

Final Paper for Spring 2021 EE396 2nd

Generation Relay Node: Team Bumblebee

Authors: Brian Griswold, Yin Aye, Thant Thiri



Department of Electrical Engineering

University of Hawaii at Manoa

Renewable Energy and Island Sustainability Program

Smart Campus Energy Lab

7 May, 2021

Abstract: ‘Bumblebee’ is Smart Campus Energy Lab’s (SCEL) 2nd generation communications module designed to relay meteorological data collected from weather boxes on Holmes Hall. Meteorological data that is collected from the weatherboxes on top of Holmes Hall and is sent wirelessly using an XBee device inside the weatherbox to the gateway in Holmes Hall. Due to the limited range of the weatherbox sensor node, ‘Bumblebee’ was developed to extend the range of the sensor node network.

Table of Contents

Introduction	2
Relay Node: Bumblebee Overview	3-14
Block Diagram	3-4
Design: Schematic and PCB Layout	4-9
Bare Arduino Board	9-10
Bootloading	10
Packet Relay Testing	11-12
Range (Field) Testing	12-18
Problems and Solutions	18-19
Future Work	19
Conclusion	19-20
References	21

I. Introduction

In 2008, the state of Hawaii and the Department of Energy decided to collaborate to reduce the dependence on imported fossil fuels. Due to Hawaii being the most fossil fuel-dependent state in the United States, the goal of the Hawaii Clean Energy Initiative is to become 100 percent energy clean by 2045^[1]. Smart Campus Energy Lab (SCEL) was made an effort under the University of Hawaii's Renewable Energy and Island Sustainability (REIS) program to support the goals of the Hawaii Clean Energy Initiative. The REIS program's goal is to help UH Manoa to run on its own microgrid that is powered by 100 percent renewable energy.

SCEL's goal is to help make this microgrid by designing, building, and deploying "weather boxes" that can collect and send meteorological data such as temperature, humidity, and solar irradiance. These weather boxes should be designed and developed to be low-cost so that they may be mass-produced. Meteorological data that is gathered from these weather boxes will provide help in planning for future renewable energy installations and data for forecasting algorithms^[2].

Bumblebee does not include any sensors, so it does not collect any data. The ideal location for Bumblebee boxes would be between distant sensors and the lab gateway. Bumblebee is based on the third generation of sensor nodes, Cranberry, and uses many of the same components including the Atmega328P microcontroller, the Xbee Pro S2C, and is powered by a solar panel and rechargeable battery. The Bumblebee box has a simplified circuit and its own PCB design and housing. Our team also does range testing and networking with Xbees.

II. Relay Node: Bumblebee Overview

Block Diagrams

The figures shown below are the overall designs of Bumblebee's power and communication distribution. Starting with Figure 1, this shows us our power system. The power block diagram describes the functionality of each of the hardware components that are being connected. To help power the Bumblebee relay node, a charging circuit and solar panel are incorporated into the board. Based on Figure 1, we can also see that we only have one 3.3V voltage regulator that is being used to supply enough current to the circuit. This is because there are no other sensors being used or are being connected to bumblebee. A 5V boost converter was implemented last semester to have the Atmega run at 5V with a 16MHz clock. Note that the 5V regulator will be applied to both version 4.0 and 4.1 PCB boards.

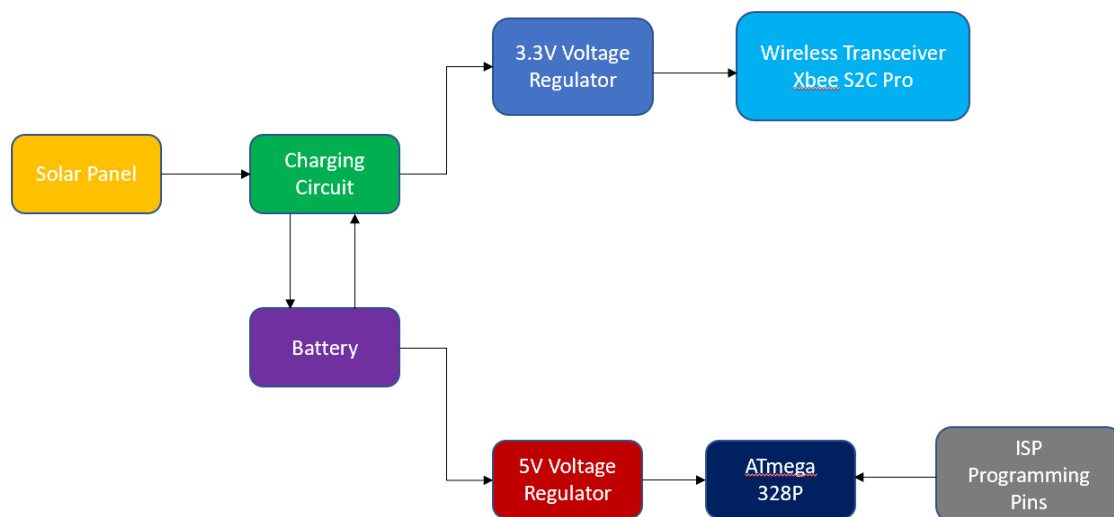


Figure 1: Power Block Diagram

In Figure 2, this shows how the different sensor nodes will communicate wirelessly. The signal block diagram illustrates the basic path of the data travelling from the sensor node and eventually to the gateway computer. We currently use the Xbee S2C Pro instead of the S2B. This block diagram shows that from the weatherboxes that collect data for its sensors, it will construct a data packet and send it to the relay node. Bumblebee will receive that data and send it to the gateway Xbee to be sent to the lab gateway. Overall, this diagram illustrates that data has to go through multiple Bumblebee nodes in order for the lab gateway to receive a packet.

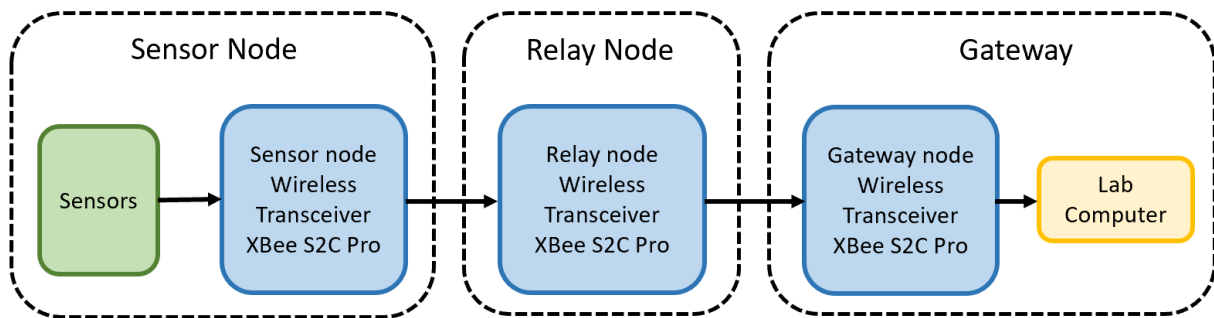


Figure 2: Signal Block Diagram

Design: Schematic and PCB Layout

Version 4.0 had a 5V boost converter implemented in the design to supply 5 volts to the ATmega328P and the 16MHz crystal. This addition was with the intent of increasing the reliability of bootloading and range testing. This semester, the team successfully boot loaded and tested with 3.3 volts and 5.0 volts supplied to the Atmega328P on the version 4.0 board. This was done to see if the added complexity of the 5-volt boost converter improved the reliability of boot loading and more importantly transmission signal strength. More than just increasing the build time, the 5-volt boost converter physically decreased the robustness of the 4.0 design. This proved to carry over to the version 4.1 design. There is still a question as to if the 5-volt boost

regulator is necessary. Bootloading and signal strength appeared to be the same if not better with 3.3 volts supplied to the ATemga328P.

However, since the end product will rely on battery voltage overnight or in inclement weather conditions the boost regulator was decided to be kept since it will supply 5 volts even with a battery drained to as low as 0.7 volts. The hope is this improves the robustness of the mesh network. To be sure a test would need to be performed to see how long only battery voltage will operate an XBee; versus how long an XBee will operate with a boosted battery voltage. The boost converter itself consumes some energy to increase the output voltage although it is specifically designed for high efficiency at greater than or equal to 90%.

When the move to the 5 volt Texas Instruments boosts converter part number TPS61222DCKR was made, the PCB layout did not connect the feedback of the boost converter to voltage out. This proved to be a fatal flaw for the version 4.1 board. Two versions of the regulator exist. A fixed version, which is the boost converter being used; and an externally adjustable regulator. The externally adjustable with R1 and R2 do not need to have FB connected to the regulator's output voltage. The fixed regulator version only mentions the FB needing to have Voltage out connected in a footnote.

We bridged a small section of 0.1mm magnet wire to supply the output voltage to the regulator FB and were able to get 4.7 Volts out with this temporary solution. The team conferred with our 323 circuits professor and stray capacitance from the relatively large magnet wire, comparatively speaking in regards to a PCB trace, and the PCB board design was noted as the most likely issue for the lack of achieving the full 5-volt boost voltage per the design specifications.

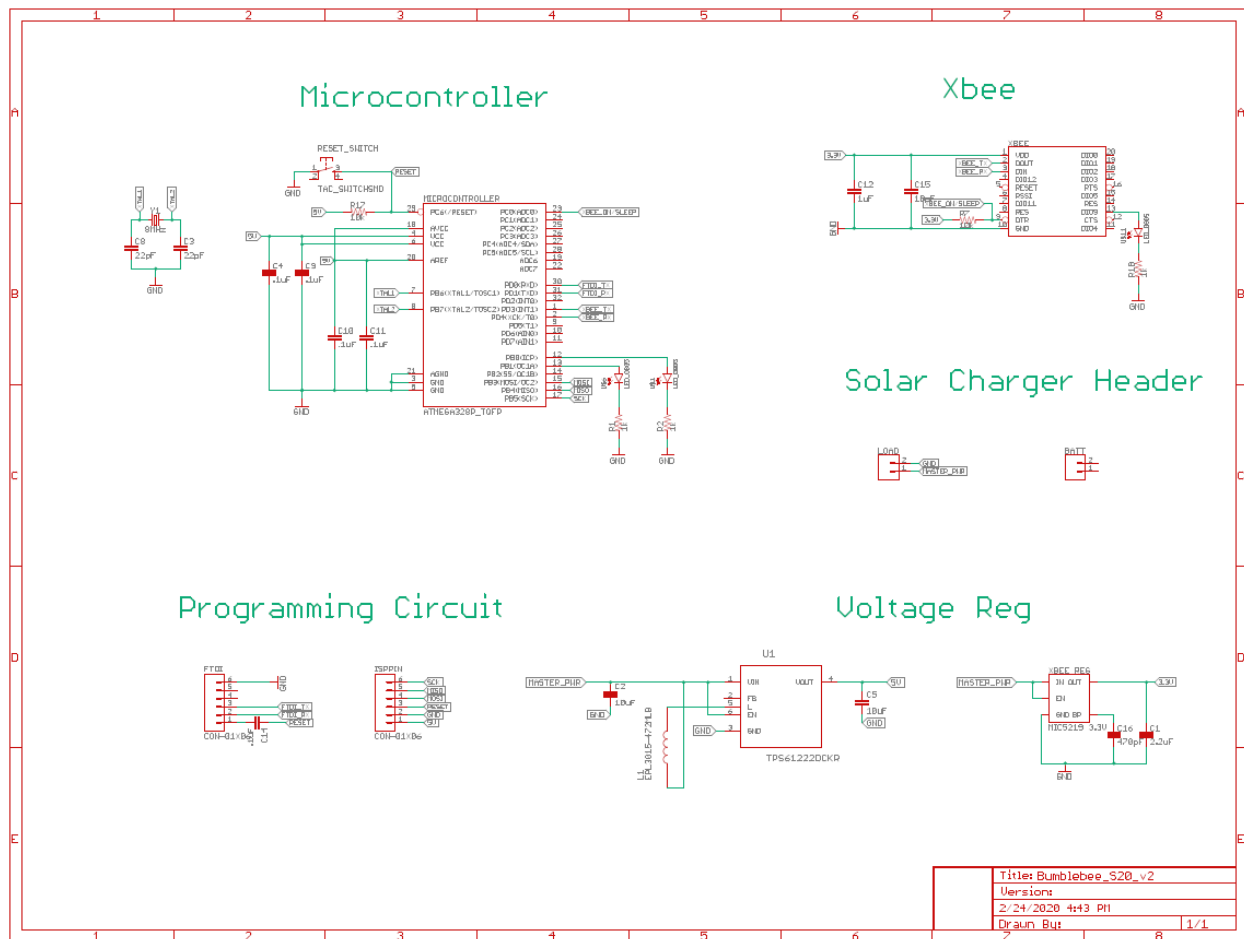


Figure 3: Schematic Layout v4.1

Moving forward FB pin 2 of the TPS61222DCKR needs to be shorted to pin 4 of voltage out. Also, the capacitors are to be moved closer to the input and outputs of the voltage regulator and inductor to minimize the voltage spiking of the inductor propagating through the PCB. Where it is a common design method to have each side of a PCB be a solid copper plate and have one side be positive voltage and the other ground; this is not a good design method for microcontroller circuits as the goal is to isolate circuits from the effects of other components. Version 4.0 manufacturing design actually isolated components better however the board manufacturing process did not produce strong pad anchoring to the board surface. The pads

failed at times while attempting to solder the SMD components. A new version will have only the necessary traces, a better manufacturer will be sourced.

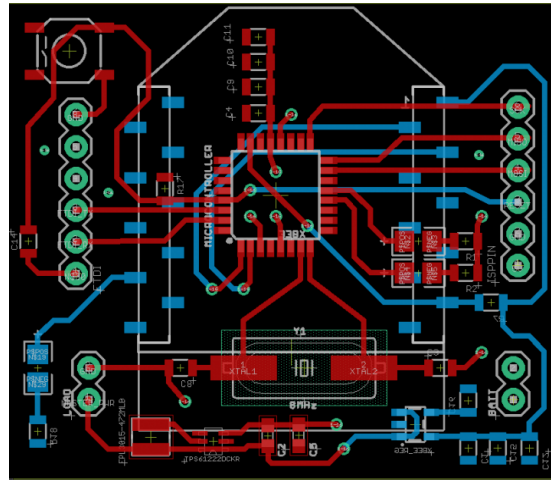
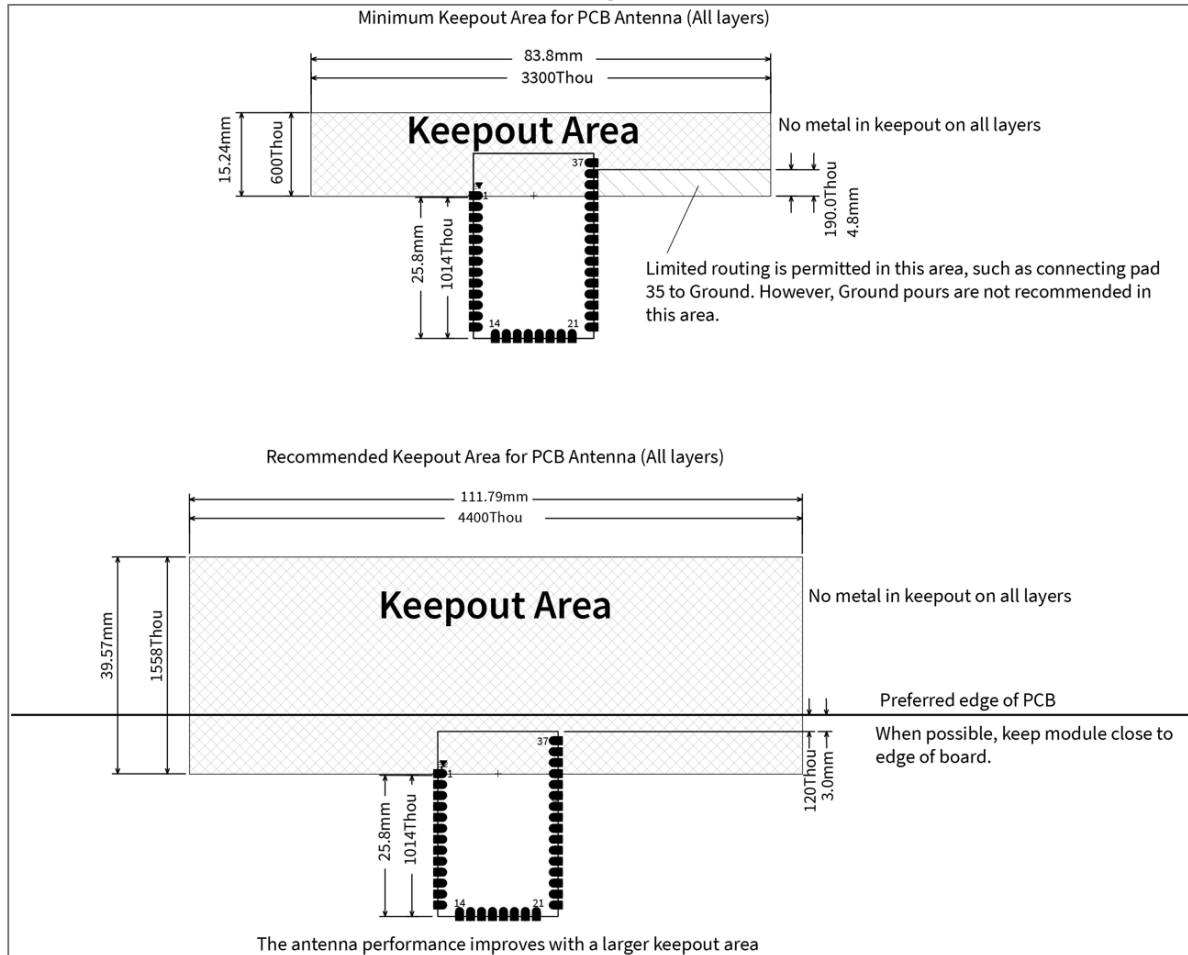


Figure 4: PCB Layout V4.1

A stable power supply with low noise, voltage ripple, is crucial for strong transmission signals. All the preceding design aspects will contribute to the communication signal strength. Power delivery to the Xbee module can further improve this by separating out the capacitors in the voltage supply line to the Xbee module. The capacitors should be placed with the lowest capacitance closest to the Xbee module input voltage and increase in the order of capacitance moving away from the input. Each individual capacitor capacitance filters different frequencies from interfering with the Xbee signal stability. The Xbee module can also be shifted slightly to increase the antenna clearance to prevent the potential for EMF interference while transmitting.

Surface-mount embedded antenna keepout area



Some design changes that are not directly related to the power and signal stability of the PCB but the overall robustness of design are recommended as well. A dedicated battery connector port on the PSB is recommended. Where it will not be needed for the deployed units its addition will mitigate the potential for reversed polarity connections, or the need to power through a solar charger module when field testing with a battery if a loose wire connection is not made. Most importantly it will reduce intermittent connections that become worse with the repeated insertion and removal of wires and components such as the solar chargers into the female headers. The solar charger units themselves will also be protected from wear and tear as they are not physically needed for field range testing. Intermittent connections were frequently

traced back to connectivity/bootloading and signal strength issues. In this capacity, the dedicated battery port will improve signal strength. Stronger female machine pin headers are also recommended and are being sourced and considered that will provide a firmer connection over time with repeated wire or component insertion. Lastly, the reset switch on more than one PCB broke off from accidental drops. The switch is to be moved out of the antenna clearance area. This offers the opportunity when moving the reset to replace it with a smaller two terminals SPST tactile switch that can be placed elsewhere when the switch is moved that is in a more protected area. The reset proved to have limited necessity this semester, bootloading and uploading sketches was done without the physical textile switch. Reset control was given to the programming Arduino via a wired connection. The trade-off for a harder-to-reach but more protected location seems to be reasonable.

Bare Arduino Board

From previous semesters, the team used a bare Arduino board or a bare Bumblebee for relay testing. We started from scratch to build a bare Bumblebee shown in Figure 5. Rebuilding the bare Bumblebee helped us understand the functionality of each component and the relay node itself. This bare Bumblebee was used for range testing and we were able to receive data packets from the coordinator XBee successfully. At the beginning of the semester, there was confusion as to the capacitor in reset being for the target or programmer Arduino. We found out that we do not need to put a capacitor in reset because the XCTU code has been upgraded to fix the problem, reset is no longer pulled low, thus the capacitor is not needed anymore to prevent the programmer from resetting when attempting to bootload a target. Previously it was believed that the capacitor made a reset happen automatically. This is not the case. It merely prevented reset

from pulling low which will mess up a bootload process. As mentioned, the XCTU code has been updated to prevent the programming Arduino from resetting when bootloading.

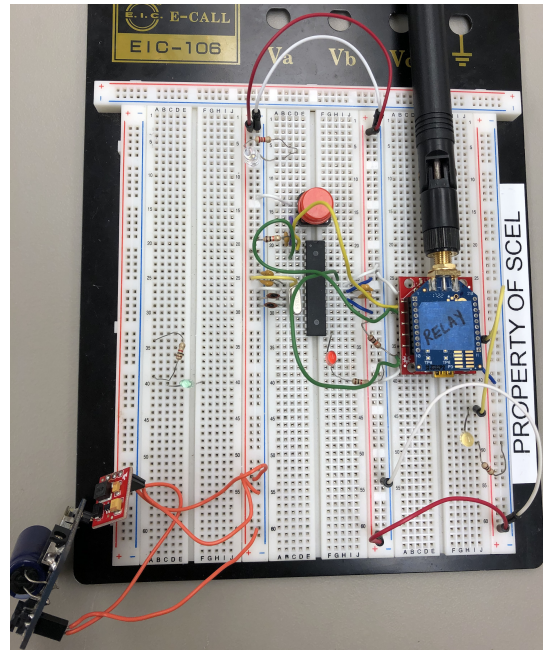


Figure 5: Bumblebee Bare Arduino Board

Bootloading

The main issue that the previous members were having is that they could not bootload the PCB board. They have been using the bare Arduino on a breadboard with an XBee module for range test but we want to use a PCB instead. We started from scratch to learn how to bootload the bare Bumblebee and then we used the PCB board using Arduino Uno to bootload. The first thing we did was upload the ArduinoISP sketch onto our Arduino Uno to make it a programmer to burn the bootloader. Then wire up the Arduino Uno to our target microcontroller on our PCB board. We burned the bootloader after making sure to select all the tools correctly and burned the bootloader. We noticed that we need to remove the microcontroller from the Arduino Uno to upload sketches to an ATmega^[4]. To confirm that a successful bootload and sketch upload was

taken by the Atmega328P a blink test was uploaded to both version 4.0 and 4.1. Although version 4.1 has a fatal flaw previously discussed in the PCB board section that requires a redesign to be able to deploy it, functionality was achieved by supplying 5 volts to its circuit via the Arduino's power supply.

It was decided that further assembly or testing of the 4.1 was not necessary or productive as other than the regulator which had the flaw in the design version 4.0 and 4.1 are the same. For the remainder of the semester, range testing was performed with functional 4.0 PCB boards. A few populated 4.0 boards that had been accidentally damaged and viewed as obsolete thus not repaired initially were repaired by various means and put back in service to perform many to one range testing.

Packet Range Testing

The XCTU and Arduino IDE programs were used to test the abilities of the XBee to transmit and receive packets. The XCTU allows us to set up, configure, and test XBee RF modules easily. Since our main purpose of the project is to have a relay node that helps to send meteorological data to the gateway Xbee, we configured three XBees: one as a coordinator and two as the routers. We can add as many XBee as we want to act as a relay node. The coordinator XBee acts as the gateway or receiving-end. The two routers are used for the sending-end (sensor node) and for the relay node to transmit packets to other XBees. XBees that are configured as routers can transmit packets to other XBees, while coordinators can only receive. XBees are also identified through their address, which can be divided into two parts, serial low and high.

We want to make sure that the configuration for all the XBees is correct and they are able to recognize each other. To confirm the communication of the XBee, we need to plug all the

XBeeS into a computer. We used the DigiMesh 2.4TH PRO for the firmware to do Many-to-One Networking. The APT setting of a coordinator was set to be enabled while the rest of the routers' API settings were left in the transparent mode. We also need to make sure that the PAN ID or the destination address for each XBee is the same because PAN ID allows the sending XBeeS to transmit to another that has the same address.

After configuring the XBeeS, attach the XBee to the PCB board using a battery as a power source and program it using the Arduino IDE program. A program to send data packets in certain time intervals can be written using Andrew Rapp's XBee library for Arduino IDE that can be found online^[3]. When programming the XBee in the PCB board, make sure to change the switch into DLINE on the XBee shield and switch it back after. If not, a sync error could occur while trying to upload the code onto the board. To see whether the sending XBee is transmitting data or there is a communication between the XBeeS, go back to the XCTU console monitor and close the port. Once the port is closed, there should be packets received if all the connections are correct.

Field Range Testing

The purpose of range testing is to take into account as many variables as possible, such as obstacles and weather, and to gather data on how far the XBee can reach certain distances. To conduct this testing, a built-in range test software program in XCTU should be used. The data values include local strength, remote strength, packets sent and received, TX errors, packets lost and percentage of packets received. By determining those data values, we can conclude how the signals would behave with different variables. These variables could be based on the weather and the more wind there is, the more likely the signal will be interrupted. This variable also includes

when students are passing by between the two XBee modules and will result in some disruption in the data.

Set-up

As you can see in Figure 6, the line-of-sight testing provides a clear pathway between the local and remote XBees while increasing the distance in between after each test. We did the same process for the range test while using XBee attached to the v4.0 PCB. With the not line-of-sight testing, there would be obstructions in between the local and remote XBee. As the tests progress, the number of obstructions in between and the distance between would increase.

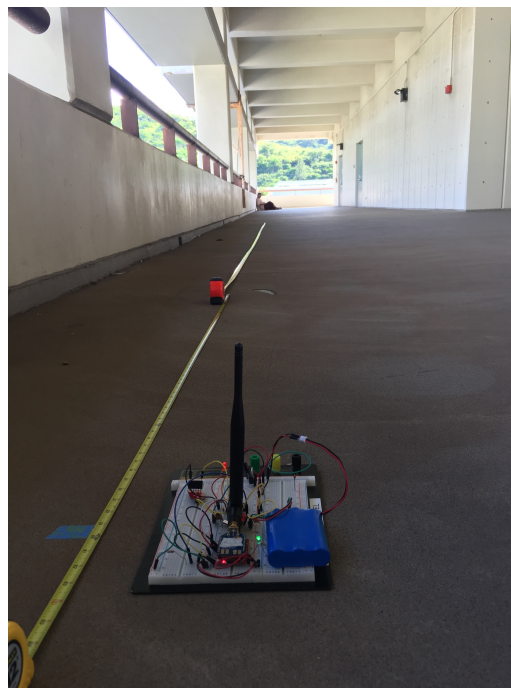


Figure 6: Line of Sight Set-up

Range (Field) Testing Results

We were able to fix our bootloading issues this semester; therefore, we used the XBee S2C Pro and we determined the performance of it by completing range-testing on both our bare

Bumblebee board and v4.0 in three different ways; line-of-sight, through the trees, and through the floors on Holmes Hall. We also did the range testing using DigiMesh as a many-to-one network.

The purpose of range testing is to take into account as many variables as possible and to gather data on how far the XBee can implement certain distances such as obstacles and weather. To conduct this testing, we used a built-in range test software program in XCTU. The data values included were local strength, remote strength, packets sent and received, TX errors, packets lost and percentage of packets received. A good signal strength would be greater than -80 dB/m.

Distance (ft)	signal Strength						Packets				other variables	date
	local	remote	sent	received	Tx Error	Packets Lost	percentage					
LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT
30												
Trial 1	-50	-50	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-48	-49	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-46	-50	15	15	0	0	100	windy,sunny	24-Jan			
60												
Trial 1	-52	-53	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-52	-52	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-55	-52	15	15	0	0	100	windy,sunny	24-Jan			
90												
Trial 1	-61	-61	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-52	-53	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-53	-55	15	15	0	0	100	windy,sunny	24-Jan			
120												
Trial 1	-63	-63	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-60	-59	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-63	-63	15	15	0	0	100	windy,sunny	24-Jan			
150												
Trial 1	-59	-59	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-64	-64	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-64	-68	15	15	0	0	100	windy,sunny	24-Jan			
180												
Trial 1	-53	-52	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-53	-52	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-54	-53	15	15	0	0	100	windy,sunny	24-Jan			
210												
Trial 1	-58	-58	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-60	-58	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-57	-56	15	15	0	0	100	windy,sunny	24-Jan			
240												
Trial 1	-58	-58	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-57	-58	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-61	-62	15	15	0	0	100	windy,sunny	24-Jan			
270												
Trial 1	-58	-59	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-58	-58	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-57	-56	15	15	0	0	100	windy,sunny	24-Jan			
300												
Trial 1	-61	-62	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-63	-63	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-62	-62	15	15	0	0	100	windy,sunny	24-Jan			
330												
Trial 1	-59	-59	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-61	-61	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-58	-58	15	15	0	0	100	windy,sunny	24-Jan			
360												
Trial 1	-62	-62	15	15	0	0	100	windy,sunny	24-Jan			
Trial 2	-62	-63	15	15	0	0	100	windy,sunny	24-Jan			
Trial 3	-59	-59	15	15	0	0	100	windy,sunny	24-Jan			

Figure 7: Line-of-Sight Range Testing Results using Bare Bumblebee

The first range test conducted was the line-of-sight testing completed on Holmes Hall 4th floor, where there were no obstacles between the local signal and remote relay module. We placed the local signal on one end of the building and moved the Bumblebee relay module farther away in increments of 30 feet until we reached the other end of the building. The weather condition of the first range test was sunny and windy. We received all the packets that were sent throughout the entire testing procedure as shown in Figure 7. As expected, the signal strength weakened as the distance increased; however, our dBs were still above -80dB/m. Overall, this range-testing was a success.

Distance (ft)	THROUGH WALLS MCCARTHY MALL										
90											
Trial 1	-52		-50		15	15	0	0	100	sunny	1/31
Trial 2	-64		-56		15	15	0	0	100	sunny	1/31
Trial 3	-56		-56		15	15	0	0	100	sunny	1/31
175											
Trial 1	-66		-66		15	15	0	0	100	sunny	1/31
Trial 2	-63		-64		15	15	0	0	100	sunny	1/31
Trial 3	-64		-68		15	15	0	0	100	sunny	1/31
244											
Trial 1	-61		-63		15	15	0	0	100	sunny	1/31
Trial 2	-63		-64		15	15	0	0	100	sunny	1/31
Trial 3	-65		-67		15	15	0	0	100	sunny	1/31
324											
Trial 1	-73		-75		15	15	0	0	100	sunny	1/31
Trial 2	-74		-74		15	15	0	0	100	sunny	1/31
Trial 3	-73		-75		15	15	0	0	100	sunny	1/31
403											
Trial 1	-80		-81		15	15	0	0	100	sunny	1/31
Trial 2	-73		-74		15	15	0	0	100	sunny	1/31
Trial 3	-74		-74		15	15	0	0	100	sunny	1/31
477											
Trial 1	-84		-84		15	15	0	0	100	sunny	1/31
Trial 2	-81		-83		15	15	0	0	100	sunny	1/31
Trial 3	-86		-86		15	15	0	0	100	sunny	1/31

Figure 8: Not Line-of-Sight using Bare Bumblebee (Through Trees at McCarthy Mall)

The second range test was done through the trees in McCarthy Mall. The set-up of this test was similar to the straight line-of-sight testing. Here, we had the local signal behind the first tree and varied the remote relay module behind each tree along McCarthy Mall, increasing the distance and the number of obstructions. As shown in Figure 8, we were able to receive and send all the packets for all the distances with no package lost in between. However, as we went farther through the distances, the signals became weak. Compared to the previous semester, we did not

get strong signals for farther distances but successfully improved our data results. Due to COVID-19, there were no people who passed by when we did the testing. Again, the results were more accurate than the previous semester because our XBees didn't lose connection.

The third range testing that we did was "line-of-sight" again but, this time we used our v4.0 PCB board with the battery as a power source. Figure 9 shows the results of line-of-sight using version 4.0 PCB board. While doing the range test, the weather was rainy and windy. As we can see, there were some packages lost at 90 ft, 270 ft, and 390 ft. We can also see that the signal strength gets weaker as the distance increases.

Distance (ft)	signal Strength						Packets				other variables	date
	local	remote	sent	received	Tx Error	Packets Lost	percentage					
LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT	LINE OF SIGHT
30	30											
Trial 1	-53	-52	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-52	-56	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-50	-54	15	15	0	0	100	windy,rainy		28-Feb		
60	60											
Trial 1	-64	-61	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-64	-64	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-61	-62	15	15	0	0	100	windy,rainy		28-Feb		
90	90											
Trial 1	-68	-68	15	14	1	1	93.33333333	windy,rainy		28-Feb		
Trial 2	-68	-69	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-70	-71	15	15	0	0	100	windy,rainy		28-Feb		
120	120											
Trial 1	-72	-72	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-71	-72	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-76	-76	15	15	0	0	100	windy,rainy		28-Feb		
150	150											
Trial 1	-63	-63	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-64	-64	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-63	-63	15	15	0	0	100	windy,rainy		28-Feb		
180	180											
Trial 1	-65	-64	15	15	0	0	100	windy,rainy		28-Feb gust of wind weakens signal		
Trial 2	-65	-65	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-65	-64	15	15	0	0	100	windy,rainy		28-Feb		
210	210											
Trial 1	-71	-72	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-72	-74	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-68	-68	15	15	0	0	100	windy,rainy		28-Feb		
240	240											
Trial 1	-71	-71	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-73	-73	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-69	-71	15	15	0	0	100	windy,rainy		28-Feb		
270	270											
Trial 1	-70	-70	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-84	-80	15	13	2	2	86.66666667	windy,rainy		28-Feb		
Trial 3	-88	-88	15	12	3	3	80	windy,rainy		28-Feb		
300	300											
Trial 1	-79	-79	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-78	-79	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-80	-80	15	15	0	0	100	windy,rainy		28-Feb		
330	330											
Trial 1	-75	-73	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-79	-80	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-81	-74	15	15	0	0	100	windy,rainy		28-Feb		
360	360											
Trial 1	-79	-80	15	15	0	0	100	windy,rainy		28-Feb		
Trial 2	-76	-82	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-71	-70	15	15	0	0	100	windy,rainy		28-Feb		
390	390											
Trial 1	-77	-84	15	14	1	1	93.33333333	windy,rainy		28-Feb		
Trial 2	-78	-76	15	15	0	0	100	windy,rainy		28-Feb		
Trial 3	-76	-75	15	15	0	0	100	windy,rainy		28-Feb		

Figure 9: Line-of-Sight Range Testing Results using v4.0 PCB Board

We also range tested through the trees at the McCarthy Mall and floor to floor at Holmes Hall. Figures 10 and 11 show the results of them. There were 3 packages lost at 477 ft and the signal strength was lower than -80 dB. While we were doing the range test, there were a lot of people walking around and we assumed that it was an obstruction. During floor to floor range tests, we figured out that there were some packages lost between the 4th floor and 1st floor. The strength of the signal also gets weaker as we go farther.

Last but not least, we also successfully transmitted the ASCII value to nodes out of direct range with Bumblebee 4.0 as a relay router using DigiMesh. However, we haven't done the official range testing yet.

Distance (ft)	THROUGH WALLS MCCARTHY MALL									
90										
Trial 1	-67	-67	15	15	0	0			100	sunny, windy, people around there
Trial 2	-56	-56	15	15	0	0			100	sunny, windy
Trial 3	-61	-58	15	15	0	0			100	sunny, windy
175										
Trial 1	-64	-59	15	15	0	0			100	sunny, windy
Trial 2	-57	-57	15	15	0	0			100	sunny, windy
Trial 3	-61	-59	15	15	0	0			100	sunny, windy
244										
Trial 1	-65	-67	15	14	1	1			93.33	sunny, windy
Trial 2	-70	-69	15	15	0	0			100	sunny, windy
Trial 3	-77	-65	15	15	0	0			100	sunny, windy
324										
Trial 1	-66	-79	15	15	0	0			100	sunny, windy
Trial 2	-73	-75	15	15	0	0			100	sunny, windy
Trial 3	-68	-71	15	15	0	0			100	sunny, windy
403										
Trial 1	-75	-75	15	15	0	0			100	sunny, windy
Trial 2	-78	-76	15	15	0	0			100	sunny, windy
Trial 3	-83	-84	15	15	0	0			100	sunny, windy
477										
Trial 1	-90	-91	15	12	3	3			80	sunny, windy
Trial 2	-88	-83	15	15	0	0			100	sunny, windy
Trial 3	-88	-79	15	15	0	0			100	sunny, windy

Figure 10: Not Line-of-Sight using v4.0 PCB Board (Through Trees at McCarthy Mall)

Distance (ft)	HOLMES HALL FLOORS								
4th-3rd									
Trial 1	-64		-64		15	15	0	0	100 sunny, windy
Trial 2	-61		-64		15	15	0	0	100 sunny, windy
4th-2nd									
Trial 1	-76		-75		15	15	0	0	100 sunny, windy
Trial 2	-69		-74		15	15	0	0	100 sunny, windy
4th-1st									
Trial 1	-72		-72		15	11	4	4	73.33 sunny, windy
Trial 2	-82		-75		15	15	0	0	100 sunny, windy

Figure 11: Not Line-of-Sight using v4.0 PCB Board (Holmes Hall Floors)

III. Problems and Solutions

The teams this semester were able to get back into the SCEL lab. This enabled a considerable amount of progress to be made. The natural attrition of members of the lab through graduation left mostly new members this semester as with previous semesters. Members who had previous experience gained it during the initial lockdown phase of covid. This left them with a disadvantage over a normal semester of experience.

Boot loading a 4.0 and 4.1 PCB was a cornerstone goal of this semester. Initially, all attempts to bootload both models yielded either sync errors or no communication. The root cause of version 4.0, not boot loading is believed to be intermittent female header pin connections. Ultimately this led to incorrect settings being selected for boot loading in attempts to gain sync. Version 4.1 was not physically powering correctly which was the root cause of the sync error on version 4.1.

At the beginning of the semester communication between teams was not strong. The SCEL is a collaborative effort with each team at some capacity overlapping slightly what other teams are doing. Towards the end of the semester, we were allowed to have one or two members from other teams come into the lab to work together. This helped immensely as it allowed us to help others diagnose issues and further to see how each other's hardware is intended to interact

with Bumblebee. Teams communicated with each other previously via zoom; however, this is not always effective with hardware diagnostic issues.

The various aspects of version 4.1 design issues discussed in the PCB area of this paper were discovered through many hours of painstaking research to diagnose each. When attempting data transfer through Rx and Tx to upload a sketch or to communicate between modules any intermittent poor connection is detrimental. To achieve stable reliable consistent bootload, upload, and transmission results is the reason why improvements were suggested in these areas of the PCB.

A step by step thoroughly documented procedural approach to diagnosing and debugging issues is a major skill set to perfect and take away from the SCEL lab. This will help further SCEL teams the most.

V. Future Work

We are planning to have a new version 4.2 PCB board to fix the issues we are having with version 4.1 which is the voltage regulator. We have made some progress on transmitting the data from one XBee to another using five XBee. We would like to do more range testing using many-to-one networking for further distance. We would also like to improve the connection or signal strength by adding more relay nodes as needed. We also want to do the range test using other hardware team's boards since our v4.0 functions properly.

VI. Conclusion

At the beginning of the semester, our team had a major learning curve since this is our first semester joining the SCEL team. However, Francis and Lauryn were able to teach and help

us to do and gave us all the information we needed to continue what they've done so far. Lauryn also taught us how to solder the PCB board. Due to COVID-19, we weren't able to let the other team members come into the lab. However, in the middle of the semester, some of the team members were able to come in and help us with the bootloading issues.

We made huge progress this semester. Ever since we were able to fix our bootloading issues with our v4.0 PCB, we were able to do the range testing with line-of-sight in Holmes Hall, not line-of-sight in McCarthy Hall, and also in between floors. Initial tests were completed with many to one communication in Digimesh set up. Zigbee tests can also be completed. Which mesh network setup will be best for the SCEL project is an item to discuss between the different teams as well in the future.

References

- [1] Hawaii Clean Energy Initiative. (n.d.). Retrieved May 08, 2020, from <http://www.hawaiicleanenergyinitiative.org/>.
- [2] (n.d.). Retrieved April 20, 2021, from <http://scel-hawaii.org/research/>.
- [3] Rapp, Andrew “Arduino library for communicating with XBee radios in API mode” Dec. 2016, <https://github.com/andrewrapp/XBee-arduino>.
- [4] “From Arduino to a Microcontroller on a Breadboard,” Arduino, 05-Feb-2018, [Online]. Available: <https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoToBreadboard>
- [6] “Many-to-one Routing,” DIGI, 29-Aug-2019. [Online]. Available: https://www.digi.com/resources/documentation/Digidocs/90001942-13/concepts/c_many_to_one_routing.htm. [Accessed: 16-May-2020].