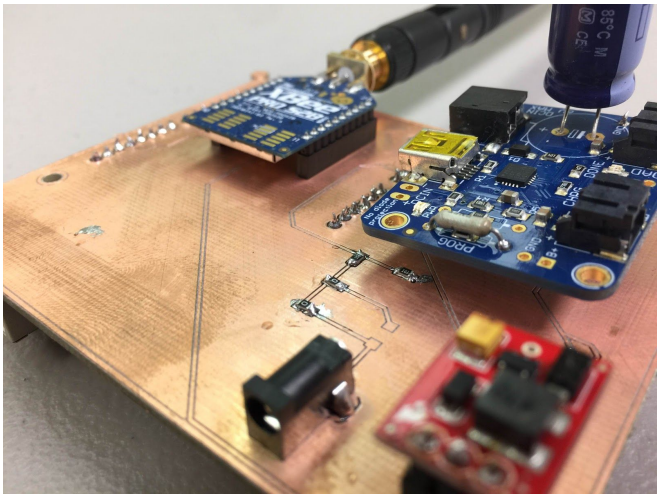
Self Sufficient Routing Module for Mesh Sensor Network (WIP)

EE 496 Final Report

Fall 2016



Smart Campus Energy Lab (SCEL)

Department of Electrical Engineering

University of Hawai'i at Manoa

Author: Savath Saepoo
Advisor: Dr. Anthony Kuh
Mentor: Tryston Fagarang

Words: 4223
Date: 12/13/16

Table of Contents

1 Abstract

A communication module to relay meteorological data collected by other weather boxes developed in the Smart Campus Energy Lab (SCEL) at the University of Hawai'i at Manoa is presented. Team Ant is a subteam of the weatherbox project that is in charge of developing, testing and documenting this communication module. The design referenced Apple's design but with less components, therefore reducing its complexity and size. Communication between the XBees for the prototype was successful and testing have yet to be conducted for the final design. Documentation of past and current work are provided to ensure that future members have the resources to continue this project.

2 Introduction

With the dependencies of oil, electricity, etc increasing each year, a need for renewable energy sources are in high demands. Hawai'i holds the most expensive electrical prices throughout the nation due to imported oil and electrical systems throughout Hawai'i and it is still rising [1]. The University of Hawai'i at Manoa paid approximately $35 million for electricity bill back in 2012 and despite the implementation of energy efficient measures, the price decreased to $34.3 million in 2014 [2]. The reason for such a low drop of prices was due to the fact that the price of electricity per kilowatt hour increased.

The Smart Campus Energy Lab (SCEL) is a subset of the Renewable Energy and Island Sustainability (REIS) program at the University of Hawai'i at Manoa. This lab is led by students that are on the leadership team and the lab as a whole focuses on the development of environmental sensor network. Students can gain a variety of skills that can assist in their

professional career, such as printed circuit board (PCB) design for hardware and programming for software. Life skills such as communication, leadership, presentations, and project management are also developed. SCEL has three projects - Weatherbox, Forecasting and Wind Sensor. The weatherbox project is the main project of the lab, which focuses on the design and development of low cost, accurate and reliable wireless environmental sensor modules. Meteorological data such as temperature, humidity, barometric pressure and solar irradiance are collected by these sensor modules and will assist in determining the optimal places to install renewable energy sources on the University's campus. Distances between these sensor modules, location of buildings and weather characterics are also accounted for. These weather data that are collected by the weather boxes are analyzed by the forecasting team and will determine future weather patterns. The Wind Sensor team is developing a wind sensor that determines wind speed and it's direction.

Developing low cost and reliable wireless environmental sensor modules in mass production and deploying them onto roofs of the buildings around campus can assist in determining the optical places to install renewable energy sources. This in turn will reduce the dependency on fossil fuels and make the University of Hawai'i at Manoa a greener campus.

3 Team Objectives

Team Ant of SCEL is a subproject of the Weatherbox project that focuses on communication between the weather box modules. Weather boxes such as Apple, Cranberry and Dragon fruit collects weather data and transmit those data to other weather boxes, but there are times when the distances between the weather boxes exceeds the range of communication. This is where Team Ant is involved. Team Ant is focused on designing and testing a communication

module that relay meteorological data to assist other weather boxes in communication. The design is capable of functioning under different weather conditions and act as a router to extend the communication range.

Besides coming up with a way to relay the data, the design, testing as well as issues and solutions were documented. This project was created this Fall 2016 so there are no prior information on team Ant, therefore documentation is an important part of this project as it is there to help future members understand the overall system and what has been worked on so far. It will assist future team Ant members to continue the work and improve upon what has already been completed.

4 XBee Background Information

Xbee is a radio frequency (RF) module used for communication. There are different series available with different specifications. The XBee that is being used for this particular project is the XBee Pro S2B (Figure 1), which is a series two XBee module. It was chosen because team Apple used it as their base module so it was also used for team Ant for consistency. According to the datasheet, it has an indoor/urban range of 300 ft (90 m) and an outdoor line of sight of up to 2 miles and a 250 kilobytes per second (kbps) data rate [3]. An advantage that this XBee has over the other XBees is that it has a RP-SMA connector that allow the user to change the antenna to either increase or decrease the range.

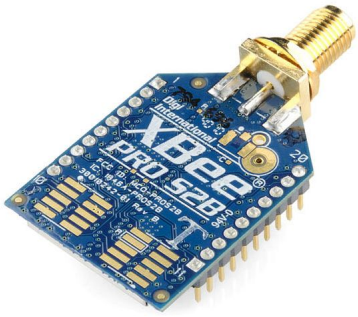


Figure 1: XBee S2B Module

The Universal Asynchronous Receiver/Transmitter (UART) is a communication protocol that the XBee uses to communicate with other modules. Modules that has this communication protocol can communicate with each other without the risk of losing data and devices can connect directly to the pins of the RF module. Communication between a microcontroller and an XBee are displayed in Figure 2.

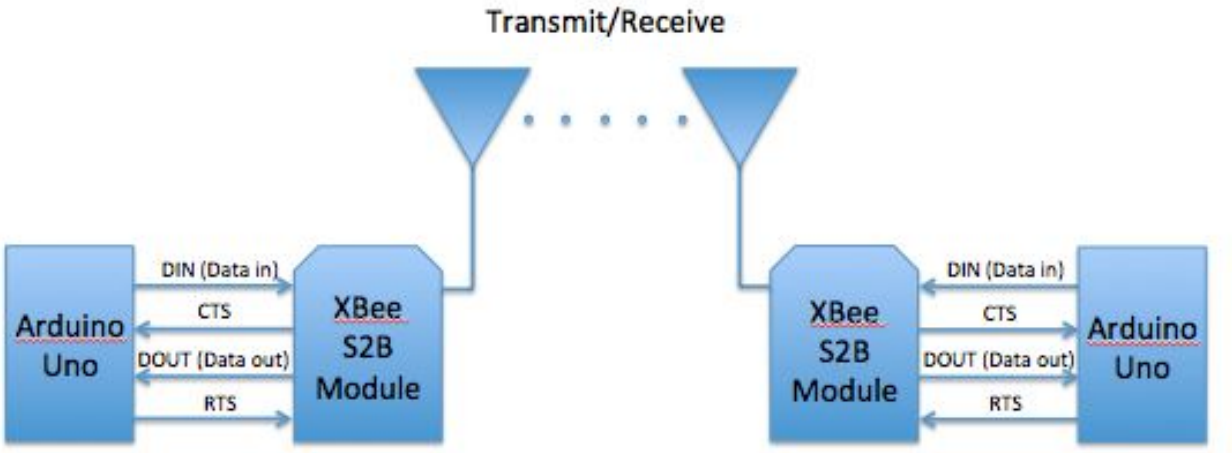


Figure 2: UART data flow diagram

The RTS, or request to send is one of the serial flow control allows the host (Microcontroller) to signal the module (XBee) to not send data in the serial transmit buffer out of the UART. The CTS, or clear to send is the second serial flow control that communicates with

the host (Microcontroller) to stop sending serial data to the module (XBee). The purpose of the serial flow control is to prevent data from overflowing the module with more data than it can handle. Expanding the figure, it can be seen how the data flow works (Figure 3). When data is being received through the antenna, it is goes through the receiver, RF RX buffer and serial transmit buffer and waits until the RTS is allows the data to be transmitted to the microcontroller. A similar case where data is being received from the microcontroller, it goes through the serial receive buffer where it waits until the CTS allows the data to be passed onto the RF TX buffer, transmitter and out of the antenna.

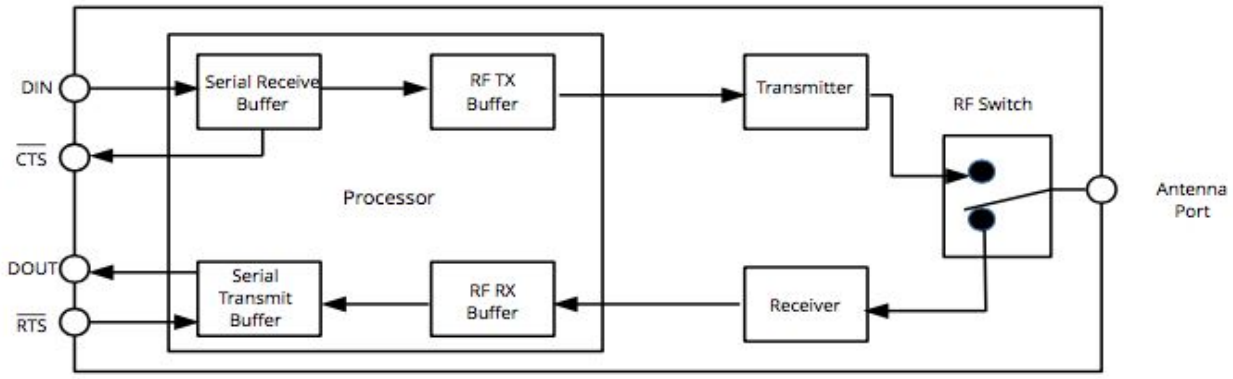


Figure 3: Serial buffers

4.1 XBee ZigBee Network

Zigbee is a specification for communication protocols that is used to create a wireless personal area network (WPAN). It is intended to be simpler and inexpensive than other WPANs such as a router, bluetooth, wifi, etc. It's application is more toward devices of low power, low bandwidth, low duty cycle, low data rates, and small scale projects. The ZigBee network layers consists of PHY, MAC, Network, Application Support Sublayer (APS) and ZigBee Device

Objects (ZDO) layers (Figure 4). These layers won't be discussed since knowing how each layer works is not necessary.

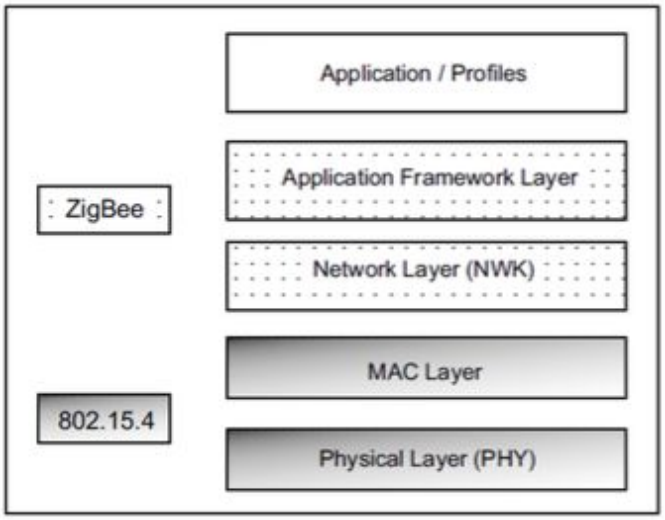


Figure 4: ZigBee network stack layers

There are three different types of devices in the ZigBee network: Coordinator, Router and an End device. A coordinator is the main node that establishes the network channel and allow routers and end devices to join the network as well as assist in routing data. The router node transmit, receive and route data throughout the network channel. The end node can transmit and receive data but cannot route data. Both the router and end node must be interconnected with one another and to the coordinator for communication to occur. In the ZigBee network (Figure 5), the coordinator starts the network and behaves essential like a router. The coordinator along with the router can allow other devices to join the network and route data. The end device must be able to transmit or receive RF data through the router and coordinator once it joins the network. Devices that allow other devices to join the network becomes the parent of that device.

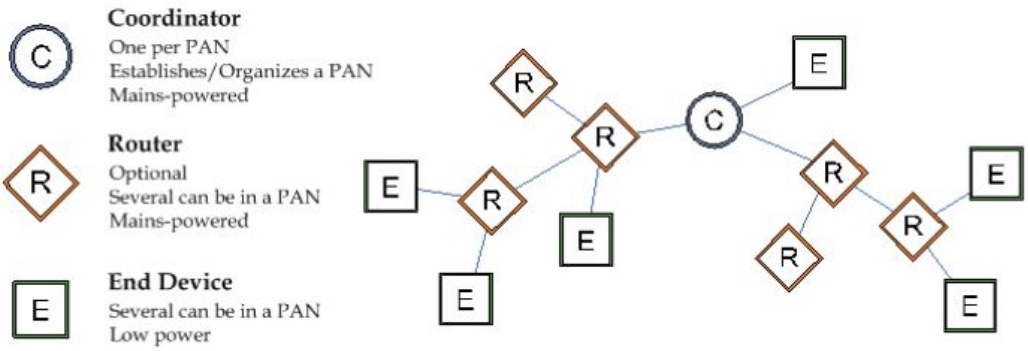


Figure 5: ZigBee network diagram

5 Testing Tools

The software that is being used to test communication of the XBees is the XCTU software.

5.1 XCTU Software

XCTU is a software application that allow users to interact with a radio frequency module such as the XBees, through a user graphical interface (GUI). The GUI is relatively user friendly, which makes it easy to use. Setting up, configuring, modifying parameters and testing the XBee RF modules can be done through XCTU. Updating XBee's parameters, addresses, ID, firmware, API/AT mode are also included in the software. For data transmission, there are frame generators that provide a simple way to generate either an API or AT frame and saves it. The frames are decoded by the frame decoder to view the information contained in the frame. Because of the many features that the XCTU software has to offer, it was mainly used to configure the XBee for communication testing purposes.

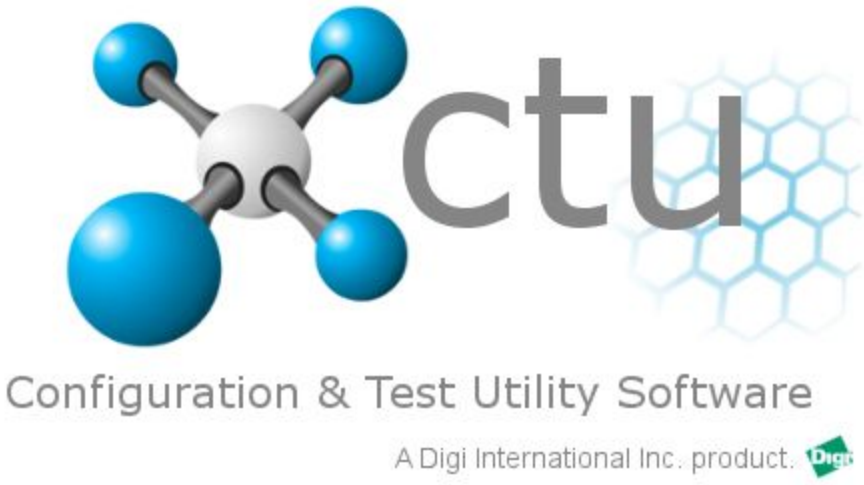


Figure 6: XCTU software used for XBee configuration

5.1.1 XBee Configurations

In order to configure the XBee, it needs to be connected to the computer via XBee USB Explorer. A new module is then added by clicking on the add XBee module button, located on the top left corner (Figure 7).

Figure 7: XCTU start screen

A pop up screen should appear that allows you to change the values of the baud rate, data bits, parity, stop bits and flow control. The baud rate must be set to the same value in order for the XBees to communicate. Since this is only for testing purposes, all the values were left as the default settings (Figure 8).

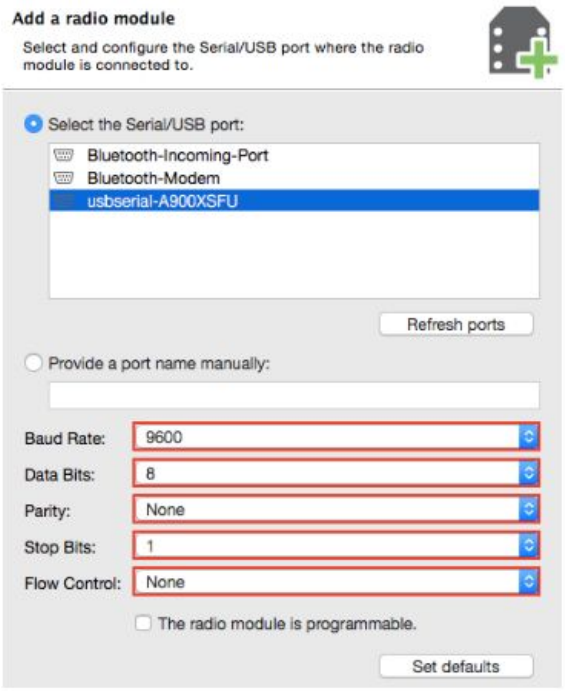


Figure 8: XBee device selection

Once the XBee device has successfully been added, the device needs to be configured in order for communication to occur between XBees module. The XBees that are communicating with one another must have the same personal area network ID (PAN ID). There are a variety of parameters that can be configured such as the Destination Address High (DH), Destination Address Low (DL), I/O (Input/Output) settings, Serial Interfacing, Sleep Modes, Security Parameters and many more [4]. The destination address is for creating a network having a connection between more than two XBee modules. The I/O settings sets the pins on the XBee module. Security parameters encrypt the data being transmitted. These parameters can be configured accordingly to the user's purpose. Once all the parameters are configured as needed, the pencil icon must be clicked to write the new changes onto the XBee module.

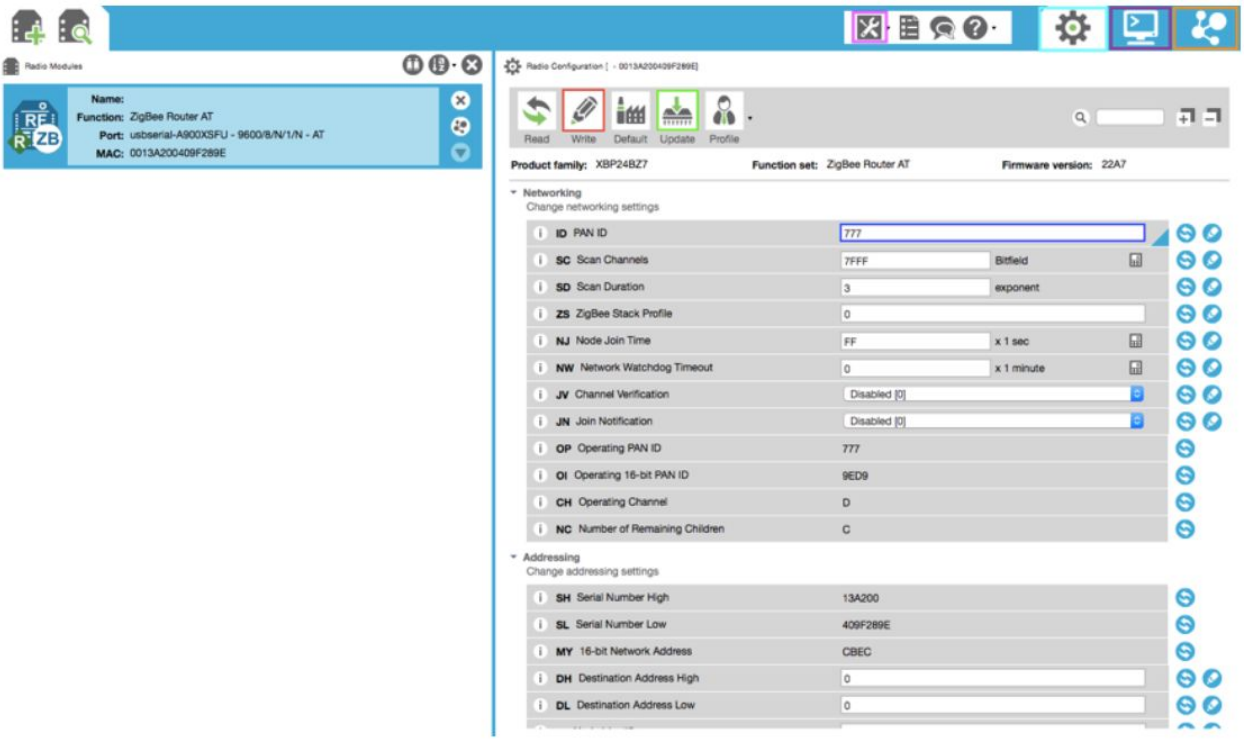


Figure 9: XBee configuration console

5.1.2 Console

The console interface is used to communicate with XBees in AT and API mode (Figure 10). If communicating in API mode, frames are generated and data packets are sent as a package in frames. The console interface can be accessed by clicking on the computer icon on the top right corner.

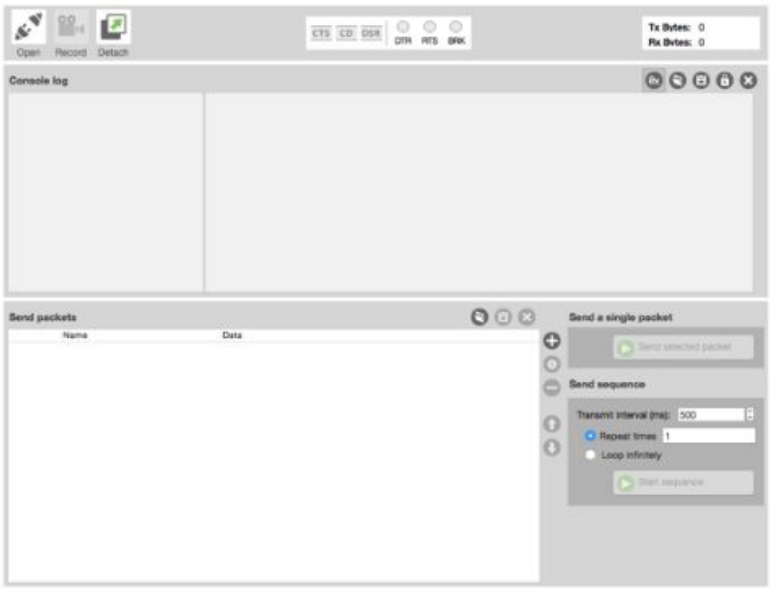


Figure 10: Console interface

5.1.3 Network Working Mode

The network working mode allows the user to observe the XBees connected in the network (Figure 11). The XBees must be on API operating mode for this observation since radio modules in AT mode are not supported in the network discovery interface.

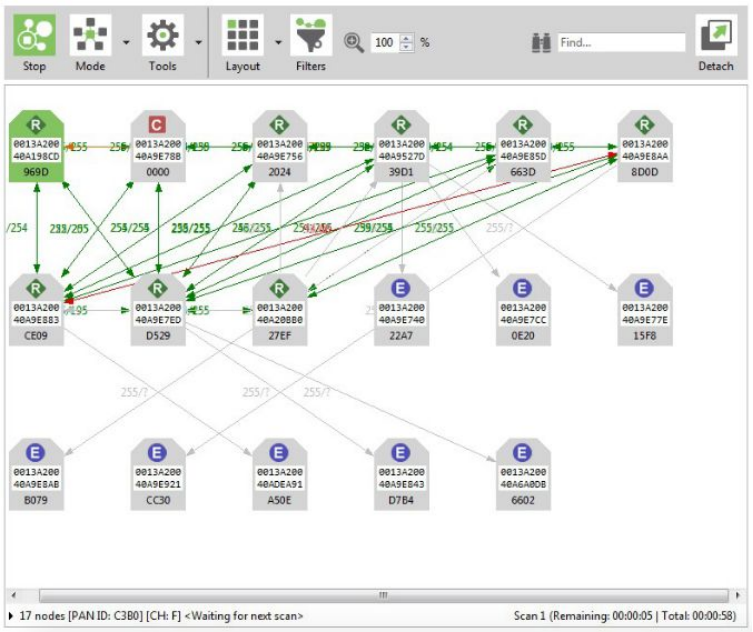


Figure 11: Network working mode interface

5.1.4 Problems and Solutions

Throughout the configuration process, a problem that reoccurred multiple times was the XCTU software not being able to recognize that the XBee module was connected to the computer (Figure 12). This error notified the user that the module needed to be reset by pressing the reset button. Pressing the reset did not resolve the error and the error stils persist. The solution to this was changing the USB cable that connected the XBee breakout board to the computer. An assumption made was that the USB cable isn't compatible could be the cause of the error but there is no concrete conclusion to why this is reason or if something else could have caused this error.

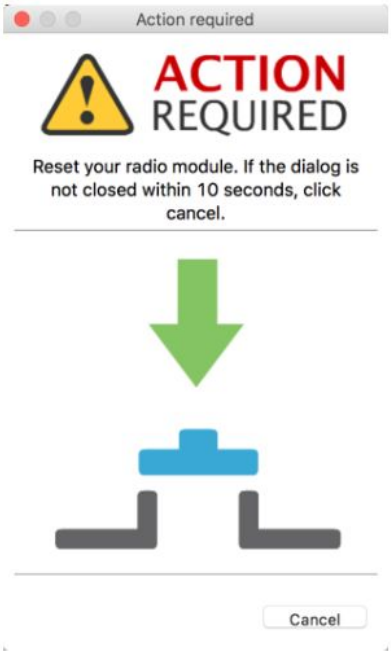


Figure 12: XCTU connection error

5.2 Arduino IDE

The microcontroller used with the XBee module is the Arduino Uno. The Arduino Uno is an open source platform that allows the development of hardware and software projects (Figure 13).

Figure 13: Arduino logo

An Arduino Integrated Development Environment (IDE) is used to program the microcontroller. An open source XBee library available on GitHub provided example codes to demonstrate the communication of XBee modules. The transmit and receive file included codes for communication between the XBee modules, which was uploaded to the microcontroller and tested if the XBee module was communicating with the XCTU software on the computer (Figure 14).

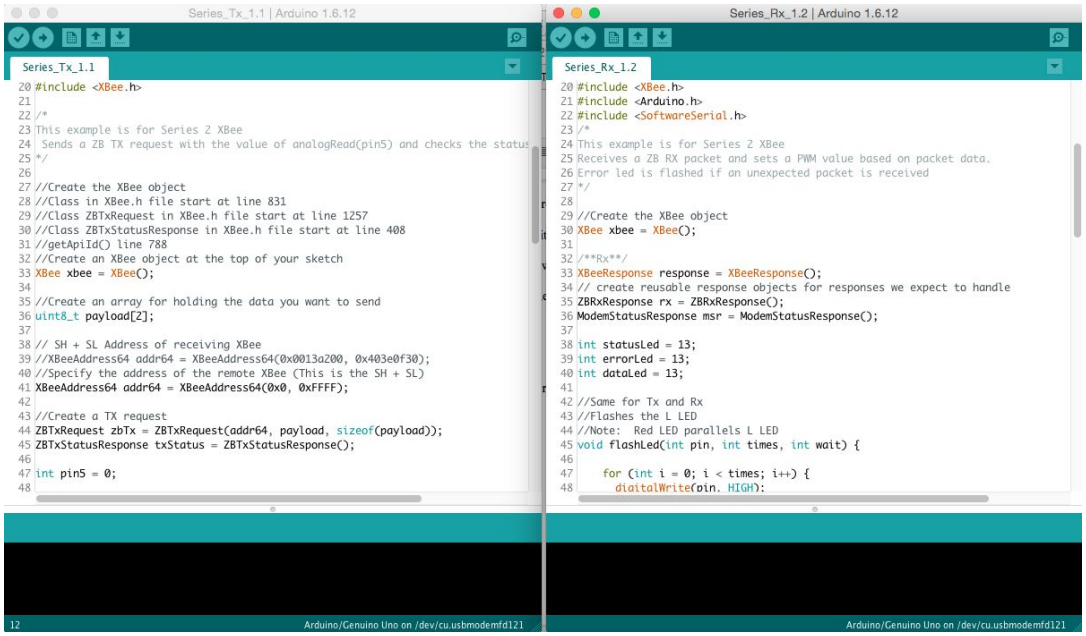


Figure 14: Transmit and receive code

6 Design

6.1 Block Diagram

The overall block diagram of Ant's design is relatively simple compared to other weather boxes design (Figure 15). It describes the overall design by illustrating how each component of the design are associated to other components. In fact, the design references Apple's design with the difference in that there are no sensor components since Ant's board doesn't collect any meteorological data. There are three subsystems of the Ant board, which are color coded accordingly: Power system are red, communication system are green and microcontroller is blue. Observing the power system blocks, the 6 V solar panel collects sunlight and converts them to electricity and is an input to the LiPoly/Li-ion charger. The charger has two outputs, one to charge the battery with a load of 3.7 V and the other goes into the voltage regulator. The voltage regulator regulates the voltage and steps it down to 3.3 V where it goes into the boost

converter and powers the XBee Pro S2B module. The boost converter boosts the voltage up from 3.3 V to 5 V to power the Arduino.

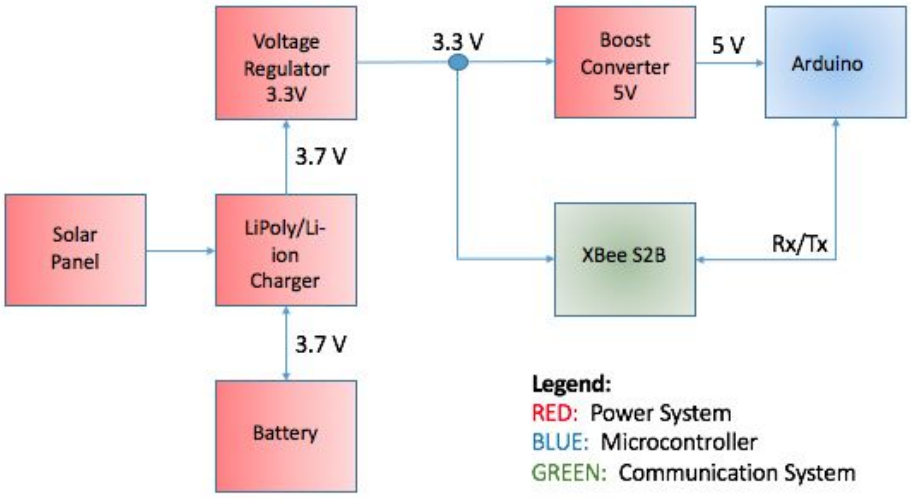


Figure 15: Overall block diagram

6.2 Schematic

Ant version 1.4 schematic references Apple's 3.4.3 schematic because Apple's board is operating successfully so using a design that is working is the most logical decision. There are no sensor components for Ant's schematic because this communication module does not collect any data. It's main objective is to relay data collected by other weather boxes. Removing all the sensor components reduced the complexity of the design as well as the size of the board. The schematic follows the overall block diagram closely (Figure 16) and the subsystems are separated for easy viewing. The reason that there is only one voltage regulator is because the output voltage is the same if connected to the 5 V boost converter and XBee Pro S2B in parallel. The Arduino is the core of the design where it communicates with the XBee module when data is being received or transmitted.

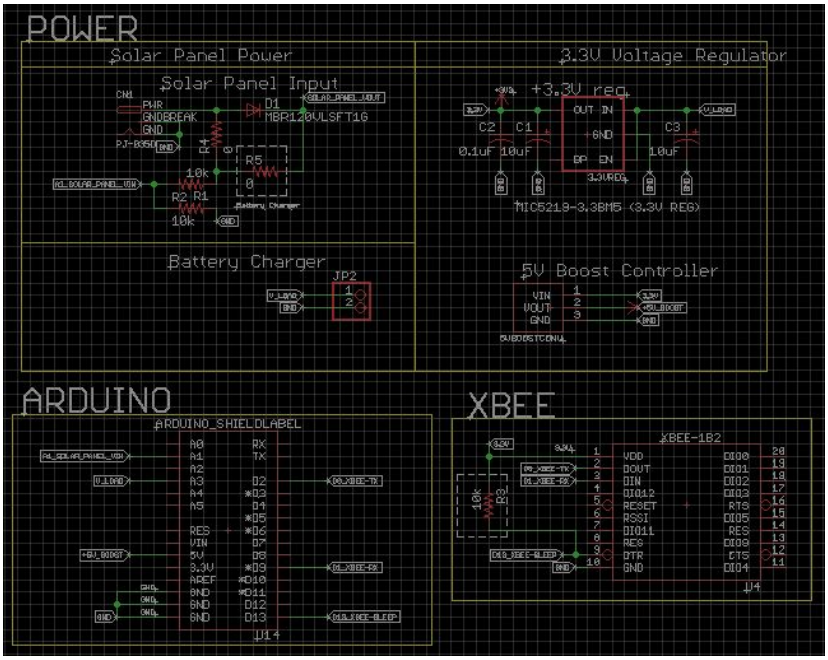


Figure 16: Ant 1.4 schematic

6.3 Board Layout

As previously mentioned in section 6.2 schematic, the complexity was reduced because the board didn't require any sensor components. This reduction made the board layout much more simple with less traces and the components were arranged accordingly. Initially, the first board layout (Figure 17 - Left) had an issue with the battery charger circuit board overlapping with the XBee module. The battery charger circuit board wasn't able to be placed properly since it was too close to the XBee module so the board layout had to be rearranged. The new board layout (Figure 17 - Right) had a minor change where the XBee module was moved to the left on top of the microcontroller. The microcontroller was on the other side of the PCB board so it did not obstruct the XBee module, making the change simple. Moving one component also reduced the number of traces needed to route everything, thus reducing the complexity.

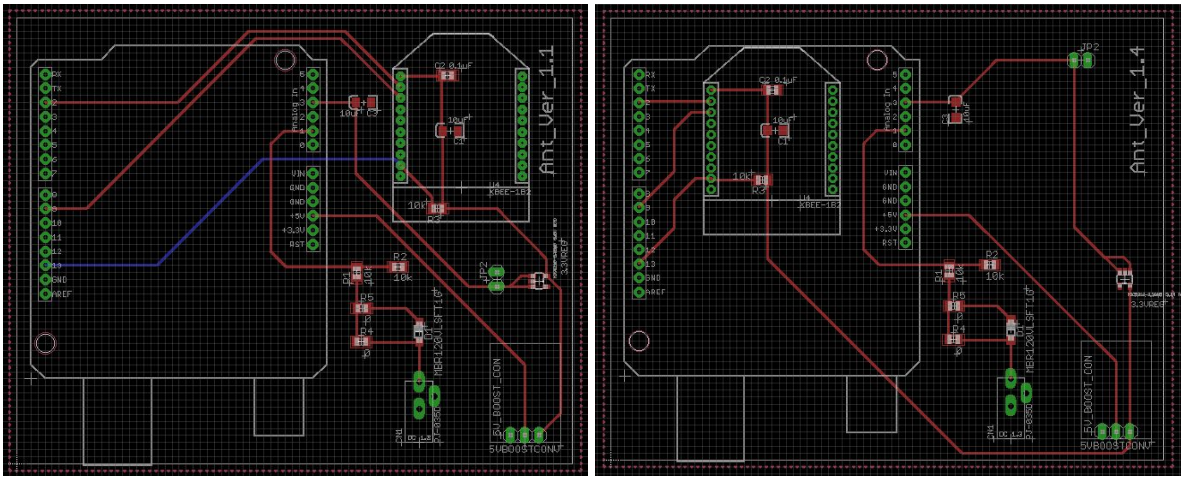


Figure 17: Ant 1.4 board schematic of previous design (Left) and updated design (Right)

7 Debugging Tests

This project had three phases; design preparation, prototype and final design/testing. The idea was to have a structured and organized task that needed to be completed and subtasks within each phase.

7.1 First Iteration - Design Preparation

The first phase is the design preparation and it focuses on getting to know the lab, the objective of each project, documentation, creating a plan, reading up on materials regarding XBees, Arduino and circuit schematic of other teams. The purpose of all of this was to make sure a solid foundation and organized schedule is established in the beginning. Configuration of the coordinator and router XBees also began and completed during this phase.

7.2 Second Phase - Prototype

In this phase, the circuit schematic referenced Apple's design because the PCB board that was fabricated works. The idea was to take a working design and modify it accordingly to what is required. Team Ant's main objective is to relay data so all sensor components were removed

as well as the switch. This reduced the complexity and also reduced the size of the PCB board. Once a schematic was created, the circuit was set up on a breadboard to test if the design was actually correct and working properly. The router XBee was on the breadboard and the coordinator XBee was connected to the XCTU software on the computer. The transmit and receive code obtained from GitHub was uploaded onto the Arduino and made the router transmit or receive depending on which code was uploaded. An LED was also added to the breadboard design to indicate whenever data is being transmitted or received.

7.3 Third Phase - Final Design/Testing

The final phase focuses on finalizing the schematic and creating the board layout on EAGLE along with documentation. There were quite a few issues regarding the EAGLE design because it was later determined that there was connection errors. All components that had a ground wasn't not grounded properly. A solution for this was that the ground was deleted and regrounded, then the signals were traced to make sure that it was grounded properly. Another issue was that the code had an error when uploading to the Arduino. There has been no explanation to why this error came up since uploading the code worked properly previously. Nonetheless, a solution that team Apple found was to rewire the Tx and Rx pins of the XBee to other pins of the Arduino instead of connecting Tx of XBee to Rx of Arduino and Rx of XBee to Tx of Arduino. Once the change was made to the schematic, another PCB board was milled. Communication testing have yet to be conducted but when the board is connected to a power source, it turns on so at the very least, it is know that the power system setup is correct. Although no test have been conducted, it is assumed that the testing results will be the same as the test done on the breadboard design since everything has been routed correctly.

8 Range Testing

A very simply range test was conducted on the prototype breadboard design and only focused on if it was communicating or not. Setup is the same with an XBee on a breadboard and another XBee connected to the XCTU software on the computer. The distance from the breadboard to the computer was from the first floor to the fourth floor of Holmes Hall, approximately 50 m. Another test was from one end of the fourth floor of Holmes Hall to the other end, approximately 100 m. There was successful communication and this is expected because the outdoor line of sight is about 2 miles according to the datasheet.

9 Results

9.1 Breadboard Design Testing

Test setup for the breadboard design had one XBee setup on a breadboard and another XBee connected to a computer (Figure 18).

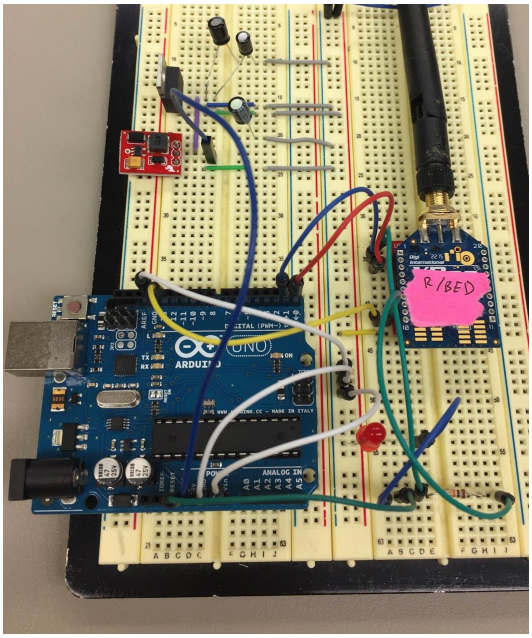


Figure 18: Prototype design on breadboard

In order to test communication, either the transmit or receive code must be uploaded to the Arduino. On XCTU, the console is the user interface where frames are created to send out data and the data being transmitted or received is viewable in the console (Figure 19).

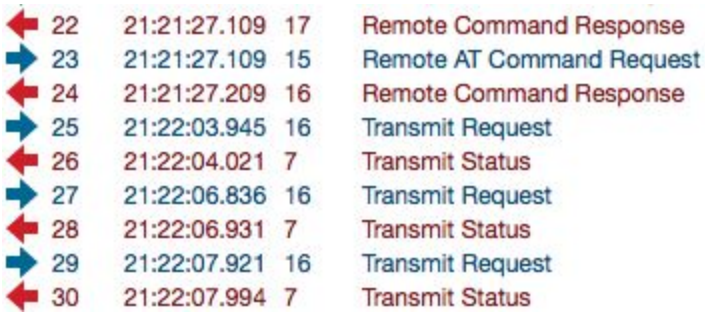


Figure 19: Live feed of data transmitted viewed on console

Initially, communication was a failure because the Tx and Rx pins of the Arduino was connected to the wrong pins of the XBee module. The Tx has to be connected to the Rx and vice versa but Tx was connected to Tx and the same goes for Rx. Once the pins were switched, communication was established and worked the way it was programmed to.

9.2 PCB Design Testing

The Ant PCB board have been milled (Figure 20) but testing for the PCB design have yet to be conducted. The reason that no test has been completed yet was because there were unforeseen issues that piled up one after the after, resulting in backtracking. Issues such as components weren't grounded properly, pins were connected to the wrong pins, unknown code upload error had taken up quite some time. Most of the phase three was spent on debugging and going back to modifying the circuit as well as the code. Although, testing hasn't been conducted on the PCB board, it is assumed that the testing will end with a positive outlook. The PCB board was traced properly and parallel the prototype design. With the prototype design working properly, the testing for the PCB board should yield the same result.

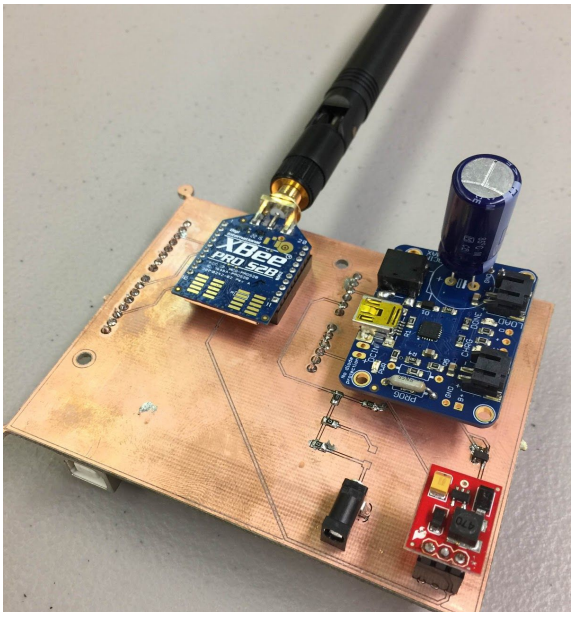


Figure 20: Ant's PCB board

10 Bill of Materials (BOM)

The Ant's bill of materials (BOM) is shown in Figure 21. Having a record of this is important because it is good practice to note down every components that went into designing your device. It also make things easier to track such as quantities. This also allow future members to see what components need to be purchase if they decide to reproduce the same product. The vendor and product ID was included so that it could easily be search for if more needs to be ordered. Compared to other weather boxes, there are less components used and the total cost is a lot cheaper. This is reasonable since there are no sensor components in our design, therefore the total cost should be relatively cheaper than the other weather boxes.

Ant Board Bill Of Materials						
#	Part Description	Part Name	Vendor	Product ID/#	Unit Cost	Quantity
1	Solar Charging Circuit	USB LiPoly/Li-Ion Charger (3.7/4.2V) MCP73871	Adafruit	390	$17.50	1
2	Microprocessor	Arduino Uno R3	Adafruit	50	$24.95	1
3	Wireless Transciever	Digi International XBee Pro S2B	Adafruit	967	$37.95	1
4	Battery	Tenergy Li-Ion 18650 3.7V 6600 mAh	Adafruit	353	$29.50	1
5	Solar Panel	Large 6V 3.4W Solar Panel 3.4 Watt	Adafruit	500	$39.00	1
6	Duck Antenna	2.4GHz Duck Antenna RP-SMA - Large	Sparkfun	558	$9.95	1
7	Voltage Booster	5V Boost Converter: NCP1402-D	Sparkfun	10968	$5.95	1
8	Voltage Regulator	3.3V Regulator: MIC5219	Digi Key	SOT23-5	$0.74	1
	Unit Sub Cost					
	$158.85					

Figure 21: Ant's Bill of Materials (BOM)

11 Power Budget

The power budget shown in Figure 22 only include the values from the data sheet because it wasn't calculated. The importance of a power budget is to determine the module's overall power consumption as well as observe which component uses the most power. The values found on the datasheet includes the idle, typical and maximum current draw of each component. The component's supply voltage is also included individually. The Arduino Uno R3 and XBee Pro S2B is separated by 5V and 3V, respectively. An assumption made about the XBee Pro S2B is that it is 99.9891% of the time and 0.01093% of the time that it is transmitting.

Ant Board Power Budget			
5V Module	Data Sheet Values		
Part Name	Idle Current (mA)	Typical Current (mA)	Max Current Draw (mA)
Arduino Uno R3	0.0001	20	50
Total:	0.0001	20	50
3V Module	Data Sheet Values		
Part Name	Idle Current (mA)	Typical Current (mA)	Max Current Draw (mA)
XBee Pro S2B	0.0035	15	220
Total	0.0035	15	220
Battery Supply			
Part Name	Supply Voltage (V)	Discharge Rate (mAh)	Usable Energy
3.7V 6600 mAh	3.7	6600	80%

Figure 22: Ant power budget

12 Future Improvements

Since team Ant was created this semester, there will be many future improvements. Testing communication of the PCB board still needs to be conducted to determine if the design works just like how the prototype worked. Another microcontroller that is smaller in size and less functionalities could be implemented instead of the Arduino. The Arduino has many functionalities that isn't needed and the dimensions is rather big so reducing these features can also cost and space. Power budget also has to be calculated and compared to the datasheet values to determine the actual current as well as the power that the PCB board consumes. Documentation is also another improvement to ensure that past and current work can be passed down to future members.

13 Conclusion

This report covered the general background of XBee Pro S2B and its applications as well as the software used to configure and test the module. The circuit schematic of Ant referenced Apple's circuit schematic and the complexity and size was greatly reduced due to the fact that many unnecessary components were removed. Testing began on a prototype design to ensure that the communication code was working properly before moving onto the final design. The final PCB design was milled and all components were soldered on. Testing have yet to be conducted for the final PCB design but successful communication on the prototype yields a positive assumption that it will yield the same result for the PCB board.

14 References

[1]"Hawaii Profile", *Eia.gov*, 2016. [Online]. Available: https://www.eia.gov/state/print.cfm?sid=HI. [Accessed: 15- Dec- 2016].

[2] Kaleo Campus News. (2013 October 3). *UH's $35 Million Electricity Bill Among Key Topics Discussed At CampusWide Conversation* [Online]. Available: http://www.kaleo.org/news/campus/uhsmillionelectricitybillamongkeytopicsdiscussedat/article_916166522c8911e3acc60019bb30f31a.html.

[3] 2016. [Online]. Available: http://www.digi.com/resources/documentation/digidocs/PDFs/90000976.pdf. [Accessed: 15- Dec- 2016].

[4] Digi International. XCTU Next Generation Configuration Platform for XBee/RF Solutions [Online]. Available: