Final Paper for Spring 2017 EE 396 2nd Generation Relay Module: Team Bumblebee

Authors: Kayla Amano, Isaiah Aribal

Date: May 3, 2017

Abstract: The objective of this project was to design a 2nd generation communications module to relay meteorological data collected by other weatherboxes. Its main purpose is to effectively increase the range of the weatherboxes.

1 Introduction

The Smart Campus Energy Lab (SCEL) is one of many research laboratories within the Center for Renewable Energy and Island Sustainability (REIS). The objective of the lab is to develop technologies and practices to promote sustainability and renewable energy usage. Currently, the main project of the lab is the development of low-cost and reliable environmental sensor modules meant to collect meteorological data, such as temperature, humidity, and solar irradiance. These sensor modules are meant to be placed on rooftops across the University of Hawaii at Manoa campus. Team Bumblebee's objective was not to create another weatherbox module, but rather to create a relay to extend the range of our current weatherbox modules.

2 Bumblebee Relay Module

Started in Spring 2017, the Bumblebee weatherbox is a second generation communications module designed to relay meteorological data collected by the other weatherboxes. The first generation relay module, Ant, was started in the Fall 2016 semester and was based off of the generation Apple weatherbox. According to the datasheet, the Xbee Pro S2B has an outdoor line of sight range of 2 miles. However, testing has proven that packets can be dropped at a much closer range. The need to increase the range of the network was the motivation for the development of a relay module. The main goal of team Bumblebee was to reimagine Ant to be compatible with the generation Cranberry weatherbox. Other short term goals for the semester included designing and fabricating a circuit board, doing Xbee field tests, and creating a working relay module.

2.1 Design

The design of Bumblebee was inspired by the design of Cranberry. Unlike Ant, which used an Arduino Uno board, Bumblebee only uses an Atmega 328P as its microcontroller. This will help to reduce the size of the board and weatherbox as a whole. What differentiates Bumblebee from Cranberry and the other weatherboxes is that Bumblebee does not contain any sensors or collect any data. Bumblebee's only purpose is to relay data, effectively increasing the range of the other weatherboxes. Because of this, the only major components that the Bumblebee board has are the microcontroller, the Atmega 328P, and the Xbee Pro S2B. A complete bill of materials is not available at this time because all the passive components and other parts have not been finalized.

2.2 Block Diagram

The block diagrams describe the overall design of both the power system and the communications of Bumblebee. Figure 1 below shows how the different weatherbox modules will communicate wirelessly.

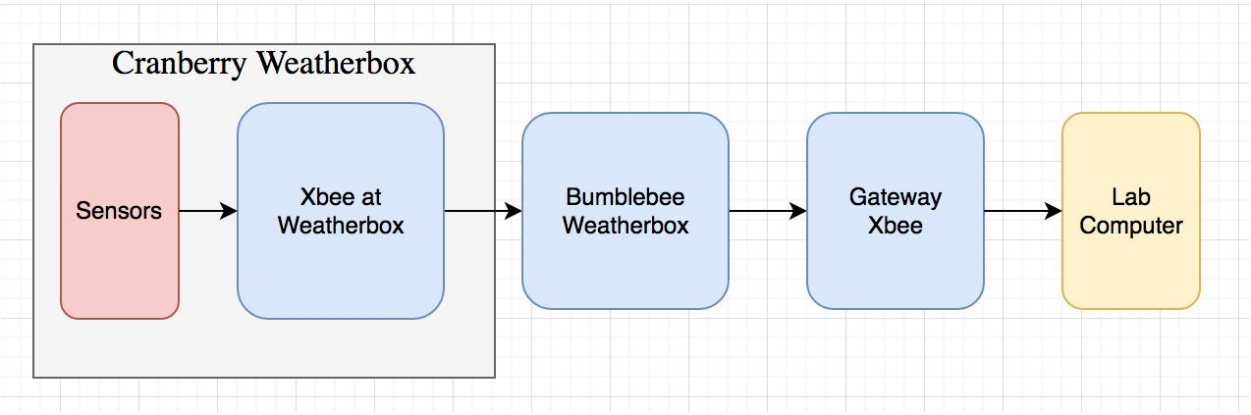

Figure 1 Bumblebee Signal/Communication Block Diagram

The signal/communication block diagram describes the basic path that data will travel between the weatherboxes. For example, Cranberry will collect data from its sensors, construct a

packet with the data, and then send the packet to the Bumblebee relay module. Bumblebee will then receive the packet and forward it to the gateway Xbee, which will then be sent to the lab computer. The above block diagram is simplified as it is possible that in order to reach the gateway xbee the packet will need to be passed along by multiple Bumblebee weatherboxes.

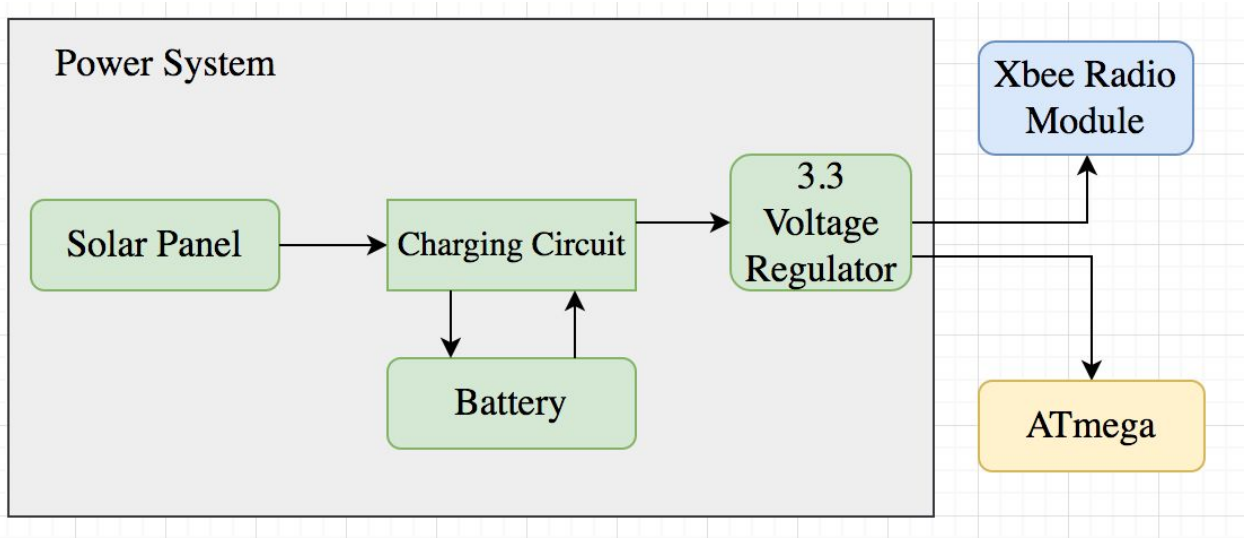

Figure 2 Bumblebee Power Block Diagram

The block diagram for the power system can be seen in Figure 2 above. The power block diagram describes how all of the hardware components are connected. Along with the circuit board the Bumblebee weatherbox would incorporate a solar panel and charging circuit to help supply power to the board. Since Bumblebee does not use any sensors and the Atmega and Xbee can operate at 3.3V, only one 3.3 voltage regulator is required. Doing so decreases the power consumption and the amount of components needed for this weatherbox. At this time the prototype of Bumblebee is on a breadboard and power is received from a laptop.

2.3 Bare Arduino Bumblebee

Figure 3 shows the circuit for the Bumblebee weatherbox. This circuit is referred to as the bare arduino because it does not include the complete arduino board, but only the Atmega 328P

microprocessor and a few necessary passive components. As of right now there is no schematic or PCB design for the board, but future plans include creating one. Without the sensors the bumblebee box is very simple and does not have very many parts or connections.

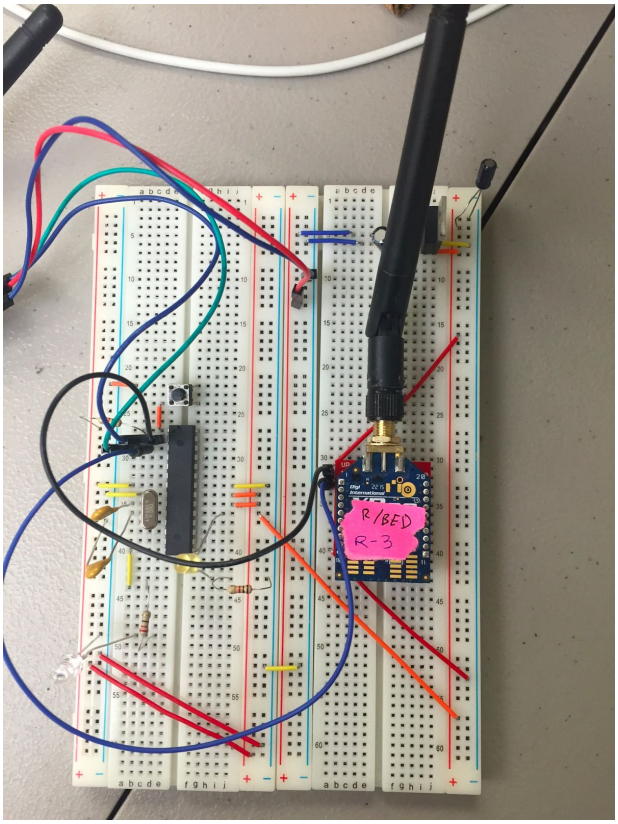

Figure 3 Bumblebee Bare Arduino on Breadboard

2.4 Packet Relay Testing

When testing the Xbees' ability to send and receive packets the programs XCTU and Arduino IDE were used. XCTU was used to configure the Xbees, manage the network, and conduct communication testing. Xbees could be configured in either API or AT mode and as either a coordinator, router, or end device. For the packet testing the Xbees were set in API mode with one Xbee set as a coordinator and the other two set as routers. XCTU is able to generate API frames to send and is then able to decode the frame on the receiving end. Where the packet

is sent from and the contents can be seen in the XCTU console. In order to communicate with each other the Xbees had to have the same PAN ID. While testing the Xbees ability to send and receive when connected to a laptop and XCTU it was found that if the coordinator tries to send a frame only it will receive its own frame.

As a second step, communication between an Xbee attached to an Arduino board and an Xbee connected to XCTU was tested. In order to program the Xbee a program had to be written and uploaded to the Arduino. There is an Xbee library by Andrew Rapp for the Arduino IDE that can be found online that has all the functions needed to write, send, receive, and read packets. Two separate programs were written, one for sending a packet and one for receiving a packet and printing the packet to the serial monitor. The Xbee could be programmed to either send a packet to a specific Xbee or simply broadcast to all the Xbees, however the focus for these tests was direct communication. The test packet sent in initial tests was a simple string, “Hi.” Numbers and other special letters could also be included in the packet. Throughout testing the maximum size of the packet was never reached. The Xbee connected to the Arduino could also receive packets sent by the Xbee connected to XCTU. The contents of the received packet could then be viewed on the serial monitor of Arduino.

Next, an Xbee connected to the bare Arduino board and an Xbee connected to XCTU was tested. Packets could be sent and received both ways. Then, a third Xbee connected to an Arduino board was added to the network. The Xbee connected to the Arduino would send a packet to the bare Arduino Xbee, which would then relay the packet to the XCTU Xbee. In order to relay the code of the relay Xbee was modified to have both the receiving and sending code. Also, the code required the received packet to be copied into a new packet to be sent out.

After the success of these tests it was attempted to use the gateway simulation and a weatherbox test packet to test the relay's ability to relay an actual packet. Up until this point only string packets were being used, however actual weatherbox packets are set up as a struct with multiple variables inside. Because there were no sensors to generate data for the packet, values were hardcoded. From testing is was shown that the code used to forward a simple string packet could also forward a test weatherbox packet.

2.5 Range (Field) Testing

Some Xbee range testing was done throughout the semester. This was not top priority at first; however, due to some design problems focus was shifted to range testing. According to the datasheet the Xbee Pro S2B as seen in Figure 3 below should have an outdoor RF line-of-sight range of up to 2 miles, and a indoor/urban range of up to 300 ft.

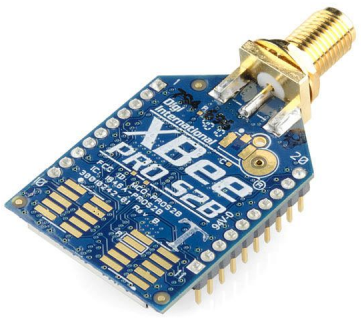

Figure 4 Xbee Pro S2B

Range testing was conducted, taking into account as many variables as possible. One of the first tests conducted was straight line-of-sight test. This was done on the 4th floor of Holmes Hall with one Xbee kept at one end of the building, while the other was moved away in increments of 30 ft all the while keeping line-of-sight. The built-in range test on the XCTU

software was used for this testing. When running these test, the data values looked at included: Local Strength, Remote Strength, Packets sent, Packets sent, Packets received, TX errors, Packets lost, and Percentage of packets received. Ideally, the signal strengths should have been around -32 dBm, no TX errors, and 100% packets received. For the line-of-sight test, the results showed that at 390 ft the signal strengths were around -63 dBm and -65 dBm with 100% packets received. Figure 5 is a chart showing distance vs the received signal strength.

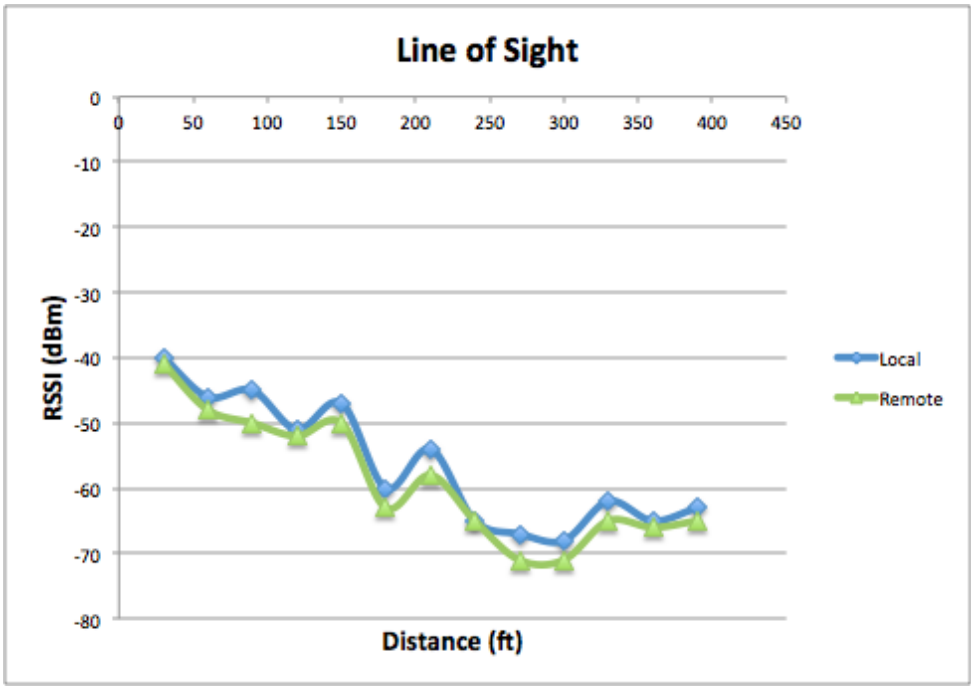

Figure 5 Distance vs. RSSI for Line of Sight

The next variable tested was not line-of-sight (through obstacles) and was also conducted on the 4th floor of Holmes Hall. Using the same setup, one Xbee was kept at one end of the building while the other moved into the next hallway being sure to stay out of sight. Immediately, at the 72 ft mark 6 packets were lost and Tx errors began to occur. As the distance between the two increased the number of Tx errors also increased until all packets were unable to be sent. Also at around the same distances the signal strengths were significantly lower.

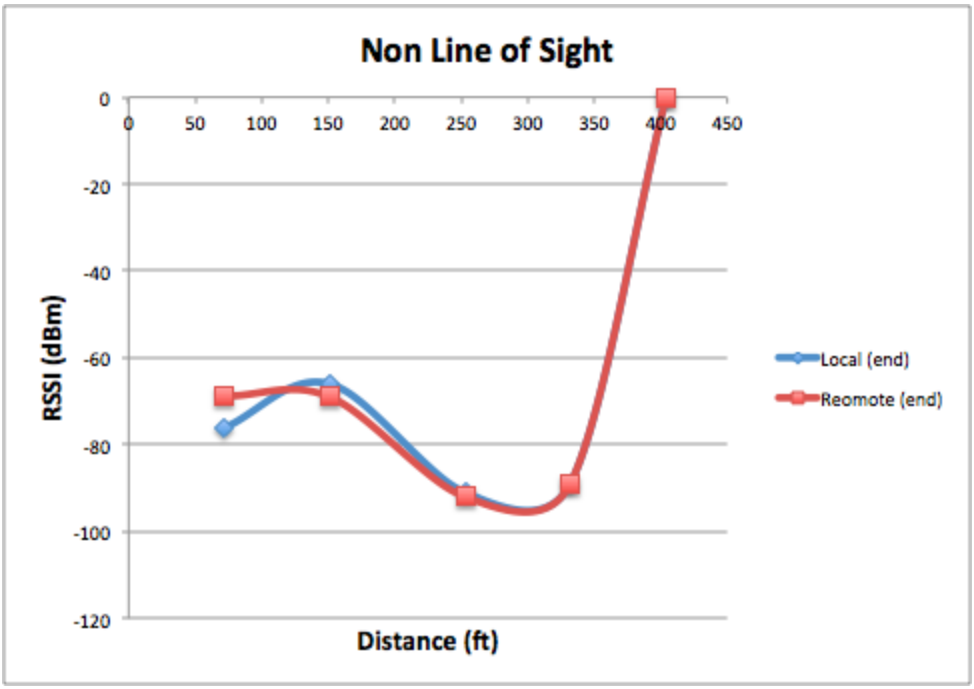

Figure 6 Distance vs RSSI for Non Line of Sight

Also not line of sight, testing between the floors of Holmes Hall was conducted. From the fourth floor to the first floor, which is approximately 56 ft, there was 100% packets received and the signal strength was varying between -77 dBm and -66 dBm. As you can see, the signal strengths are a little weaker than the line-of-sight values, but we were still able to receive 100% of the packets.

There are still other useful variables to test; one of which include the effects of different weather conditions. Being in Hawaii, rain can be very prominent at times and it would be good to know if rain would affect the range or strength of the RF signals between the Xbees. All of the range testing data we gathered can be found in Table 1 below.

Table 1: Xbee Range Test

	Signal Strength				Packets						
Distance (ft)	Local		Remote		Sent	Received	Tx Errors	Packets Lost	Percentage	Other Variables:	Date
30	-40		-41		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Not consistent sig	4/4/17
60	-46		-48		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
90	-45		-50		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
120	-51		-52		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
150	-47		-50		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
180	-60		-63		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
210	-54		-58		25	24	0	1	96%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
240	-65		-65		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
270	-67		-71		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
300	-68		-71		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
330	-62		-65		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
360	-65		-66		25	24	0	1	96%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
390	-63		-65		25	25	0	0	100%	Outside. Holmes hall 4th floor. Weather clear, windy. Line of sight	4/4/17
Through Walls											
	Signal Strength				Packets						
Distance (ft)	Local (start)	Local (end)	Remote (start)	Reomote (end)	Sent	Received	Tx Errors	Packets Lost	Percentage	Other Variables:	Date
72	-72	-76	-75	-69	25	18	1	6	72%	Not line sight. Through building	4/6/17
151	-72	-66	-74	-69	25	13	0	12	52%		
253	-90	-91	-90	-92	25	8	17	0	32%		
332	-89	-89	-89	-89	25	2	23	0	8%		
404	0	0	0	0	25	0	25	0	0%		
Through foliage											
Distance (ft)	Local (start)	Local (end)	Remote (start)	Remote (end)	Sent	Received	Tx Errors	Packets Lost	Percentage	Other Variables:	Date
64	-45	-58	-46	-60	25	25	0	0	100%	Through foliage (by IEEE)	
Floors											
Distance (ft)	Local (start)	Local (end)	Remote (start)	Remote (end)	Sent	Received	Tx Errors	Packets Lost	Percentage	Other Variables:	Date
14	-61	-68	-63	-69	25	25	0	0	100%	4th to 3rd floor	
28	-75	-77	-76	-78	25	25	0	0	100%	4th to 2nd floor	
56	-76	-63	-77	-66	25	25	0	0	100%	4th to ground floor	

3 Problems Encountered and Solutions

The earliest problem encountered was when trying to upload programs to the microcontroller. Because the microcontroller was not taken from an Arduino, and therefore had to be bootloaded, the problem was originally thought to be the bootloader. However, from research it was discovered that the microcontroller needs to be reset during the uploading process to successfully upload. The Arduino board does this automatically each time a program is uploaded. In order to replicate this on the bare arduino the reset button had to be pushed after uploading began. Resetting at the correct time took trial and error and having verbose mode on helped to get the timing correct. After the Preliminary Design Review another team who had also had this problem suggested using a capacitor in place of the reset button to automatically reset the microcontroller. This solution worked for the first few attempts at uploading, but upon further testing proved to be inconsistent. The reset button was replaced and has continued to be used when uploading.

The microcontroller was originally configured to run on 5V with an external 16MHz clock. However, because the Xbee runs on 3.3V, multiple voltage regulators and a 5V boost converter would have been necessary to supply both components with the appropriate voltages. Based on the design of Cranberry it was known that the microcontroller could run on 3.3V, but the microcontroller needed to be reconfigured. In an attempt to run at 3.3V and eliminate the need for the external clock the microcontroller was bootloaded with a program that was supposed to allow both. Although the microcontroller was bootloaded successfully, no program could be uploaded to the microcontroller thereafter. The problem was determined to be the bootloader program. After asking for help it was recommended to not use the internal clock and instead use an 8MHz external clock and the Arduino Pro (3.3V and 8MHz) bootloader. This was successful. Because it was important for the Atmega to run on 3.3V the design of the PCB was held off until this problem was resolved. Ultimately, it took so long to resolve this problem that focus for the remaining semester was turned to range and relay testing.

Progress continued and it was possible to send simple string packets to the Xbee connected to the bare arduino and have it forward the packets to another Xbee. When it was attempted to relay weatherbox test packets, the relay Xbee, which was connected to the bare arduino circuit, was unable to correctly receive or read the packets and could therefore not forward them. At first the problem was thought to be the code because changes had been made to send and receive the weatherbox test packets. Original versions of the code, which only sent simple string packets, were eventually found and tested. The relay Xbee still did not work. Next, the voltages of the circuit were tested. It was found that the LM3940 3.3 voltage regulator was only outputting 2.7V. Looking at datasheet for the voltage regulator, it was observed that the

capacitor values being used were incorrect. However, the required capacitance values were not available, so the LM3940 was switched out with the LM1086 3.3 voltage regulator. Following this replacement 3.3V was being supplied to the board. Relay tests were also then successful. This also served as a lesson to upload any code to Github, so that all changes to the code could be tracked.

4 Future Work

As this is the first semester that the Bumblebee weatherbox has been in development there is a lot of future work and improvements that can be done. The most important of which will be designing a PCB. Currently, the bare Arduino version of Bumblebee is working properly and with a few improvements would be ready to be turned into a PCB. One addition to the board should be debugging LEDs. Because the board is receiving power from the laptop it is possible to view what is happening on the serial monitor of the Arduino IDE or within the XCTU console. Once Bumblebee is fabricated and not connected to a laptop the debugging LEDs will be important for determining if packets are not being received or sent properly. While pressing the reset button to upload a program to the microcontroller has been effective, it would be better the code could be automatically uploaded. Also, if a solution is found to bootload the microcontroller to work with its internal clock that would eliminate the need for an external clock and reduce the need for another part.

Another future goal for team Bumblebee will be working on creating a larger network mesh for the weatherboxes. As of right now each data collecting weatherboxes send their data packet straight to the gateway Xbee. With the continuation of development of data collecting weatherboxes and the addition of Bumblebee, it will hopefully be possible to expand the

network. This goal also includes potentially allowing data collecting weatherboxes to relay packets and the weatherboxes being able to find the shortest routes to the gateway.

For future range testing longer distances for line of sight should be tested. Also, there are a variety of variables that could be tested. Some variables that have been identified for possible future testing are weather, through different materials, elevation, and location.

5 Conclusion

At the beginning of the semester it was decided that instead of continuing Ant, team Bumblebee would take inspiration from Ant and Cranberry to design a new relay module. The relay module's goal was to effectively extend the range of the weatherboxes. Major problems encountered throughout the semester were problems uploading to the Atmega and supplying the board with 3.3V. As of the end of spring 2017 there is a working prototype Bumblebee that is a bare Arduino constructed on a breadboard. Currently, there is no schematic or PCB design, but this will be the top priority for the next semester. In terms of packet relay ability, the Bumblebee prototype is able to receive a weatherbox packet, read it, repackage it, and send it to the gateway. As of right now the data gathering weatherbox is hardcoded to send directly to the address of the relay, and the relay is hardcoded to directly send to the address of the gateway. It would be good if every weatherbox could find out which path to the gateway they need to take. Also, Bumblebee has only been tested to handle relaying from one source at a time. In the future more research and testing will need to be done to accommodate more weatherbox nodes into the network. Some basic range tests for line of sight and non line of sight was conducted. For line of sight there were no errors for the length of Holmes Hall. However, with just a short distance obstacles caused a large amount of errors and a drop in signal strength. In the future, more

thorough range testing should be conducted. Even other variables not being tested such as time of day and temperature should be recorded.

`

References

1. B. Amano and K.P. Castro. “WIP: Wireless Environmental Sensor Module Generation 3” University of Hawai i ́at Mānoa. Dec. 2015, revised May 2016.
2. Rapp, Andrew “Arduino library for communicating with XBee radios in API mode” Dec. 2016, https://github.com/andrewrapp/xbee-arduino
3. S. Saepoo. “Self Sufficient Routing Module for Mesh Sensor Network” University of Hawai i ́at Mānoa. Dec. 2016.
4. “XBee / XBee-PRO ZigBee RF Modules” User Guide, Digi International Inc., Apr. 2008, revised July 2016, http://www.digi.com/resources/documentation/digidocs/PDFs /90000976.pdf.