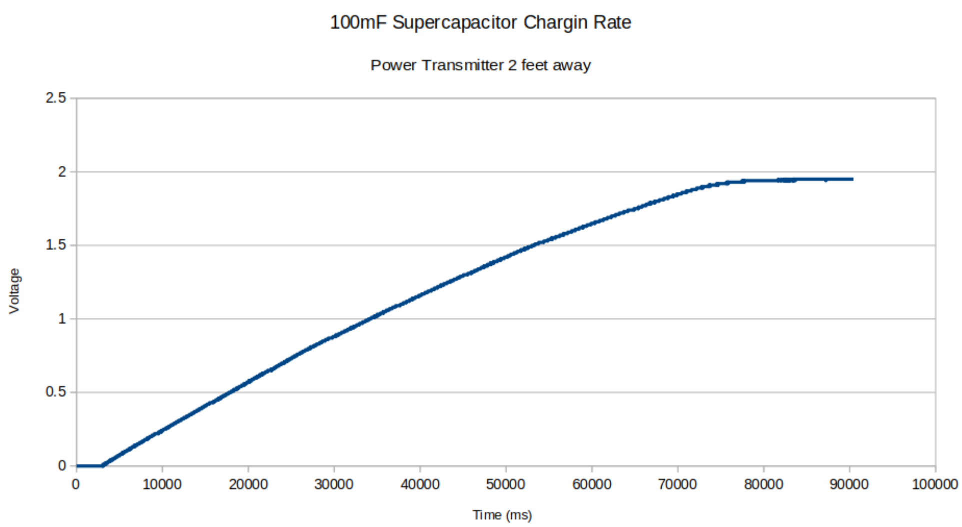
Hardware:

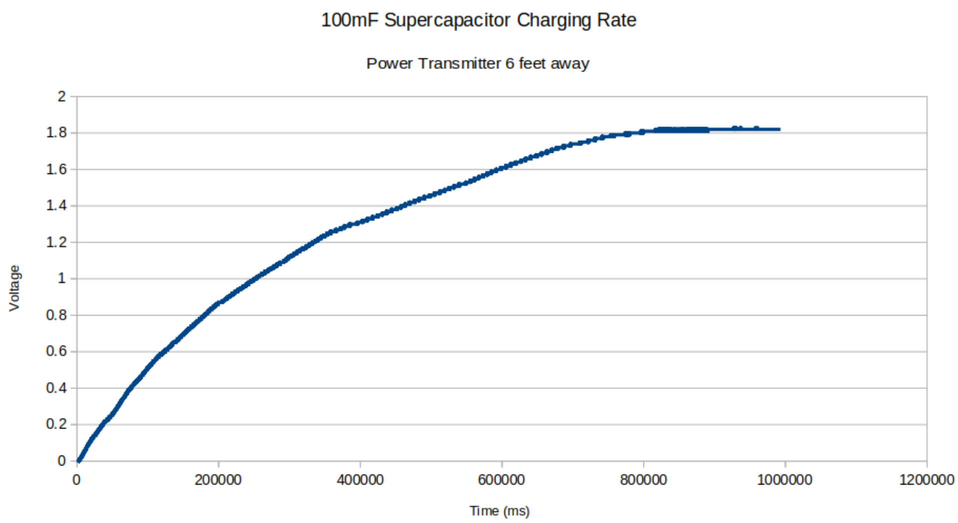
Supercapacitor Charging Data

In learning about the power harvesting module, we chose to test the system's power harvesting capabilities. The data shown below is our finding with using each of those capacitors at different distances. For the electrolytic capacitor we found that it had difficulty charging up if the transmitter was farther than 6 inches away, likely due to the leakage current.

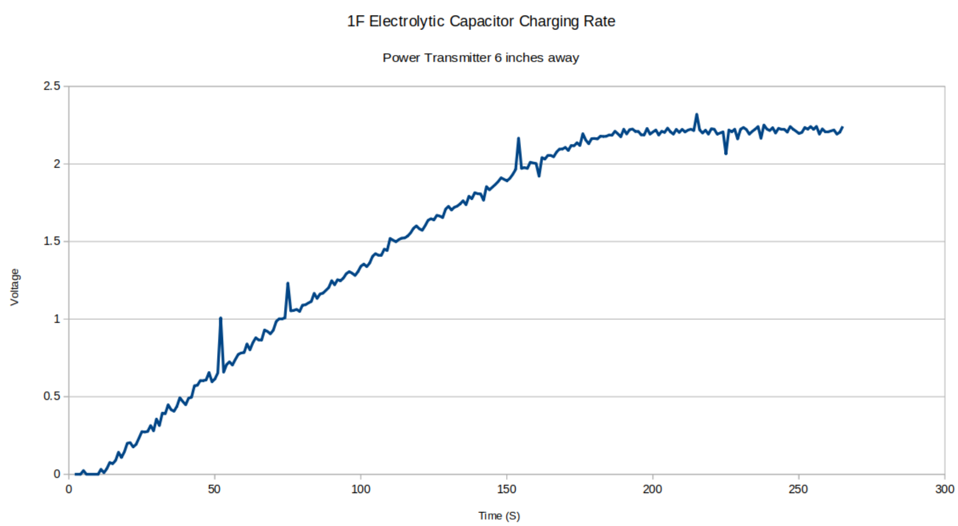
100mF Supercapacitor



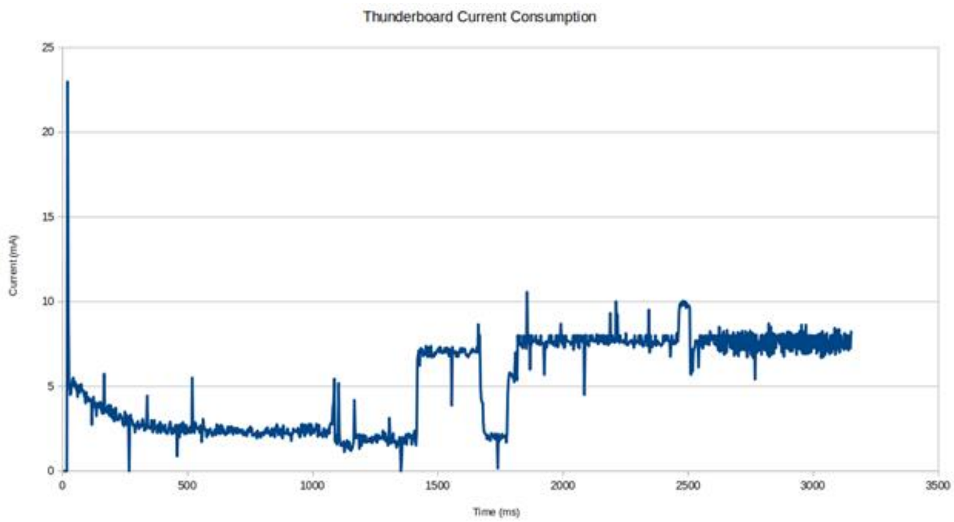




1F Electrolytic Capacitor

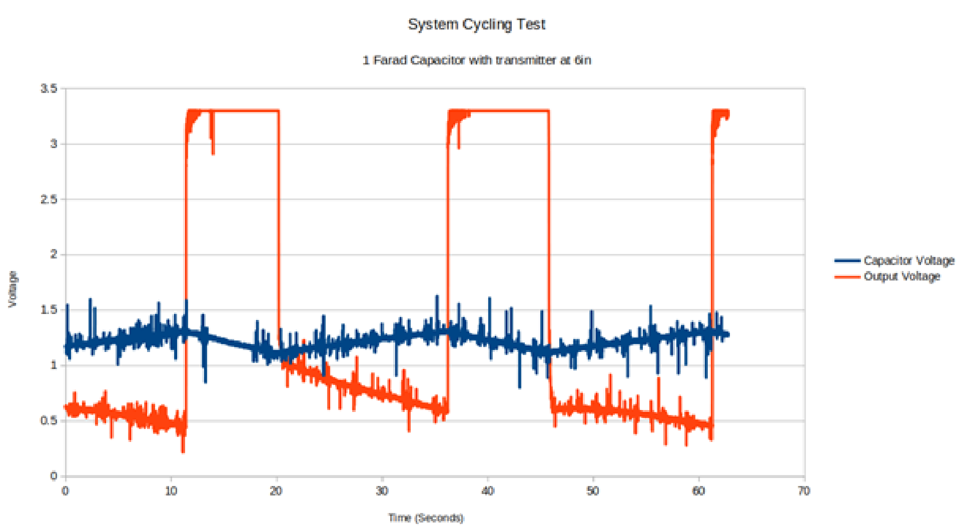


Thunderboard Sense II Energy Usage



In testing the energy usage of the board running our transmit code, we found the average current usage through the first beacon cycle to be approximately 3.9mA. This matches with our other testing data that found a capacitor over 500mF is necessary for the system to work.

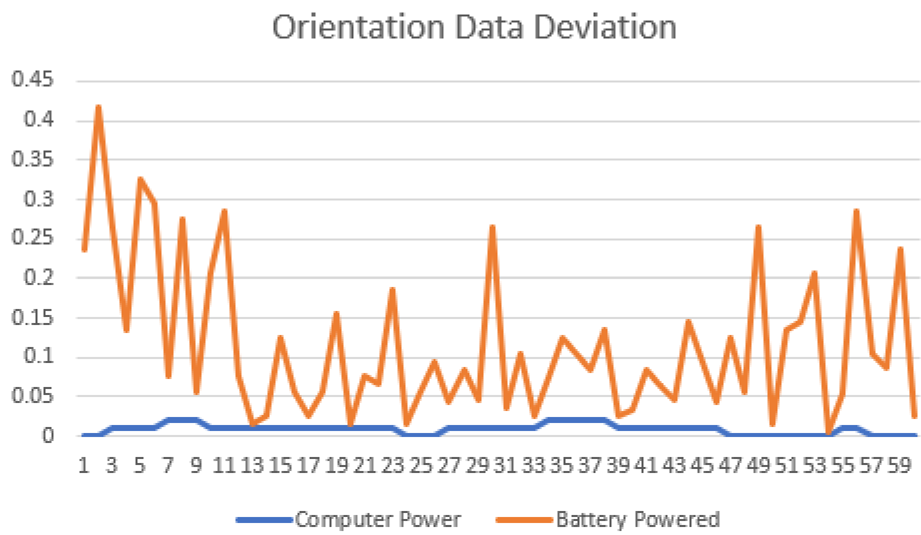
Wireless Powered System Discharge Test



By using a 1 Farad capacitor we were able to test the system fully cycling several times. We found that the power provided by the capacitor was more than sufficient for the thunderboard to turn on and cycle as it was able to transmit multiple times before shutting back down to recharge which matches with our calculations of only a 500mF capacitor being needed.

Software:

Data Collection and Display



We found the orientation data precision by reading sixty values of the orientation data when the board was powered by through the programming cable and when it was powered by a battery. We averaged these values and subtracted each measurement from the average to find the above deviations. We then averaged the deviations to find the precision. The precision of the computer powered orientation data was 0.008633 degrees and the precision of the battery powered orientation data was 0.108336 degrees. This is something that we would like to explore further.

6. Closing Material

6.1 Conclusion

With the project wrapping up, we feel that we have sufficiently designed and tested this device as to meet the requirements of the project. Unfortunately, due to current issues in shipping and ability to meet in person, our final PCB will not arrive before the deadline of the project. However, we have designed this project in steps and have implemented the full design in two pieces, as presented in this document. This consists of a power harvesting portion and a computational portion. Together, they form the full system, and for the purposes of this document, all the data presented is reflective of the final project.

6.2 References

1 Journée J. M. J., and W W. Massie. *Offshore Hydromechanics*. Delft University of Technology, 2001.

2 AN960 EFR32 Matching Guide. Silicon Labs.

3 AN0002.2 EFR32 Series 2 Hardware Design Considerations. Silicon Labs.

4 AN928.2 EFR32 Series 2 Layout Design Guide. Silicon Labs.

5 Silicon Labs Open-source libraries and code repositories

6.3 Appendices

Appendix 1 – Operations and Setup Manual

Operators Manual

Preface: For the operations manual we chose to make the following assumptions about the materials available to the user.

- Raspberry Pi 4 complete with the Node-Red Server pre-loaded
- Traditional glass hydrometer
- Cylindrical tank (larger than 18x6")
- Granulated sugar
- PowerCast Transmitter
- Battery-less hydrometer with integrated PowerCast harvesting antenna

Four steps for setup:

- Raspberry Pi Setup
- Hydrometer Power Receiver test
- Sensor Calibration
- Finished Usage

1. Raspberry Pi Setup
    - Start by plugging in the raspberry pi to a power source and connecting it to a monitor, mouse, and keyboard.
    - Once running, open the web browser and it should immediately go to the sensor monitor information page.
    - If it doesn't automatically open, navigate to the following URL: http://localhost:1880/ui
    - Two empty graphs should be displayed as well as the reset button, angle, and specific gravity outputs. These values will remain blank until the sensor is powered on.

2. Hydrometer Power Receiving Test
    - Once the base station is running, place the hydrometer and PowerCast transmitter at least 6" apart and angle the transmitter so that it faces the hydrometer.
    - After a short period of time, you should begin to see data show up on the base station plots as the Pi receives data from the hydrometer

Figure 1-Transmitter face

3. Sensor Calibration
    - The calibration sequence begins once the hydrometer is receiving power.
    - Begin filling the tank with room temperature water and use the glass hydrometer to record the specific gravity. Place the battery-less hydrometer module in the tank and plug in the transmitter as it faces the tank.
    - Note: Higher transmitter proximity to the hydrometer results in faster transmission times, reducing calibration time. It is recommended to place the transmitter within 2 ft of the tank.
    - Wait 15 minutes and record the last angle transmitted to the base station.
    - Remove the battery-less hydrometer from the tank.
    - Begin adding sugar to the tank and mixing it in. Check the specific gravity with the glass hydrometer frequently until it reaches 1.050
    - Add the battery-less hydrometer back to the tank and wait 15 minutes. Record the final angle transmitted to the base station once again.
    - Remove the battery-less hydrometer from the tank.
    - Continue adding sugar to the tank and mixing it in. Check the specific gravity with the glass hydrometer frequently until it reaches 1.100
    - Add the battery-less hydrometer back to the tank and wait 15 minutes then record the final angle transmitted to the base station once again.
    - On Raspberry Pi, open *ScanTest.py* and set each calibration angle to the recorded angle. *ScanTest.py* can be found on the desktop.
    - Restart the Raspberry Pi.

4. Finished Usage
    - Now the hydrometer can be placed into the tank for measurement.
    - Position the PowerCast transmitter in proximity of the tank. The suggested range is within 6 ft to reduce transmission times.

Appendix 2 – Alternative Versions

As mentioned in section 5.2, the system used to collect the data provided in this document is multiple subsystems combined to create the completed system rather than the fully assembled board. The final board is designed and defined, however. The schematic and layout of the full PCB can be seen in section 5.2 in the Hardware section. In this appendix the bill of materials and pin definitions of the final iteration of the PCB are provided.

Table A – Bill of Materials

Ref	Value
ANT1	1001013
C1	1u
C10	2.0p
C11	4.7u
C12	10n
C13	10p
C14	10u
C15	0.1u
C16	1u
C17	10p
C18	0.22u
C19	10n
C2	10p
C20	280m
C21	100n
C22	100n
C23	10n
C24	100n
C25	100n
C3	10u
C4	0.22u
C5	0.1u

C6	1.0p
C7	10p
C8	1u
C9	10u
D1	LED
IC1	EFR32BG13P732F512GM32-C
IC2	MX25R8035FM1IH1
J1	Simplicity MINI
J2	Power Connector
J3	CONSMA002
L1	3.0n
L2	1.8n
L3	4.7u
L4	22n
PS1	P2110B
R1	0R
R10	100k
R2	5.1k
R3	100
R4	0R
R5	4.7k
R6	5.1k
R7	4.7k
R8	100k
R9	330k
SW1	RESET
TP1	SWCLK_TP
TP10	VMCU_TP
TP11	GND_TP
TP2	SPI_MOSI_TP
TP3	I2C_SDA_TP

TP4	SWDIO_TP
TP5	SPI_MISO_TP
TP6	I2C_SCL_TP
TP7	SWO_TP
TP8	SPI_CLK_TP
TP9	EHM_INT_TP
U1	TMP100
U2	ICM-20648
Y1	HFXTAL

Table B – EFM32BG13 Pin Definitions

Pin	Pin Name	Net Name	GPIO Function	Description
1	PF0	DBG_SWCLK	DBG_SWCLKTCK	GPIO (5V)
2	PF1	DBG_SWDIO	DBG_SWDIOTMS	GPIO (5V)
3	PF2	DBG_SWO	DBG_SWO #0	GPIO (5V)
4	PF3	IMU_INT		GPIO (5V)
5	RFVDD			Radio power supply
6	HFXTAL_N			High frequency crystal input pin
7	HFXTAL_P			High frequency crystal output pin
8	RESETn			Reset input, active low
9	RFVSS			Radio ground
10	PAVSS			Power Amplifier (PA) voltage regulator VSS
11	2G4RF_ION			2.4 GHz differential RF input/output, negative path

12	2G4RF_IOP			2.4 GHz differential RF input/output, positive path
13	PAVDD			Power Amplifier (PA) voltage regulator VDD input
14	PD13	SPI_MOSI	US0_TX #21	GPIO
15	PD14	SPI_MISO	US0_RX #21	GPIO
16	PD15	SPI_CLK	US0_CLK #21	GPIO
17	PA0			GPIO
18	PA1	MEM_CS		GPIO
19	PB11	IMU_CS	US0_CS #3	GPIO
20	PB12	PTI_DATA	FRC_DOUT #6	GPIO
21	PB13	PTI_FRAME	FRC_DFRAME #6	GPIO
22	AVDD			Analog power supply
23	PB14	I2C_SCL	I2C0_SCL #8	GPIO
24	PB15	I2C_SDA	I2C0_SDA #10	GPIO
25	VREGVSS			Voltage regulator VSS
26	VREGSW			Voltage regulator switching node
27	VREGVDD			Voltage regulator VDD input
28	DVDD			Digital power supply
29	DECOUPLE			Decouple output for on-chip voltage regulator
30	IOVDD			Digital IO power supply
31	PC10	EHM_RESET		GPIO (5V)

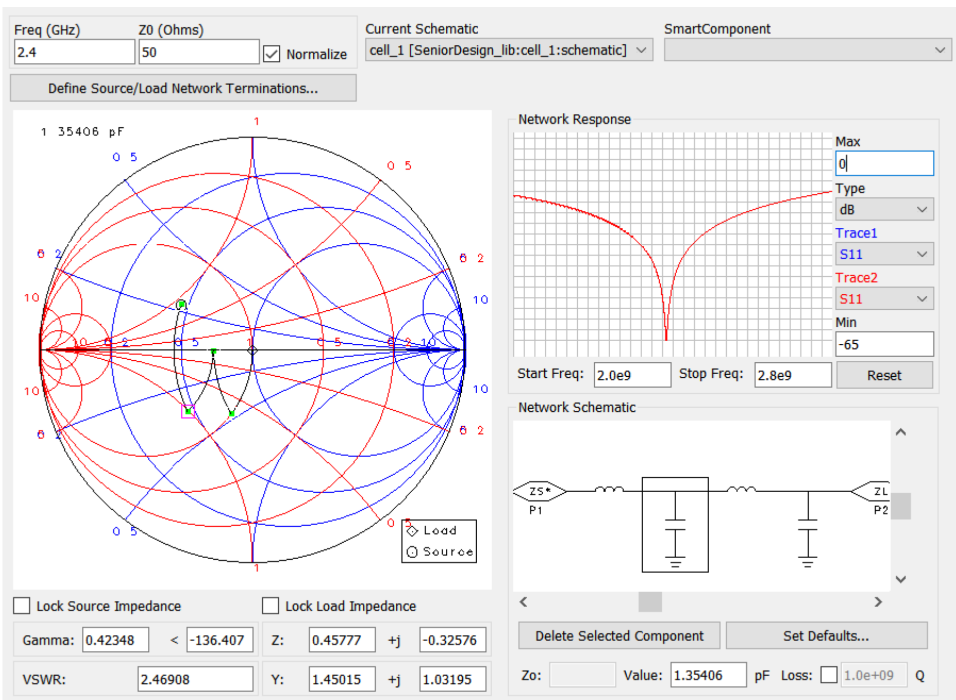
32	PC11			GPIO (5V)
33	VSS	Ground		Expose pad

Appendix 3 – Other Considerations

RF Design

This was the first device any of us had designed that incorporated wireless RF circuitry and there was a significant learning curve. Not only did this board transmit data wirelessly via Bluetooth, it is also powered wirelessly via energy harvesting wireless circuitry. Each of these antennas had to be impedance matched to 50 Ohms which required some research to be done on how to design these matching networks.

Figure A – Antenna Matching Network Design



Using an input impedance of 23+j11.5 - as given by AN930 by Silicon Labs – the impedance matching circuit yielded the circuit seen in Figure A. This circuit matches the input impedance of the transceiver to the 50 Ohm antenna to an impressive –65 dB insertion loss. This matching circuit is approved with the FAA to an output power level of up to 20 dBm.