WIP: Environmental Sensor Network Nodes

EE 496 Final Report
Fall 2018

Renewable Energy and Island Sustainability (REIS)
Smart Campus Energy Lab (SCEL)

Department of Electrical Engineering
University of Hawaii at Manoa

Undergraduate Students: Emily Kane, Emily Lum, Clyde Felix
Faculty Advisor: Dr. Anthony Kuh

Date: December 16, 2018

Abstract:

The objective of this project is to improve the hardware of the third generation weatherbox modules by increasing functionality, data reliability, and ease of use. The fourth generation weathbox implemented new components such as a GPS and a real time clock which can be used to for tracking and organizing data when a node network is implemented. A weatherbox module is to be self-sustaining and provide meteorological data for an extended period of time. Cranberry deployed Version 4.1 to the roof of Holmes Hall and are currently debugging issues with Version 4.1 and starting the design for Version 4.2. The data collected from these boxes will assist in planning future renewable energy installations and will also provide data to be used in forecasting.

1. INTRODUCTION AND MOTIVATION

In 2012, almost 90% of Hawaii's electricity was produced using imported fossil fuels, resulting in the state having some of the highest electricity fees in the nation [1]. Hawaii's average energy costs are about 3 times higher than the national average. To lower these electricity costs, Hawaii must look towards more renewable and sustainable sources of energy. The Smart Campus Energy Lab (SCEL), which operates under the Renewable Energy and Island Sustainability (REIS) program, aims to support a "greener" at the University of Hawaii at Mānoa campus by creating and implementing technology that will assist with the planning of sustainable practices. One project that helps achieve this goal is creating a weatherbox module that can collect and transmit environmental data. These weatherboxes, which are designed and fabricated in SCEL, are to be low cost, power efficient, self-sustaining, and easily mass-produced so that they can be used widely across campus. The data from the weatherboxes can be used to optimize the planning of new renewable energy installations and ensure they are implemented in the most beneficial and efficient way.

2. OBJECTIVE

The main objective of the Smart Campus Energy Lab is to provide resources to help assist with the implementation of sustainable energy practices at UH Mānoa. The focus of the Environmental Sensor Network Nodes project is to create a network of weatheboxes that provide accurate meteorological data and span the university's campus. The data collected from these weatherboxes include temperature, humidity, solar irradiance, and pressure. Then, they are all transmitted to the SCEL server using a wireless communications module.

This semester, the main objective of the Environmental Sensor Network Nodes project is to improve the hardware of the third generation sensor module, which is known as the Cranberry weatherbox. Cranberry recently implemented a GPS and real time clock to improve the functionality. We planned to populate, debug, and deploy a Cranberry Version 4.1 board and to start designing Cranberry Version 4.2 To meet these objectives, the team created a Gantt Chart that is shown in the figure below.

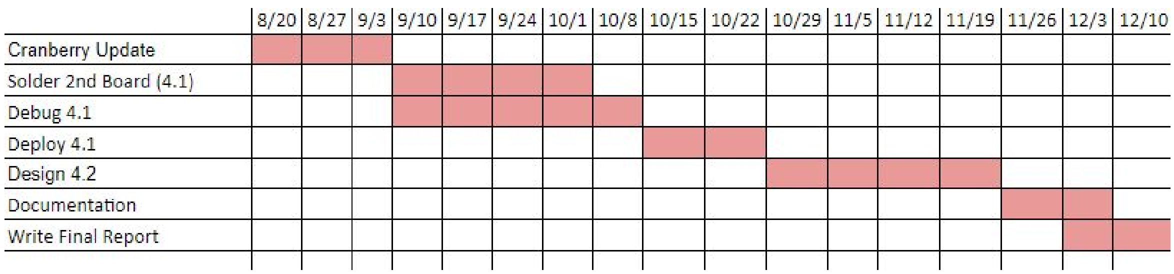


Figure 1: Gantt Chart for Cranberry (Fall 2018)

<u>3. HARDWARE DESIGN OVERVIEW</u>

The first generation weather box in SCEL is Apple, and the main motivation of Cranberry is to improve upon the hardware of Apple by reducing size and cost, and using a more efficient power system. The design of Cranberry 4.1 consists of two 2.375" x 2.375" printed circuit boards (PCBs) mounted on top of each other using headers. The stacked boards allow for a smaller footprint and easier repair than if it was combined on one board. The main board consists of the microcontroller unit (MCU), Xbee, Global Positioning System (GPS), and power components. The sensor board has the power system, weather sensors, real time clock, and LEDs used for debugging. In the figure below is the hardware block diagram for Cranberry 4.1. It shows the earlier stated component along with the connections of the two boards.

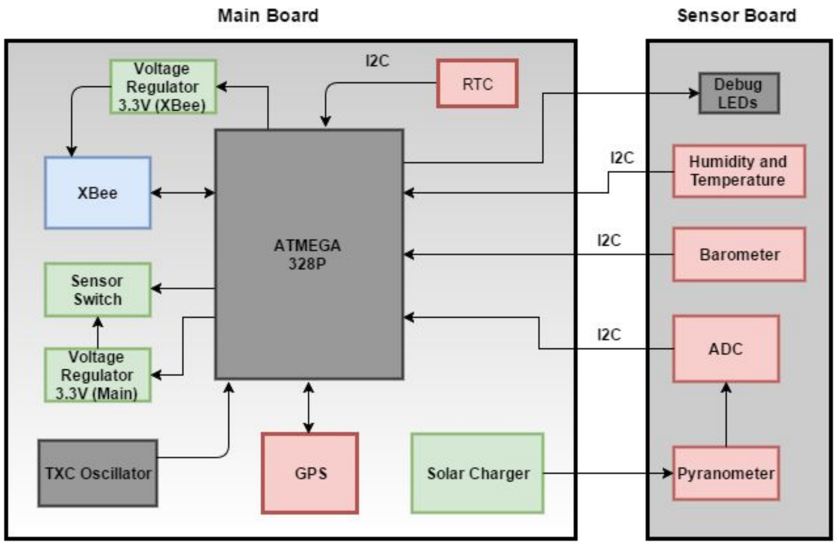


Figure 2: Cranberry Hardware Block Diagram

The revision history for the Cranberry board is below, with the following boards used and referenced to complete the current design:

- Cranberry Version 3.5
  - Designers: Brandon Amano and Kim Pee Castro (Completed: SP2016)
  - Fixes several design flaws and builds upon design of Version 3.2
- Cranberry Version 4.0
  - Designers: Jennifer Chun, Joslyn Hamada, and Emily Lum (Started: FA2016)
  - Builds upon Version 3.5 design and add components such as GPS and real time clock for increased functionality.
- Cranberry Version 4.1
  - Designers: Jennifer Chun, Joslyn Hamada, Emily Kane, and Emily Lum (Started: FA2017)

○ Fixes wiring issues of Version 4.0 and changes layout for easier use

3.1 Overview of Main Board

The main board is the bottom of Cranberry's two stacked boards, it has the MCU, XBee transceiver, and the GPS. Subsections 3.1.1, 3.1.2, and 3.1.3 describe the components of the main board further.

3.1.1 Microcontroller - ATMEGA328P

The main component of the Cranberry board is the MCU, Cranberry uses a Atmel ATMEGA328P microcontroller. The package chosen has a 32-pin VQFN, and it operates on a supply voltage of 1.8V - 5.5V. Based on the datasheet, the MCU contains 23 GPIO lines and is I2C compatible [2]. Even though the microcontroller has a low current draw, and external quartz crystal oscillator is also used to reduce the clock speed from the default 16MHz to 8MHz. The microcontroller circuit diagram is shown in Figure 3 below.

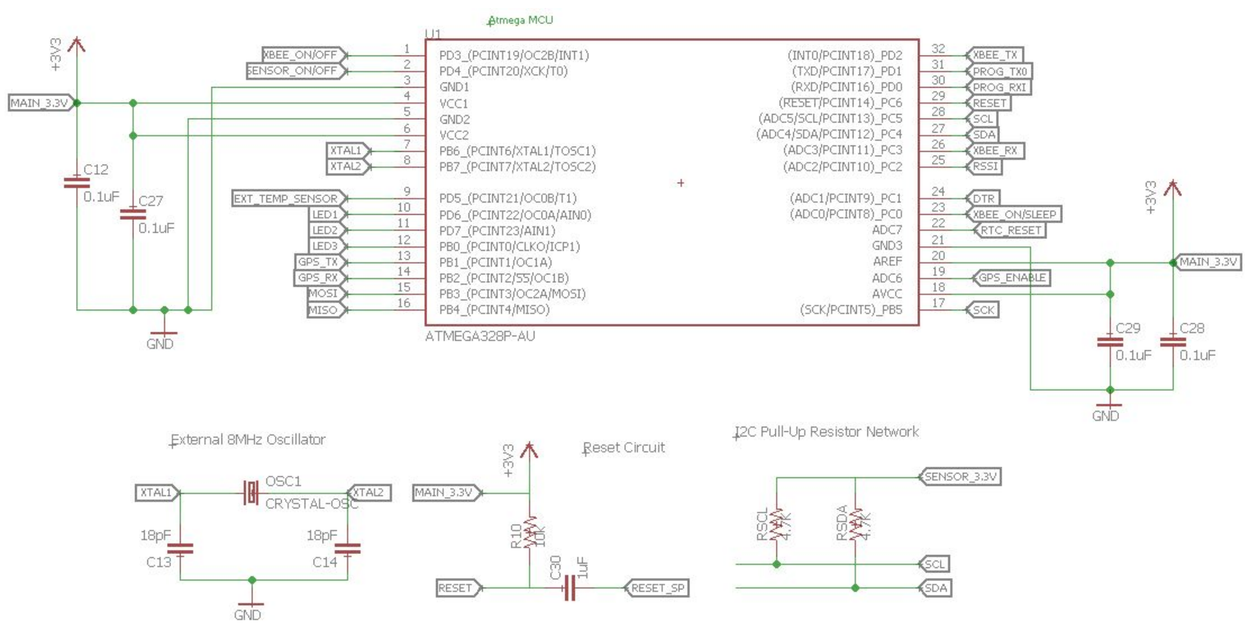
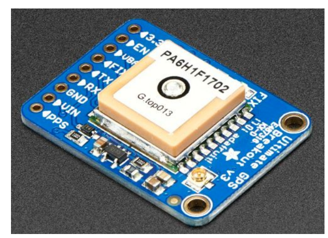


Figure 3: Microcontroller Circuit Diagram

3.1.2 XBee Pro S2B Transceiver

To transmit data, the Cranberry weatherbox uses the XBee Pro S2B RF module. It operates using the ZigBee protocol. One benefit of the XBee is that is requires minimal power while still providing reliable data, it is also low cost [3]. The XBee has transmissions distances of about 90m indoor, and 2mi outdoors with a line-of-sight environment. The XBee operated in API mode, and it sends data in a packet-like structure every 30 seconds, which allows for addressing, parameter setting, and packet delivery feedback.

3.1.3 GPS - Adafruit Ultimate GPS Breakout



Adafruit's Ultimate GPS Breakout board is going to be used to keep track of the location of each weatherbox that is deployed. Since the GPS has a very high power consumption (20mA), after it is initialized it will be put to sleep. This can be done through the ENABLE pin that will be controlled by the MCU. The Apple weatherbox uses the same model, so, the firmware for the GPS has already been written. It's communication model is I2C. Cranberry chose the breakout board instead of the GPS module because of how it is packages. The package pins are under the component which would make soldering more complicated.

3.2 Overview of Sensor Board

The sensor board is the other half of Cranberry's overall design, it stacks on top of the main board and has pressure, temperature, humidity, and solar irradiance sensors. It also includes the power system and a real time clock. The subsections below describe the sensors in more details.

3.2.1 Humidity and Temperature - HIH6131

The HIH6131 is a single sensor that measures both humidity and temperature. The error for humidity is +/- 4.0% and +/- 0.5℃ for temperature [4]. The sensor has a temperature compensation of 41℉ - 144℉ which ensures that the readings are accurate. It also has a low power consumption of 1µA during its sleep mode and 650µA when it is taking the measurements. This sensor is very useful for Cranberry's weather box due to its small package, low power consumption, and accurate data measurements.

3.2.2 Barometer - MPL115AT1

To measure pressure, the MPL115AT1 is implemented onto the sensor board. It is a digital barometer that has a range of 50kPa - 115kPa [5]. The package is very small, which is useful for smaller spaces, but it can be difficult when soldering. It has a power consumption of 1µA in sleep mode and 5µA when operating. An external integrated ADC is used to convert the pressure and temperature date to a digital output that is sent to the microcontroller using I2C.

3.2.3 Solar Irradiance - Apogee SP-212

The sensor Cranberry uses to measure solar irradiance is the Apogee SP-212 silicon-cell photodiode that measures shortwave radiation. The irradiance sensor is modeled with a cosine wave that has an error of less than 20Wm^{-2}. The output is between 0-2.5V which is then process by the ADC on the sensor board. The data is adjusted by a calibration factor of 0.5Wm^{-2} per mV to obtain the shortwave radiation in Wm^{-2}. Other weatherbox teams use the

SP-215, but Cranberry is unable to use it due to its output voltage of 0-5V. The current draw of the sensor is 10µA when active.

3.2.4 ADC - ADS1115

Cranberry uses the ADS1115 as its 16-bit analog-to-digital converter, it is low-power that is I2C compatible. The ADC is used by all the sensors that communicate with the microcontroller through I2C. The solar irradiance sensor does not have an internal ADC it uses the external sensor in the board. It can also accept data from more sensors if new sensors are implemented in further semesters. It has a low current draw of 200µA when operating and 0.5µA when in sleep mode.

3.2.5 Real Time Clock - DS3231

The real time clock chosen will be used to timestamp the transmitted data which will make the data we have more organized when the final node network is implemented. The DS3231 sensor has an internal 32kHz timing crystal and temperature sensor. The temperature sensor can detect changed in temperature, and then detect the timing crystal's change in frequency. Cranberry Version 4.0 used the breakout board for the real time clock, which included an external battery slot and a reset, but Version 4.1 uses the DS3231 for reduction is cost and size. The real time clock has uses 0.2mA when active and 0.11mA when idle.

3.3 Power

The power system components are spread across both the main and sensor board. The sensor board includes a solar charger, DC power jack for the solar panel, the battery connector, and hardware switches that control the power going to the sensors and debug LEDs. The two boards use 3.3V, which is provided by a 6V solar panel and a 3.7V lithium ion battery. The panel recharges the battery during operation with the solar charger (MCP73871). The battery voltage then goes to two voltage regulators (MIC5219) that are both on the main board. The voltage regulators regulate the voltage to 3.3V which are for the MCU and the XBee.

The hardware switches (TPS27081A) on the board can control the voltage going to the sensors and the debug LEDs. These switches are used to turn off the voltage going to unnecessary components when the battery voltage is too low which would reduce the overall power draw of the board and give the battery more time to recharge with the solar panel.

3.4 Housing Design

The Cranberry weatherbox uses a 3D printed housing design which is a weatherproof container to protect from water damage. The current housing for Cranberry is designed by SCEL's Housing Team and it is shown in Figure 4, which is below.

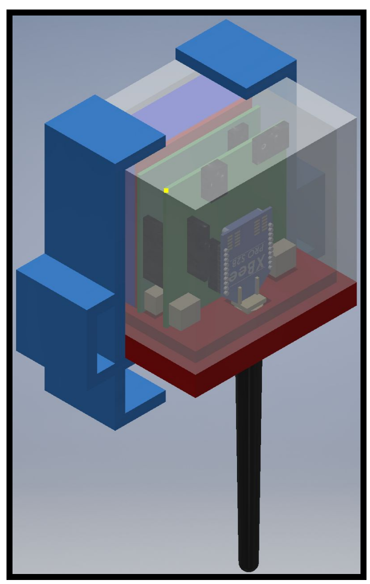


Figure 4: Cranberry Housing

The housing consists of three main pieces, the opaque layer shown in the figure houses the board and the lithium ion battery. The red part of the housing is the lid, and it also has a divider to separate the board from the battery. It also has holes in it for the XBee antenna and for the wire for the solar panel. The blue pieces on the sides of the housing keep it together and also allow clamps to feed through the opening so that it can be secured to the roof.

4. CRANBERRY HARDWARE TROUBLESHOOTING

4.1 Problems and Solutions

Throughout the semester, we ran into a lot of problems when debugging and deploying version 4.1. The first problem we ran into was an issue that was from the previous semester. The voltage regulator that was providing voltage to the microprocessor was heating up and smoking. We figured out that the ADC was soldered upside down. Because of this, we could not use the sensor board that the ADC was soldered on due to the board being shorted. Soldering another sensor board also took up more time than we expected and slightly pushed us back in our Gantt chart.

Another problem we ran into was the solar irradiance crimps. Since we did not have the correct size crimps for the crimp housing, we had to solder the wires directly onto the board. We also had an issue with the solar panel reading. The value of the solar panel voltage was not changing so we talked to the firmware team and they found that the header was not defined in the code. When they updated the code, the solar panel voltage started scaling correctly.

During the first deployment, the XBee stopped transmitting when we were on the roof. We had to take the box down and while debugging, we tested the XBee voltage on the multimeter and found that it was outputting 1.4V instead of 3.3V. We noticed that the XBee header was somehow shorted when we were putting it on the roof. Resoldering a new XBee header on the board fixed the issue.

For the second deployment, the board stayed up for around two days before it stopped transmitting. We noticed that after the two days, the data was transmitting in sporadic intervals such as early morning and late afternoon. We plotted the data that was collected on Matlab and noticed that the battery had not been charging the entire time it was on the roof. When we took the box down and measured the battery voltage, the multimeter read ~3V when the last time the XBee transmitted the voltage was at 1.6V. We tried resoldering the charging chip but while debugging this issue, we found another one. The XBee stops transmitting when the solar panel voltage went above 6V. We tried measuring the XBee voltage regulator and saw that it would drop to ~300mV when the panel voltage went above 6V. We suspect this was the reason why the XBee wouldn't transmit during noon due to the sun providing too much voltage to the panel. Some issues that were brought up were the wrong regulator part, bad charging chip, and burnt out board. We tried a multitude of methods to try and debug this issue. We soldered on a new voltage regulator, replaced the solar panel, resoldered the charging chip again, soldered a whole new sensor board and we still have not found the issue. This problem will be carried to the next semester and will hopefully be debugged.

4.2 Debugging Tools

A tool that was very useful for this semester was the multimeter. There are multiple test points sprinkled throughout the board so testing voltages with the multimeter were made easier. The main issues we had were also found through the multimeter. This tool was also used for continuity tests during soldering or when there were connections that were not soldered on properly.

Another very useful tool that we used in the lab was Atom. Atom is an open source text editor that includes a compiler for the weatherboxes. The firmware team uses PlatformIO to compile and build the code. This software also provided a serial monitor that displayed all of the data that is collected from the weatherbox. Atom was used when we needed to test for the values for the solar panel and battery, and for programming the board initially.

A new tool that was not previously used in other semesters was Matlab. Matlab was used to generate graphs of the data that the module collected while it was on the roof for two days. These graphs showed us that the battery voltage had not been charging, and that the solar irradiance sensor was reading properly if the operating voltage was not at 3.3V. Matlab was very important in our debugging process, as Microsoft Excel could not plot more 250 data points.

4.3 Cranberry 4.2 Redesign

This semester, the team started redesigning for Cranberry v4.2. Some ideas that we have in mind for version 4.2 are to decrease board size, add more test points, remove the external temperature sensor, rearrange mounting holes so that they are at the corners of the board, and potentially fix issues with version 4.1. Because we have become more comfortable with soldering and board design, we would like to bring Cranberry back to its original size of 2” x 2”. We would also like to add more test points on the board. Debugging was a little difficult due to having to reach in between the two boards with the multimeter. Since we do not use the external temperature sensor, we would like to remove it since it takes up a bit of room. We will also be shrinking the board size so removing it would make routing a little easier. Currently on the board, the mounting holes were added after the board was routed so they are not aligned. We would like to align and have them on the four corners of the board. An all-in-one sensor was also currently being looked into for ease of use. This sensor, the BME280, is being used by the newer iterations of the weatherboxes such as Guava and Lilikoi. Because Cranberry has working sensors that have been tested for over two years with stable firmware, the switch over to the BME280 will likely not be made.

5. CRANBERRY WEATHERBOX DOCUMENTATION

Documentation is one of the most important pieces of information that is needed to complete this project. Previous iterations of Cranberry heavily emphasized good documentation, so there is a lot of important information provided. In order to move Cranberry forward, documentation about the design, problems, and future work is updated and uploaded onto the wiki. This information includes schematic and PCB layout for Cranberry 4.1 as well as the bill of materials, power budget, and Eagle parts library.

One of the major design improvements for Cranberry is its significant increase in power efficiency and battery runtime. The team recently did a battery test to see how long the board would last and it went for two days before dying. Below in Table 1 is the theoretical power budget and power consumption for all of the major Cranberry 4.1 components.

Table 1: Theoretical Power Budget for Cranberry 4.1

Cranberry Version 4.1 Power Budget			
3.3 Volt Module	**Datasheet Values**		
Device Name	**Idle (mA)**	**Typical Current Draw (mA)**	**Max Current Draw (mA)**
XBee Transmit	15.00	205.00	220.00
Barometer	0.01	0.01	0.01
Humidity (HIH6031)	0.00	0.65	1.00
V. Reg 3.3V (Main)		0.35	0.90
V. Reg 3.3V (Xbee)		0.35	0.90
Atmega 328P MCU	0.70	1.70	2.70
Irradiance ADC	0.01	0.15	0.30
Irradiance Op Amp		0.80	2.20
Adafruit GPS (MTK3339)		20	20
RTC (DS3231)	0.11		0.2
Total Current Draw (mA)	15.83	229.01	248.21
Supply Voltage (V)	3.30	3.30	3.30
Total Power Consumption (mW)	**52.23**	**755.72**	**819.08**

The current theoretical power budget is made with assumptions that sensors are polling every thirty seconds and otherwise remain in idle. Leakage current from the Xbee module is negligible, since the Xbee is solely used as a transmitter, and that 80% of the battery energy is usable before the battery cuts to protect from under-voltage. The total power consumption is measured to be roughly 756mW. The theoretical power budget was made to estimate how long the board is able to last without the battery being recharged but because we have run a few battery tests in the past, the power budget will not be needed unless another device is added to consume more power.

Data that was collected during the second deployment while the board was still up was exported to a CSV file. Using Matlab, we generated graphs that plotted the data vs. time. The data that we plotted was used for debugging, such as solar irradiance, solar panel voltage, and battery voltage. The three graphs that were plotted are shown below in Figures 5, 6, and 7, respectively.

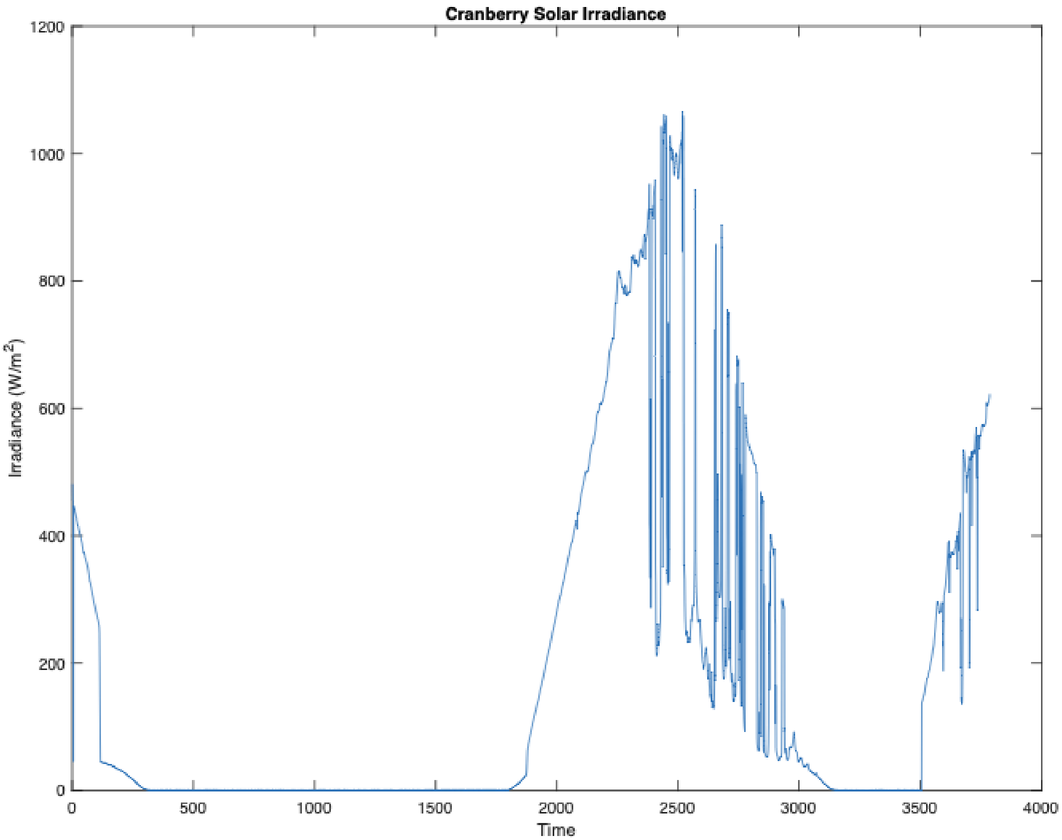


Figure 5: Solar Irradiance vs. Time

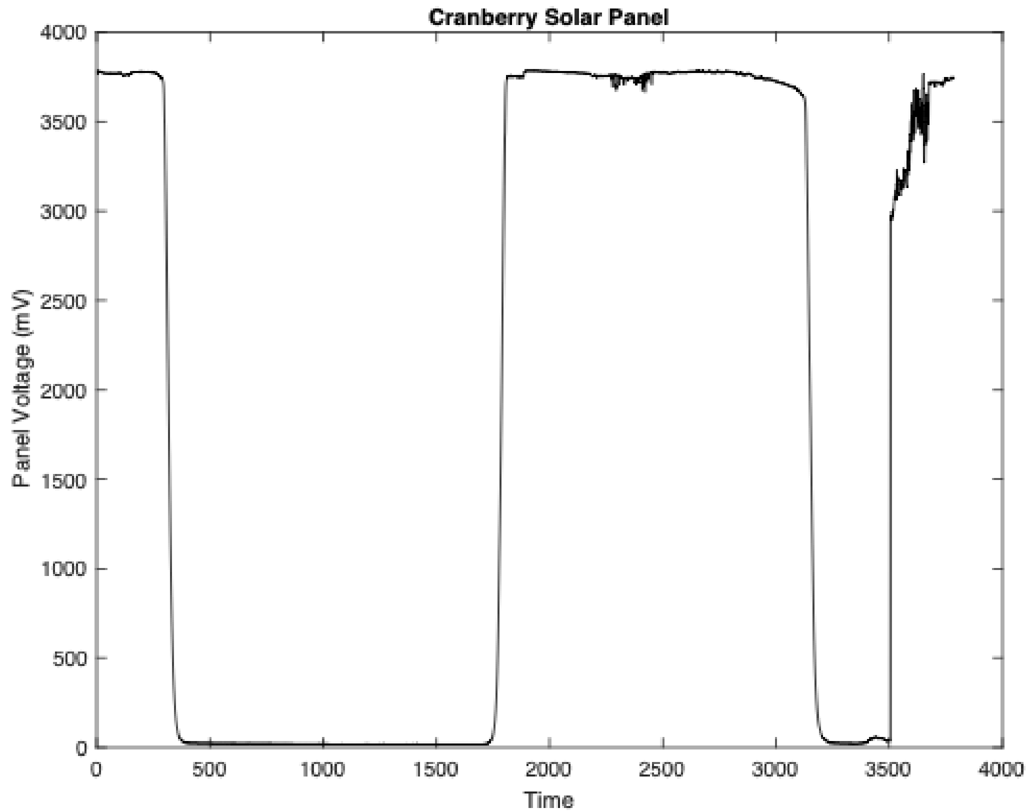


Figure 6: Solar Panel Voltage vs. Time

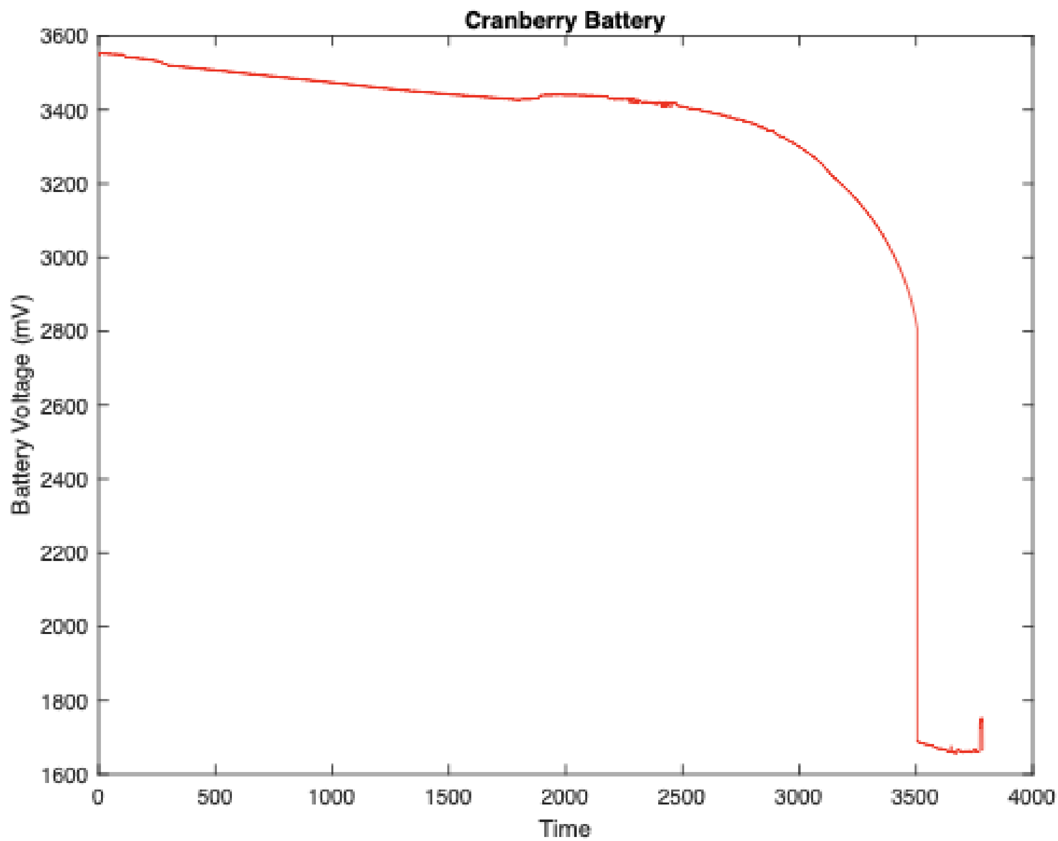


Figure 7: Battery Voltage vs. Time

The bill of materials (BOM) is a complete list of components and hardware used in the Cranberry design. Information about unit cost, quantity, and part description is included with each component used. The BOM includes all of the components that are currently on Cranberry 4.1, excluding PCB manufacturing costs and ABS filament for the 3D printer with the total cost being about $545.89. The BOM is listed below in Table 2.

Table 2: Bill of Materials (BOM) for Cranberry v4.1

#	Part Name	Unit Cost	Quantity	Sub-Total
1	Solar Irradiance Sensor	$235.00	1	$235.00
2	PCB Manufacturing Costs	$30.00	2	$60.00
3	6V Solar Panel	$59.00	1	$59.00
4	Solar Irradiance Leveling Plate	$35.00	1	$35.00
5	3.7V Lithium Ion battery	$29.00	1	$29.50
6	XBee Pro S2B	$29.00	1	$29.00
7	Humidity Sensor	$15.13	1	$15.13
8	Polarized 470 uF Decoupling Capacitors	$2.26	5	$11.30
9	External Temperature Sensor	$9.95	1	$9.95
10	Solar Irradiance ADC	$6.51	1	$6.51
11	Barometer Sensor	$5.10	1	$5.10
12	Status and Debugging LEDs	$0.38	12	$4.55
13	ATMEGA328P MCU	$3.70	1	$3.70
14	XBee Pin Headers	$1.48	2	$2.96
15	Polarized 2.2uF Decoupling Capacitors	$0.69	4	$2.76
16	Mechanical Sliding Switches	$1.37	2	$2.74
17	Miscellaneous Discrete Components			$17.74
18	Adafruit Ultimate GPS Breakout	$15.95	1	$15.95
19	RTC Module	$14.95	1	$14.95
	Cranberry v4.0 Total Cost			**$545.89**

6. INTER-TEAM COLLABORATION

Inter-team collaboration is very important during debugging process. Most of the collaboration was done with the first generation stable weatherbox, also known as Team Apple, the firmware team, and the relay box team, also known as Team Bumblebee. Because both us and Apple were having a lot of issues with the battery charging and XBee transmitting, we would ask each other for assistance during our lab hours. Apple helped us debug the XBee transmitting issue where the XBee voltage was outputting 1.4V. They also had a few suggestions for the current issue of the XBee being unable to transmit due to solar panel voltage being above 6V. The firmware team assisted us with our solar panel voltage by declaring the header in the code. They also helped us look at the XBee firmware when we were not transmitting. During the first deployment, our box was to transmit to Bumblee who would then transmit our data packet to the gateway. At the time, we needed to collaborate with Bumblebee to get our boxes to work together.

7. FUTURE WORK

For the whole semester, Cranberry has been through heavy debugging and will continue to debug version 4.1 in Spring 2019 until it is ready to be deployed. In the meantime, Cranberry version 4.2 will be designed with the specifications listed in Part 4.3.

By Fall 2019, Cranberry is hoped to have designed, debugged and deployed version 4.2, and mass deploy this latest version across campus. Cranberry 3.5 was initially a stable weatherbox that worked but due to housing issues, it was not able to stay up for very long. With the current housing team, a new iteration of Cranberry would be able to stay on the roof longer. Because of Cranberry's stability, there are no immediate improvements that can be made, and we would like to finalize this project with version 4.2 and have this be mass deployed across campus for data collection.

8. CONCLUSION

The main objective of this project was to improve generations of Cranberry by increasing functionality and reliability. The final goal was to deploy a Cranberry board to the roof of Holmes Hall by the end of the semester. Due to unforeseen circumstances, Cranberry version 4.1 has been deployed twice and failed twice. During the second deployment, the box was on the roof for two days but could not stay up due to the battery voltage not being able to charge. Lots of debugging was made but eventually, Cranberry could not be deployed to the roof this semester. This issue was not something that was expected due to the solar panel voltage schematic being the exact same as the previous generation 4.0 and working. Currently, Cranberry version 4.1 is still undergoing heavy debugging and will hopefully be deployed to the roof of Holmes Hall in Spring 2019. A new generation is also currently being designed, with it be the final iteration of Cranberry, which would be mass deployed across the campus.

9. REFERENCES

1. "Hawaii Energy Facts & Figures" State of Hawaii Department of Business, Economic Development and Tourism., May 2014, <http://energy.hawaii.gov/wp-content/uploads/2011/08/HI-Energy-FactsFigures_May2014_2r.pdf>
2. "Atmel ATMEGA328/P" Datasheet, Atmel Corporation, revised Nov. 2015, <http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>
3. "XBee / XBee-PRO ZigBee RF Modules" User Guide, Digi International Inc., Apr. 2008, revised Mar. 2015, <http://ftp1.digi.com/support/documentation/90000976_W.pdf>
4. "Honeywell HumidIcon Digital Humidity/Temperature Sensors" Datasheet, Honeywell Sensing and Control, revised May 2015, <http://sensing.honeywell.com/honeywell-sensing-humidicon-hih6100-series-product-sheet-009059-6-en.pdf?name=HIH6131-021-001>
5. "Miniature I2C Digital Barometer" Datasheet, NXP Semiconductors, revised Feb. 2013, <http://cache.nxp.com/files/sensors/doc/data_sheet/MPL115A2.pdf>