Final Report

Team #27
Alexander Bermeo, Randall Barnes, Cory Packard

Project for: Faculty Sponsor

Faculty Contact:

Dr. Benedito Fonseca
Northern Illinois University
Engineering Building 344
590 Garden Road
DeKalb, IL 60115-2854
(845) 753-0557
# Table of Contents

Abstract ................................................................................................................................................. 4

1 INTRODUCTION ................................................................................................................................. 5
  1.1 Background ..................................................................................................................................... 5
  1.2 Purpose of Project ............................................................................................................................ 5
  1.3 Previous Work Done by Others ....................................................................................................... 6
    1.3.1 Existing Products ..................................................................................................................... 7
    1.3.2 Patents Search Results ............................................................................................................ 8

2 PROJECT DESIGN ............................................................................................................................... 9
  2.1 Optimum Design ............................................................................................................................. 9
    2.1.1 Alternative Design I: Using Frequency Modulated Receiver with Arduino .................... 9
    2.1.2 Alternative Design II: Utilizing Raspberry Pi versus Arduino and FM Receiver ........... 14
    2.1.3 Alternative Design III: Battery Power Supplies versus DC Adapters ......................... 16
    2.1.3 Objective .............................................................................................................................. 19
    2.1.4 Setting Up SDR ..................................................................................................................... 21
    2.1.5 Testing the SDR .................................................................................................................... 22
    2.1.6 Capturing Raw Samples of the SDR Receiver .................................................................... 23
    2.1.7 Creating the Relationship between SDR and Python ....................................................... 25
    2.1.8 Displaying the Collected Raw Samples in Python ............................................................. 26
    2.1.9 SDR and Raspberry Pi Configuration .................................................................................. 26
    2.1.10 Configuration the XBee Modules ...................................................................................... 27
    2.1.11 Raspberry Pi and XBee Module Configuration ................................................................. 29
    2.1.12 Sensor Configuration through XCTU ............................................................................... 30
    2.1.13 Design Data Fusion in Python to Combine Two Sensors ............................................. 32
    2.1.14 Design the Detector in Python with Two Sensors ........................................................... 33
    2.1.15 Creating the Wireless Sensor Network and Using ZigBee ........................................... 33
    2.1.16 Design Data Fusion in Python to Combine the Entire Wireless Sensor Network ....... 34
    2.1.17 Design the Detector in Python with the Entire Wireless Sensor Network ................... 34
    2.1.18 Locating the Direction of the Unauthorized Transmissions ........................................... 35
    2.1.19 Displaying the Results ......................................................................................................... 36
Abstract

In the dawn of innovation in which technology interconnects the machine and user to more intentional levels than ever before, security of interconnected technology must be harnessed in order to ensure confidentiality of the consumer’s information in addition to the protection of device performance. A transmitter could be used to maliciously emit an unauthorized signal in order to gain access to sensitive information or control over a device. In order to prevent such an attack, a wireless detector should be implemented to warn a user of malicious activity.

The detector is comprised of a mesh network of multiple sensors and is required to scan a frequency spectrum. When an unauthorized signal is detected, the user must be alerted. Each sensor is comprised of three main components: a software-defined radio, a Raspberry Pi, and an XBee device. The software defined radio is used to scan a given FM spectrum for signals. The Raspberry Pi is used to connect the SDR (software-defined radio) to the XBee device. Once the Raspberry Pi device receives information, it relays that information to the XBee. The purpose of the XBee is to wirelessly connect each node to a main fusion center. The fusion center is where data processing and detection takes place. Additionally, the fusion center displays information through a user-interface and alerts the user when an unauthorized transmission is detected.
1 INTRODUCTION

1.1 Background
The evolution of technology is increasing exponentially which increases the use of autonomous devices that rely on wireless signals to operate and communicate with other equipment. It is now difficult to locate any modern-day operation that does not incorporate the use of these devices. A disadvantage to the growth of autonomous technology is the unpredictable methods of attack that organizations or individuals can present to various communities. One form of attack may be to tamper with the communication of these devices that can result in overtaking the device or sensitive information becoming compromised. The consequence of a device being hijacked has its own serious possibilities that can lead to damage to the device, infrastructure, and civilians. Prevention of these situations are to begin in the future in order to maintain the same pace of the expanding world of new technology. The proposal introduces a device that detects unauthorized transmissions in a frequency spectrum using a wireless sensor network.

1.2 Purpose of Project
The proposed project is a wireless sensor network designed to alert to any unknown frequency modulated transmissions within a region. The purpose of creating the device is to establish a safer and a more reliable environment for the use of autonomous technology. The functionalities of autonomous technologies require the use of wireless signals to operate and to communicate. A device can be vulnerable to tampering within their communications from outside sources and the demand to protect these devices are high. The sensors within the wireless network combine the collected data from a certain FM channel and process the information through a detector. A decision from the detector can notify the user if there are unauthorized transmissions inside the defined parameter. Opportunities for the user to take executive action is
then created to prevent the unauthorized transmissions from potentially damaging the device or obtaining sensitive information. Functionalities of the proposed project allows personal users or companies to develop effective procedures in the case of a detection of unauthorized transmissions signals within their defined parameter. One method to stop a rogue transmission would be to implement a jammer into the network. The purpose of the jammer is to prevent the rogue device that is transmitting signals from conducting any operation it was programmed to. Another method would be to physically remove the rogue device after it has been triangulated. The overall outcome of the process of detecting and preventing unauthorized transmissions can prevent companies or individuals from becoming a target.

1.3 Previous Work Done by Others
The article, *Arduino Based Wireless Intrusion Detection Using IR Sensor and GSM* by Prakash Kumar and Praeep Kumar is particularly useful because of the utilization of an Arduino [1]. It is anticipated that the project would use a Raspberry Pi in combination with Zigbee. Kumar explains reasoning and a method to create a wireless intrusion detection system for a GSM network. Arduino-based wireless intrusion detection is like the proposed design because it deals with detecting anomalies or signals that do not fit within a predefined signature. Another perk of utilizing a wireless sensor network (WSN) compared to a wired network is that the wireless networks tend to be much cheaper. Additionally, the use of an anomaly-based system creates a baseline for normal traffic which makes it easier to identify when unauthorized/unrecognized signals are sent to the network. The setup used in the article is for WI-FI networks; however, an anomaly-based detection system could also be used with an FM network.

Michel Barbeau and Evangelos Kranakis explains in better detail the specifics of an anomaly-based detection system using radio frequency fingerprinting (RFF) [2]. The detection
method described in the article compares “fingerprints” of several different radio devices in a system. Additionally, the detection method setup utilizes a classifier with a transient and feature extractor along with a Bayesian filter to increase detection and correct classification. The completed system described in the article had an average success rate of 94-100%. Once again, the functionalities of the setup are used for 802.11 networks and detection of spoofed MAC addresses; however, can also be used with an FM network.

*Spectrum Patrolling with Crowdsourced Spectrum Sensors* by Ayon Chakraborty and Arani Bhattacharya has similarities with the proposed design in that it uses multiple sensors to monitor RF usage in a spectrum [3]. The article explains in detail about specific sensor differences and diminishing the cost while simultaneously increasing the detection probability as much as possible.

1.3.1 Existing Products
For the project, there are not many commercial products on the market currently. For the individual components such as ZigBee, Arduino, and the FM receivers, there are many alternatives to consider. In the case of ZigBee, it would be possible to utilize a different wireless communication device such as Bluetooth or Wi-Fi. However, it becomes difficult to use these products due to difficulty found in trying to modify these products. ZigBee also presents a simple solution when it comes to breaking down the communication protocols between devices. There is also a ton of documentation on ZigBee devices. Additionally, Arduino provides a simple, easy to use interface for connecting the FM receivers to the XBee. Arduinos price range are much more suitable for the project in compared to other alternatives. The functionalities of Arduinos range from hobby projects to robotic implementation in certain companies. For the FM receivers, software defined radio receivers would be used. The software defined radio provides the ability to sense FM frequencies while adding simplicity when building the final product.
1.3.2 Patents Search Results

*System and method for monitoring and enforcing a restricted wireless zone*

Diener, N. R. et. al wrote a patent for a system that monitors/enforces unauthorized signals in a wireless zone [4]. The network also utilizes a jamming device in order to try and jam unauthorized transmitters in the area. If an unauthorized unit is detected, the jamming device would emit a frequency to disable the unauthorized transmitter. Unlike the proposed system, the patent demonstrates a system that operates in the unlicensed 2.4 to 5 GHz 802.11 WLAN band. Whereas the proposed network would operate in the FM band (approximately 88-108MHz).

*Detecting unauthorized radio communications devices*

Worley and Devoy have a patent for detecting unauthorized radio devices. It works by comparing the wanted signal with the unwanted signals and with respect to a predefined threshold [5]. The described method in the article utilizes many filters and different thresholds for various frequencies.

*Method and apparatus for detecting illicit RF data transmissions*

Grube G et. al describes method and an apparatus for detecting illicit or unauthorized data transmissions in an RF network [6]. The apparatus compares the incoming information with the anticipated information type. The apparatus should detect an illicit signal if the signal is not of the same data type as a baseline data type.
PROJECT DESIGN

2.1 Optimum Design

2.1.1 Alternative Design I: Using Frequency Modulated Receiver with Arduino

Beginning with the initial raw data gatherer, the FM receiver conducts periodic monitoring of the frequency spectrum. FM was chosen rather than AM because of its higher signal-to-noise ratio [7]. The project is intended to detect any unauthorized transmissions within an area of 1600 square meters, the sensors are distributed in a checker-like form to increase the accuracy of the detection. The antenna of each receiver obtains a frequency wave which is then processed through an electronic amplifier to increase the power of the signal. Demodulating is not performed within the receivers because the data gathered must be original information to obtain knowledge about the power within the spectrum. After the receiver has obtained the monitored information from its surrounding area, the data is then processed through the Arduino.

The Arduino Uno microprocessor has three functionalities which are processing the data from the receiver, transitioning the data to be sent through a wireless network using a module and supplying power to the other components of the sensor. Processing the data from the receiver requires code that packages the raw data received. The code is then designed to accommodate the processing of each sensor value through the fusion center. After the Arduino completes the code involving the receiver data, the following code may be tailored for the transferring of data using a wireless sensor network. The bridge that allows the Arduino to use the wireless network is an XBee-PRO S2C. The code must be constructed to allow a communication between the XBee while transferring the collected data through the wireless network. The layout of a sensor is illustrated in Figure 1.
Figure 1: Sensor Components [8]

The method of transporting data through devices is based on a wireless sensor network configuration [9]. In order to operate in a wireless sensor network, a standard node connectivity must be chosen. The standard operating was chosen to be ZigBee because of its multi-hop feature and mature technology. For the Arduino to operate with ZigBee, a module would be placed in-between the two and function as a gateway. The XBee-PRO S2C module is ideal for the project because it supports the communications protocols of ZigBee [10]. A breakout board must be attached to the XBee devices for the Arduino to serve as an adapter and allow for configuration.

The network layers of Zigbee consist of three important aspects which are routing, ad hoc network creation, and self-healing mesh. Routing defines how one radio can pass a message through a series of other radios, ad hoc network creates the entire network of radios, and self-healing mesh is the reconstruction process when a node is broken or missing. These three layers of ZigBee are modified using the XBees module. A ZigBee network consist of a coordinator,
router, and end device [11]. The coordinator should be connected to the fusion center and be located at the base of the project. The coordinator should behave as the brain of the wireless network and controls the other XBee devices. The router would behave as a messenger for communications between the XBees that are a long distance away from the coordinator. The end device behaves as a normal node and can be placed under sleep mode to save power. The network topology for the project is designed specifically to maximize the detection outcome by using an extended star network arrangement. A diagram of the topology of the ZigBee network can be seen in Figure 2. The design allows for highest detection rate because of the evenly distributed sensors.

![ZigBee Network Topology](image.png)

**Figure 2: ZigBee Network Topology**

At the center of the network, the coordinator may interact with a fusion center that may be programmed on a laptop. The process of data fusion is integrating the multiple collected data from each sensor. The main purpose of the data fusion center in the wireless sensor network is to implement a greater quality of service to achieve reliable and accurate decisions. An example of
the process of a fusion center is shown in Figure 3, where the four different set of data are being transferred to the fusion center for combining. The technique for the process may use probability theory because of its functionalities in random events that may be applicable to the design. Bayesian method may be used within the process of probability theory. The Bayesian [12] methodology provides consistent approach for dealing with uncertainty associated with the measurements of each sensor. The programming language used to design the fusion center and detector is Python because of its sensor fusion, tracking features, and high-performance computing. Both coding for the data fusion and detector may involve Bayesian methods. Once the data fusion is complete, the next section of code may consist of a detector with an established threshold. Calculating the appropriate threshold for the design may involve running a sample operation of the project when there are no unknown transmissions within the area. After the fusion center completes gathering the results of each sensor, the establish threshold can then be determined based on those values. The threshold may be developed through the power measurements of the frequency spectrum. If there no transmissions signals in the area, the power measured in the spectrum may be relatively low. When the sensors collect the measurements of the spectrum while a signal is being transmitted within the area, the power measured may be higher. The detection is based on the power of the transmission signal. The detector may process the data transferred from the fusion center and compare that data with the threshold. If the value is above the threshold, there is an unknown transmission detected. When the value is below the threshold, the defined area is absent of any unknown transmissions. There may be extra code that would display essential information involving the decision of the detector.
Displaying information about the overall result of the design may consist of a program in Python. The application should illustrate four main components of the project, the frequency spectrum, placement of each sensor, location of the unknown transmission, and the power within the spectrum. Using a simulated frequency spectrum may allow the user to understand what is being monitored, as well as indicating where on the spectrum the unknown transmission can be classified. An illustration of the system diagram may allow the user to view where each sensor is located and would also illustrate the precise location of the unknown transmission. The last essential information presented would be the power calculated within the spectrum. It may display the set threshold of the detection and notify the user if the calculated power is greater than the threshold. The overall presented information could allow any user to understand the fundamentals of the project.

A wireless sensor network must operate for an efficient amount of time. The power source for each sensor could be connected to the Arduino which should power the FM receiver and XBee module. Using eight AA batteries to power the sensor may give each a lifespan of approximately nine hours. For the project, the nine-hour span is enough to demonstrate the
functionalities of the design. End devices in the ZigBee network could allow for better power consumption because of their *sleep mode* feature. The sleep mode feature allows them to power themselves down intermittently throughout the monitoring process.

An enclosure may be built for each sensor to prevent any damaging possibilities from the outside environment. The enclosure may be made of plastic or practical board material. These two options may allow for an inexpensive solution for the protection of the sensors. 3D printing the enclosure may require the use of SolidWorks to create the design that should suit the parameters of the circuits. As seen in Figure 4, the enclosure may be developed to reduce cost while adding the rough protection to the circuits. The design could be created to fit only the required circuits and power source in a tight space. The small design reduces the amount of material needed to fully enclose the entire system.

![Sensor Enclosure](image)

**Figure 4: Sensor Enclosure**

2.1.2 **Alternative Design II: Utilizing Raspberry Pi versus Arduino and FM Receiver**

The method of communicating the received raw data from the FM receiver over the frequency spectrum is through the microprocessor. Original design consists of an Arduino Nano which contain the required functionalities needed to conduct the experiment, but it also creates limitations on the sensor. These limitations can involve the processing capabilities, power
consumption, and requiring a specific FM receiver. A suitable alternative for the Arduino is a Raspberry Pi, which contains its own limitations and advantages.

Arduinos are microcontroller motherboards which can run one program at a time repeatedly. Microcontrollers, in general, are good for doing simple tasks multiple times. An alternative component compared to the Arduino microcontrollers is the Raspberry Pi minicomputer. Raspberry Pi is a small general-purpose computer that usually runs with a Linux operating system. It can run multiple programs at once and is more complex than an Arduino. The Arduino is best for doing simple repetitive tasks, whereas the Raspberry Pi is good for running more complicated programs and doing more intense calculations.

The purpose behind using a microprocessor in the sensor network is to be a connection to route information from the FM receiver to the ZigBee device. The microprocessor needs enough power capacity to allow for reliable operation of the SDR. The decision between Raspberry Pi and Arduino was made based on the type of SDR used. A USB compatible SDR may be more suitable for use with a Raspberry Pi due to the USB port built into the board. Other components that may side with using the Raspberry Pi minicomputer instead of the Arduino is any add on features that may be implemented on each sensor. These can refer to liquid crystal displays, rechargeable battery capabilities, and any added complex processing features to improve the overall abilities of the sensor.

The Raspberry Pi can run more complex calculations, which means more computations could be done inside the sensor instead of at the fusion center. However, extra computation power comes at a price. A Raspberry Pi costs $35, whereas the Arduino Uno only costs $15, and the Arduino Nano only costs $8. Price difference becomes a significant deciding factor in which microprocessor to use. The use of Raspberry Pi could allow for the use of fewer sensors.
Figure 5 gives a detailed description on what the differences are between the Raspberry Pi and Arduino. One can observe that the Raspberry Pi has a much faster 1.2GHz processor, than the Arduino 16MHz processor. The Raspberry Pi also has a dedicated graphics processor, and considerably more RAM than the Arduino. The Raspberry Pi consumes considerably more power and may be much more difficult to power using batteries than the Arduino. Both the Arduino and the Raspberry Pi are powered with a 5V input. A USB port is also included on the Raspberry Pi. Depending on the Raspberry Pi, it draws around 3.7-5.1 Watts of power (at 400% CPU load) and 1.4-1.9 Watts at idle. The unmodified Arduino only draws around 100mW of power. The Raspberry Pi could only be powered for approximately 3 hours with a 3000maH battery pack whereas the Arduino could last longer.

![Figure 5: Raspberry Pi Specifications](image)

<table>
<thead>
<tr>
<th>Introduction Date</th>
<th>SoC</th>
<th>CPU</th>
<th>Instruction set</th>
<th>GPU</th>
<th>RAM</th>
<th>Storage</th>
<th>Ethernet</th>
<th>Wireless</th>
<th>Video Output</th>
<th>Audio Output</th>
<th>GPIO</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>802.11n / Bluetooth 4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HDMI / Composite</td>
<td>HDMI / Composite</td>
<td></td>
<td></td>
<td>$35</td>
</tr>
<tr>
<td>2/26/2016</td>
<td>BCM2837</td>
<td>Quad Cortex A53 @ 1.2GHz</td>
<td>ARMv6-A</td>
<td>400MHz VideoCore IV</td>
<td>1GB SDRAM</td>
<td>micro-SD</td>
<td>10/100</td>
<td>none</td>
<td>HDMI / Composite</td>
<td>HDMI / Headphone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/25/2015</td>
<td>BCM2835</td>
<td>ARM11 @ 1GHz</td>
<td>ARMv6-A</td>
<td>250MHz VideoCore IV</td>
<td>512 MB SDRAM</td>
<td>micro-SD</td>
<td>none</td>
<td>none</td>
<td>HDMI / Composite</td>
<td>HDMI / Headphone</td>
<td></td>
<td>$5</td>
</tr>
<tr>
<td>2/2/2015</td>
<td>BCM2836</td>
<td>Quad Cortex A7 @ 900MHz</td>
<td>ARMv7-A</td>
<td>250MHz VideoCore IV</td>
<td>1GB SDRAM</td>
<td>micro-SD</td>
<td>10/100</td>
<td>none</td>
<td>HDMI / Composite</td>
<td>HDMI / Headphone</td>
<td></td>
<td>$35</td>
</tr>
<tr>
<td>7/14/2014</td>
<td>BCM2835</td>
<td>ARM11 @ 700MHz</td>
<td>ARMv6</td>
<td>250MHz VideoCore IV</td>
<td>512MB SDRAM</td>
<td>micro-SD</td>
<td>10/100</td>
<td>none</td>
<td>HDMI / Composite</td>
<td>HDMI / Headphone</td>
<td></td>
<td>$35</td>
</tr>
</tbody>
</table>

2.1.3 Alternative Design III: Battery Power Supplies versus DC Adapters

In order to fulfill the goal of detecting unauthorized transmissions, there should be a source of power for the electrical components. In modern applications, the design would be
used in wireless applications, so having all the sensors attached to the same power supply would not be feasible in most scenarios. Realistically, it would be potentially difficult to power all the sensors with power only supplied by the standard grid due to the need for multiple extension cords and unguaranteed outlets depending on geographic location. Ideally, the design would consist of self-powered radio receivers to imitate independent sensors within the wireless network.

To determine the appropriate battery power supply, the power demands should be evaluated. For the possible microcontrollers, there are power specifications to consider. For an Arduino, an input voltage between 7 and 12 volts is recommended in which the output current is limited to .5 amps when powered by USB and 1 amp when powered by external battery or power supply (x). A Raspberry Pi is powered by a 5-volt micro USB connection. It is recommended that the power supply can supply 2.5 amps. Most Zigbee devices have a power demand of 1 milliwatt (Cirronet). The typical operating voltage of a Zigbee is 3.3 V with a current draw of .15 amps. Without knowing the exact specifications of the radio receiver, one can guess the receiver would have power consumption around the same magnitude.

![Figure 6: 6 Volt Alkaline Battery Supply](image)
With this information, one can conclude that a standard 5 V battery power supply may not be the greatest solution due to the strong current draw from the circuit components unless multiple supplies are used in parallel to increase the overall power capacity of the sensors. It would increase the price of the design with battery packs like the pack displayed in Figure 6. The price of the battery pack in Figure 6 is $6.98. The battery pack may not last very long especially with the nature of nonchargeable batteries. The more ideal solution may be using a rechargeable 12volt battery pack. It would provide more voltage support for the sensors and could be shared among some of the other circuit components rather than each sensor having an individual supply in this case. The price for such a supply like the one pictured in Figure 7 is $16.33 which would be reasonable if shared among multiple components. The supply would allow more independence from the grid to allow for more testing in other environments such as outdoors. More than likely, there could be a combination of battery power packs, laptops, and DC adapters like Figure 8 for the sensor components. The price range for the DC adapter is about 7 to 10 dollars. It would diversify the power sources and better represent independent sensors a part of the wireless sensor network.

Figure 7: 12 Volt Rechargeable Battery Supply
2.1.3 Objective

The proposed project design consists of a network of distributed SDR receivers that jointly detect unauthorized transmissions in a frequency band. The receivers are accompanied by a Raspberry Pi, XBee module, and a power source. A combination of each of these components create one sensor within the distributed wireless network. For the communication between these sensors, the functionalities of a Zigbee network is used. The sensors transfer information to the controller of the wireless sensor network which is the coordinator. The coordinator is responsible for maintaining the network and is used to adjust the routing pathway of each sensor.

Information processed by the coordinator is then combined using data fusion. Once the collected data is combined, the information is then evaluated by a detector with a set threshold. The determined threshold decides whether there are unauthorized transmissions within the area. After the detector creates a decision, the information is transferred into a program that displays the location of each sensor, strength of the unauthorized transmissions, average power calculated,
and a frequency spectrum indicating the unauthorized transmissions. The application of the project operates in a periodic fashion, monitoring the frequency spectrum continuously.

A sensor must be able to receive FM signals from frequency ranges that are allowed by FCC laws. Because the project is aimed to prevent hackers from interfering with certain devices, the unauthorized transmissions are expected to have low power. The receivers chosen have capabilities to monitor frequency bands and detect any low power signal. The receivers collect raw data samples from the specific frequency range and is accompanied by a minicomputer for the computing. The minicomputer most appropriate for the operations of the project is the Raspberry Pi 3 because of its compact physical form and low price. The data from the receivers is transferred to the connected Raspberry Pi for processing which includes the preparation for wireless communication. The Raspberry Pi is connected to a power supply which allows the other devices connected to operate in a wireless fashion. The wireless sensor network consists of 10 sensors distributed around an area that utilizes the Zigbee networking protocols for sending the collected information. The Zigbee networking was preferred over Bluetooth because of its mesh functionality and hopping feature. In order to use the protocols within a Zigbee network, each sensor contains a XBee radio module. Both the Zigbee network and XBee modules use the IEEE 802.15.4 networking protocols for fast peer-to-peer networking [13]. The entirety of the distributed 10 sensors creates a wireless sensor network.

Among the 10 sensors within the Zigbee Network, a chosen sensor behaves as the coordinator. The coordinator is located at the base station of the wireless sensor network and is connected to a computer. The operation of the coordinator is responsible for selecting the channel, PAN ID, and adjusting other components of the network. The coordinator collects the data gathered by each receiver and then transfers that information into a fusion center. After data
fusion is completed, the processed data is then evaluated by a detector. The detector decides if there are unauthorized transmissions in the area. The design of the detector is constructed in Python and requires an established threshold. The threshold is based on the power within the frequency band. Detection of the unauthorized transmissions occur when the data processed by the Python code determines that it is above the threshold. The program to simulate the important locations within the project is to be constantly running on an external monitor. If the detector indicates any unauthorized transmissions, the program updates the user on the essential information.

A computer program is created to display the results of the project and demonstrate important features of the project. The position of each sensor is displayed on a generated map of the defined area. When there are unauthorized transmissions detected, the program displays a black dot on the map to indicate the location of the unauthorized device transmitting. The program also displays the power of the unauthorized transmissions on a table next to the map. If time permits, there may be an illustration of the frequency spectrum indicating where the unauthorized transmissions would be located within the wireless sensor network.

2.1.4 Setting Up SDR

Software Defined Radio receivers require Zadig software installation onto the computer in order to operate with certain drivers available for the device. The drivers allow easy access to the SDR receiver through the command prompt of the operating system in Windows 10. The Zadig application contains generic USB drivers such as WinUSB, libusb-libusb0.sys, and libusbK. SDR receiver connects to a computer with a USB connection, the WinUSB driver allows access to the SDR through the command prompt from Windows 10. Modification of the SDR receiver is now available through the command prompt and creates a simple method of
loading raw samples measurements of the receiver. Figure 9 illustrates the process of setting up the SDR receiver.

Figure 9: SDR Receiver Setup [13]

The appropriate drivers installed onto the computer allows for the SDR receiver to be connected to a certain USB port. Once inserted into the port, the computer should recognize the device and would install additional applications associated with the SDR. To capture raw samples of the SDR receiver, it is important to use the WinUSB driver instead of the automatic applications installed with the receiver which displays only a general illustration of the monitored frequency spectrum of the device. The applications do not contain the functionalities of capturing the raw data samples of the SDR. Initial USB port from the computer used to configure the SDR must be the dedicated port for the device. Installed drivers cannot recognize the SDR receiver if the device is plugged into a different USB port from the initial configuration.

2.1.5 Testing the SDR

Access to the SDR requires certain code inputted into the command prompt. The three essential commands for the SDR through the installed drivers are:

- `rtl_test`: test the device and its functions
- `rtl_sdr`: enables saving raw samples into a file
- `rtl_tcp`: loading data, changing frequency and gain

The commands above allow access and configuration of the SDR receiver from the command prompt without preinstalled applications that are associated with the receiver. Through
the installation of the external drivers, it is now accessible to obtain and compile the raw samples of the receiver.

Additional testing of the SDR receiver requires installation of the software GQRX which is an open source software defined radio application. GQRX is powered by GNU Radio and allows simple examination of the SDR device. An FM radio transmitter is used to test the functionalities of the GQRX software and is also used as the unauthorized transmitting device while prototyping the entire project. The FM transmitter allows the GQRX software to capture the exact frequency it is transmitting in through the SDR receiver. Figure 10 displays the GQRX software detecting the FM radio transmitter at 90.1 MHz.

![GQRX Software](image.png)

**Figure 10: GQRX Software [15]**

### 2.1.6 Capturing Raw Samples of the SDR Receiver

Before capturing raw samples, a test for the SDR is required. Running the command `rtl_test` detects the SDR device and displays the related properties of the certain SDR receiver.

Figure 11 illustrates the displayed information after using the code `rtl_test` in the command prompt of the computer. Capturing raw samples of the SDR receiver requires the use of the
*rtl_sdr* command. For testing purposes, the tuned frequency of monitoring is 90.1 MHz and the sampling rate is correlated to the bandwidth of the SDR receiver. The maximum bandwidth the chosen SDR receiver is established at 2.4 MHz.

The SDR receiver is now programmed to monitor a frequency centered at 90.1 MHz with a sampling rate of 2.4 MHz. A timeframe set for the SDR receiver to collect the raw samples should be exactly ten seconds for simple examination of the data saved through the command prompt.

![Figure 11: rtl_test Command](image)

Configuring and obtaining raw samples of the SDR receiver through the command prompt is now available. Capturing raw samples using the command prompt saves the information into a binary file with a generated automatic file name into the main drive of the computer [16].
2.1.7 Creating the Relationship between SDR and Python

Renaming the file containing the captured raw samples makes the process of accessing the file less complex. The command `device.read_data()` in Python allows for loading the raw sample values into a variable. The variable displays complex samples because of the SDR receiver using quadrature signals (IQ). Initial few seconds of the raw samples are not valuable due to the automatic gain associated with the SDR. Code is created to capture the first few seconds of the raw data and to store it into a dummy variable that is not used for analysis.

The `rtl_tcp` command is used to allow direct samples to be captured through Python. Running the `rtl_tcp` code in the command prompt creates a server from that location and allows Python to adjust the setting of the SDR receiver directly from the command window. The illustration below shows the startup of the SDR receiver using the command prompt. It is set it a frequency of 90.1 MHz with a sampling rate of 2.4 MHz. After setting the server using `rtl_tcp`, Python can now access the SDR receiver directly.

```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Bermeo> rtl_tcp -s 2400000 -f 90100000
Found 1 device(s).
Found Elonics E4000 tuner
Using ezcap USB 2.0 DVB-T/DAB/FM dongle
Tuned to 90100000 Hz.
listening...
Use the device argument 'rtl_tcp=127.0.0.1:1234' in OsmoSDR (gr-osmosdr)
source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
```

Figure 12: rtl_tcp Setup
2.1.8 Displaying the Collected Raw Samples in Python

The SDR receiver is configured to capture raw samples around the frequency of the unauthorized test device found using the GQRX software. The transmitting frequency of the FM radio device is at 90.1 MHz. Accessing the file in Python would require loading the file into a variable and assigning a sequence of byte I and Q values into a complex vector matrix. Code is created to display a frequency spectrum of the capture raw sample values which is illustrated in Figure 13. This concludes the structure of the relationship between the SDR receiver and Python.

Figure 13: Frequency Spectrum

2.1.9 SDR and Raspberry Pi Configuration

To establish a wireless sensor network, each of the 10 SDR receivers are required to operate without direct connection to a computer. This requires the functionalities of a minicomputer to support the SDR receiver as a wireless device. Raspberry Pi computing capabilities are appropriate for the operations of the SDR receiver. Each of the wireless SDR receivers are connected to a Raspberry Pi and the drivers to utilize the functionalities of the SDR are required to be installed onto the minicomputer. Installing the WinUSB driver onto the Raspberry Pi requires programming from a computer through USB connection. The operating
system Linux allows for configuration and installing of the WinUSB driver onto the Raspberry Pi.

The installation of the appropriate drivers must first be completed before inserting the SDR onto the Raspberry Pi. Without the drivers, the minicomputer cannot recognize the SDR device connected. When the process of installing the drivers is successful, the Raspberry Pi can utilize the same commands used to access the SDR through a computer's command prompt. No additional wires or components are required for the physical connection between the Raspberry Pi and SDR receiver because of the USB-C port already fused into the board. Figure 14 illustrates the connection between the two devices.

![Raspberry Pi and SDR Receiver](fritzing)

**Figure 14: Raspberry Pi and SDR Receiver**

### 2.1.10 Configuration the XBee Modules

A Zigbee network is used for wireless communication between each sensor and the module capable for this is the XBee-PRO. The Zigbee network requires configuration of which sensors behaving as the coordinator, route device, or end device [17]. Software to configure each XBee module is XCTU [10], which is a free multi-platform application for testing and setting up DIGI devices. Using the XCTU software allows for easy arrangement of each XBee module.
In order to connect the XBee module to a computer for configuration, the module must connect to a USB adapter. An alternative method of this procedure is using an Arduino Uno as an adapter for the XBee. This is a convenient method to configure each module because the Arduino board is easily accessible through the engineering department at Northern Illinois University. Purpose of using the Arduino is for easy configuration of the XBee module, in compared with using the Raspberry Pi or purchasing an adapter for this project. Figure 15 illustrates the connection of each pin arrangement required between the XBee Module and Arduino.

![Arduino and XBee Connection for Configuration](image)

**Figure 15: Arduino and XBee Connection for Configuration**

Initial test of the wireless sensor network consists of one coordinator at the base center and an end device operating wirelessly at another location. This initial test solely demonstrates the working functionalities of the Zigbee network consisting of two modules. Communication between the end device and coordinator is displayed using the XCTU software.

The entire wireless network consists of one coordinator at the base center, three routers to help reach further out devices, and six end devices that consume less power. Each XBee module is configured through the Arduino and uses the XCTU software to define the role of the module.
2.1.11 Raspberry Pi and XBee Module Configuration

The Raspberry Pi must have a disabled default serial TTY console to allow the installation of a new serial port to access the XBee module. Power corresponding to the XBee module requires a 3.3V input. To satisfy this condition, a regulator must be used at the output of the 5V pin on the Raspberry Pi.

The XBee module assigned as the end device is connected to the Raspberry Pi as illustrated in Figure 16 using a regulator circuit to obtain 3.3V into the XBee. For prototyping purposes, powering the Raspberry Pi comes from the power outlet. Testing the wireless communication between the coordinator and end device, which now is operating with a Raspberry Pi, requires writing a program onto the Pi that sends the data through the XBee module. The data is then received by the coordinator and is displayed on the computer. This demonstrates the working functionalities of the ZigBee wireless communication network.

Figure 16: Raspberry Pi and XBee Module
2.1.12 Sensor Configuration through XCTU

Each sensor is composed of a Raspberry Pi, SDR receiver, XBee module, and an external power source. The combination of each component is shown in Figure 17. This is a merger of each prior subunit involving the three main devices. Before establishing a wireless sensor, the SDR receiver must be configured to tune into the frequency 90.1 MHz by programming the Raspberry Pi it is connected to. Raw data samples collected by the SDR receiver at 90.1 MHz can now be transferred wirelessly using the XBee module. A timeframe for collecting samples is programmed in the Raspberry Pi to allow up to ten seconds of monitoring for testing purposes.

![Figure 17: Sensor Components](image)

All networking among the Zigbee network can be coded in XCTU. XCTU can designate the coordinator, router and end devices to help in the data relaying. Figure 18 shows the configuration page in XCTU to get the Zigbee network initialized. A basic block diagram is shown in Figure 19, detailing the process of transporting the collected raw sample of the SDR receiver into Python.
Figure 18: Zigbee Network Layer Configuration Screen

Figure 19: Process of Transporting the Raw Samples to Python [18]
2.1.13 Design Data Fusion in Python to Combine Two Sensors

Advancing in the previous testing of one wireless sensor, the trial of two separate wireless sensors transporting collected raw samples to the coordinator is now tested. The coordinator receives the raw samples of both sensors in separate files and is stored into the main drive on the computer. Code is then created to access each of the two binary files in Python and to store the raw sample measurements into two variables.

For the detection of the unauthorized transmission, the collected raw samples of each sensor are combined and then processed through a detector. The technique for combining the raw sample measurements is based on the probability theory because of its functionalities in random events that may be applicable to this project. Bayesian method is used for the data fusion of each collected samples which provides a methodology of consistent approach for dealing with uncertainty associated with the samples of each sensor. Figure 20 illustrates the process of fusion the two sensors into Python.

![Figure 20: Data Fusion from Two Sensors](image-url)
2.1.14 Design the Detector in Python with Two Sensors

Data fusion in Python combines the raw samples of two sensors and stores it into a variable. In order to design a system to detect if there are any unauthorized transmissions within the wireless network around a certain frequency, there first must be collected measurements of free space. This refers to collected samples of the SDR receivers when there are absolutely no transmissions operating at a frequency of 90.1 MHz. This frequency is chosen for testing purposes of the project using the FM radio device as the unauthorized transmitter. Once Python processes the samples of free space within the fusion center, the combined power of the free space serves as the threshold of the detector for the network of two sensors.

The structure of the detector incorporates the threshold determined from free space and allows for a system to alert if there are unauthorized transmissions within the wireless network based on the power of the transmitter. A test run is conducted while the unauthorized test device is transmitting to demonstrate the detector with the established threshold. After the raw samples of both sensors is processed through the data fusion and is combined into a single value, the detector then compares this value against the determined threshold. Raw samples collected by the sensors increased because of the power of the unauthorized test transmitter. This leads to a data fusion combined value greater than the threshold of free space. The detector decides that there is an unauthorized transmission within the wireless network of the two sensors.

2.1.15 Creating the Wireless Sensor Network and Using ZigBee

A ZigBee network consists of a coordinator, router, and end device. The coordinator is connected to a computer located at the base center of the network. Raw samples of each sensor are transported to the coordinator of the network. The routers can collect raw samples and behave as a messenger for transporting the information between the coordinator and end devices. Process of assisting the end devices to reach the coordinator requires more power consumption.
End devices are solely to collect the raw samples and can be placed under sleep mode to save power. The network topology for this project was designed specifically to maximize the detection outcome by using a rectangle network arrangement. A diagram of the topology of the ZigBee network can be seen in figure 21. This may allow for the highest probability of detection because of the evenly distributed sensors.

![ZigBee Network Topology](image)

**Figure 21: ZigBee Network Topology [19]**

2.1.16 Design Data Fusion in Python to Combine the Entire Wireless Sensor Network

Combining the ten sensors of the wireless network involves a similar process of when combining two sensors into the data fusion. Nine binary files are transported to the coordinator from the wireless sensor network and there is a total of ten binary files gathered at the coordinator including the sensor at the base center. Data fusion within the Bayesian methodology is modified to incorporate the ten collected raw samples of each sensor.

2.1.17 Design the Detector in Python with the Entire Wireless Sensor Network

Modifying the data fusion to incorporate the ten sensors allows for a new detected threshold to be determined. Another free space test is conducted to determine the power collected from the ten sensors without any transmissions at frequency 90.1 MHz. A new
threshold is determined after this test and is now incorporated into the detector for the entire wireless sensor network. Once a threshold has been developed from the background noise and the authorized signals, the Python code should compare the correlation between potential unauthorized signals and the developed threshold. The code also determines unauthorized transmissions through radio frequency fingerprinting. This method develops a “fingerprint” that characterizes a signal transmission and is very hard to imitate. Electronic fingerprints make it possible to identify wireless devices by their radio transmission characteristics. Every transmitter has a rise time signature when keyed for the first time. This rise time signature is caused by variations of component values during manufacturing. Once the code develops callsigns based on the rise time signature of each authorized transmission device on a given frequency spectrum, it becomes easy to detect when a new device is used within the wireless sensor network.

2.1.8 Locating the Direction of the Unauthorized Transmissions

The idea of locating the direction of the unauthorized transmissions is a possibility through means of the SDR receivers within the wireless network. Modifications of the clock of each SDR must be adjusted to be synced together to create a 10x phase coherent network. A program is created in Python to display the direction findings of the unauthorized transmissions within the wireless network.

The program within Python utilizes the variables that contain the raw samples of each sensor and are processed through a complex system that determines the highest power measured from four specific sensors. Four decisions can be made to estimate the location of the unauthorized device. Each decision corresponds to a specific sub parameter within the network topology. A program is designed to determine between the four different sub parameters based on the power collected of each SDR receivers. Figure 22 illustrates the sub parameters within the network topology.
2.1.19 Displaying the Results

The entire result of the project is displayed on a computer at the base center of the wireless sensor network. Displayed results consist of the monitored frequency spectrum, layout of the wireless sensor network, heatmap displaying the location and intensity of the unauthorized transmissions, and a notification describing the power collected through the SDR receivers. The frequency spectrum shows real-time information fused together from each of the receivers. The heatmap displays the layout of the wireless sensor network on top of the location and intensity of the unauthorized transmission in the bottom left corner of the screen. The heatmap is created by calculating a matrix based on the information received by each sensor. The contour of this matrix is then processed in Python, and then mixed with a colormap. Numerical data from each sensor is displayed in the bottom right corner in the form of a matrix, this data consists of power measurements collected by each sensor. Individual sensor frequency spectrums are displayed.
adjacent to the numerical data to give the user extra information. Figure 23 illustrates the information that may be displayed on the base center.

![Figure 23: Example of Displayed Information](image)

### 2.2 Prototype
Each of the sensors were assembled as designed; however, a breakout board housing the XBee-PRO module and three 130 Ω resistors in series was also used to lower the input current into the XBee. Figure 24 uses physical pictures to represent the layout of the sensor. Figure 25 is the fully actualized sensor in real life. Due to the COVID-19 global pandemic, the group was not able to assemble the complete sensor network. Three sensors were able to be set up to be the entire network. Having only three sensors is not ideal for lowering the probability of false alarm.
When more sensors are added to the wireless sensor network, the probability of false alarm could be reduced by utilizing a cluster technique to average sensors based on location instead of just a general average of all sensors regardless of location. To illustrate the existing sensor array with just the three sensors, Figure 26 shows the layout of the network. The cluster technique is more seamless with multiple sensors. For instance, Figure 27 shows the code for clustering while Figure 28 shows detection produced by clustering four nearby sensors in a network of eight sensors.
Figure 26: Wireless Sensor Network with Three Sensors

Figure 27: Cluster Detection Code (a)
Figure 27: Cluster Detection Code (b)

Figure 27: Cluster Detection Code (c)
Another goal of the design was to create an enclosure to protect the components from external environment. The enclosure was created in SolidWorks leaving all the components in...
the interior except for the XBee-PRO module and the SDR. Leaving the receivers on the outside minimizes signal attenuation to be the most efficiency out of the sensors.

Figure 29: Sensor Enclosure Opened

Figure 30: Sensor Enclosure Closed
3 REALISTIC CONSTRAINTS

3.1 Engineering Standards

With every design, there potentially could be standards already set in place to ensure the universal application of the product or service. With the emphasis of signal detection in the design purpose, a close examination must be made regarding signal transmission and reception standards. The proposed design would potentially need to conform to the transmission standards of the United States Federal Communications Commission (FCC), Institute of Electrical and Electronics Engineers (IEEE) standards, and the Zigbee Alliance.

The first fundamental element that must be subject to existing standards is the SDR. According to the FCC Office of Engineering and Technology Laboratory Division, a Software Defined Radio (SDR) is subject to a Permit-But-Ask (PBA) procedure prior to equipment authorization [20]. The FCC rules require that any radio, in which the software is designed or expected to be modified by a party other than the manufacturer that would affect the operating parameters of frequency range, modulation type, maximum output power or other radio frequency parameters outside the range under which the transmitter has been approved in accordance with the Commission rules, must comply with the requirements in Section 2.944 (a) and must be certified as a software defined radio. The radio transmitter is subject to FCC standards to prevent interference in the broadcast frequencies. A specific frequency must be permitted by the FCC for more widespread usage of such a design.

The next fundamental element that must be subject to existing engineering standards is the XBee module. According to the Zigbee RF Modules User Guide [13]:

The XBee RF Module complies with Part 15 of the FCC rules and regulations. To fulfill FCC Certification the OEM must comply with the following regulations:
1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product.

2. XBee RF Module may only be used with antennas that have been tested and approved for use with this module. The complete design may include an enclosure which would hide the external label as defined by the FCC rules, so, in realistic contexts, the design would include the proper markings if used or sold commercially to comply with standards. Zigbee is also governed by my IEEE 802.15.4 wireless standard [21].

Also, the following FCC notices are associated with the Zigbee devices [13]:

- The XBee and XBee-PRO RF Module have been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by Digi could void the user’s authority to operate the equipment.

- Original Equipment Manufacturers (OEMs) must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to the Part 15 of the FCC Rules.

- The RF module has been certified for remote and base radio applications. If the module is to be used for portable applications, the device must undergo SAR testing

There is also a Zigbee Alliance [22] which is the standard bearer of the open IoT. The Alliance encourages the mesh network aspects and universal language that interconnects smart objects through the medium of Zigbee 3.0.

3.2 Economic Constraints
   If the design were to be implemented globally or at least throughout the United States, it would not be feasible to implement millions of standalone sensors near every location of IoT
devices. The application may work best in urban scenarios until the interconnection and
detection becomes optimized.

3.3 Environmental Constraints
If rechargeable batteries are used in the design, the batteries cannot be disposed through
the typical medium of the landfill. The disposal must follow local, state, and federal regulations
depending on the contents of the battery cells [23].

3.4 Sustainability Constraints
Based from current research up to this point, there are no sustainability constraints that
are known as of now.

3.5 Manufacturability Constraints
A manufacturability constraint that can be foreseen is that the sensor enclosure should not
inhibit the detection of signals, so a material that prevents reflection would be preferred in the
design. If the design is modified to be smaller, device cooling could become an issue depending
on the ventilation of the enclosure.

3.6 Ethical Constraints
When dealing with sensitive information, confidentiality must be maintained and
enforced. Otherwise, the consumer must be notified of the way their data is used up front to
avoid various lawsuits regarding privacy or piracy of information.

3.7 Health and Safety Constraints
As stated by the Zigbee RF Modules User Guide [13], a separation distance of 20 cm or
more should be maintained between the antenna of this device and persons during the device
operation. The FCC also noted that exposure to very high RF intensities can result in heating of
biological tissue and an increase in body temperature which could cause damage to certain body
organs. At relatively low levels of exposure to RF radiations, the evidence for production of
harmful biological effects is ambiguous and unproven. There could be risk regarding radio
frequency exposure as more research is done on the health effects of radio communications. The devices that may be used would more than likely be low powered, so the likelihood of significant health risk would be relatively small in magnitude. With the implementation of more sensors and devices communicating with each other, one may assume that there could be more amplified effects of RF intensity and exposure that could be marginally harmful.

A battery-based solution would introduce a safety hazard to the consumer due to the slight possibility of battery leakage, explosion, or failure. A fault current could damage the sensor and leave a geographic area in lack of detection.

3.8 **Social Constraints**

Based on current research up to this point, there are no social constraints that are known as of now.

3.9 **Political Constraints**

Primarily, the political constraints would consist of abiding by the government guidelines based off of the FCC, the United States Environmental Protection Agency (EPA), and other investigative agencies like the Federal Bureau of Investigation (FBI) or the Central Intelligence Agency (CIA) if there were to be any breaches of security during any of the processes. The systems should not transmit data that could put national security at risk.
4 SAFETY ISSUES

In order to make the sensor network completely wireless, battery packs would have to be utilized to power all components. Due to the relatively high current draw and relatively large power consumption, the best choice for batteries would be lithium-ion cells. The perks of these batteries are that they are cheap, have very high energy density, and have a higher energy conversion rate than most battery types (up to 95%). The specific battery type would be an 18650 Li-Ion cell.

A standard 2000mAH 18650 cell can continuously discharge 10 amps at 3.7V per cell. This means that in order to power a Raspberry Pi along with the other components of the circuit, the battery pack would need to have a minimum of 2 cells in series to reach a high enough voltage. Multiple cells are attached in parallel to increase the battery packs capacity. Two cells in series would make a 2000mAH battery with an output voltage of 7.4V giving it a capacity of 14.8 watt-hours. Assuming the sensor node draws an average of 5 watts, a 2000mAH battery could power a sensor for 2.96 hours (2 hours and 58 minutes).

The use of lithium-ion batteries has downsides. Li-ion batteries are typically prone to combustion. Overcharging, exposure to heat, or damage to the cells can cause a catastrophic malfunction of the battery and cause it to swell and catch fire. The lithium within the batteries is an extremely volatile substance and reacts harshly with the atmosphere. A short circuit can also cause the batteries to overheat, resulting in damage and possible explosion. The use of lithium-ion batteries should only be used in a temperature range from 10 C to 55C. They should only be charged in temperatures that range from 5 C to 45 C. Ideally, the batteries should be kept at room temperature to avoid degradation.
In order to mitigate risks due to overcharging or undercharging, a battery charging circuit is utilized. The battery charging circuit monitors and controls the battery usage in order to maintain a safe voltage. In order to maintain a low budget, repurposed 18650 cells from old laptop battery packs would be used. Each cell would be closely inspected to have the same voltage and capacity as the other batteries to lower the probability of any catastrophic failures. If the voltages of the individual cells in a home-made battery pack are different, charging the batteries could cause overcharging.

In the long run, these batteries would have to be replaced, as lithium-ion batteries have a life of about 500 charges. After this point, the batteries become unreliable and have a lessened ability to hold a charge. Overall, it would be safer to replace the batteries after a long duration of usage to prevent extreme degradation or failure [24].

5 IMPACT OF ENGINEERING SOLUTIONS

The proposed project is intended to be a proof-of-concept. By detecting unauthorized transmissions in the FM band, it would be reasonable to expect that a similar technique can be used to detect transmissions in other frequency bands namely the Global Positioning System (GPS) band. The necessity for the application of detecting unauthorized transmissions covers a variety of spectrums. When it comes to Internet of Things (IoT), there are multiple interconnected and interrelated computing devices that are continuously collecting and transmitting information to ensure a cohesive user experience. With the increase of appliances and vehicles that operate with the IoT standard, security is an emerging issue with such connectivity. Sensors should not reveal the collected data to adjacent nodes, and the user should
be made aware of how the data is managed and applied. Otherwise, there would be a conflict when it comes to the confidentiality and integrity of the system [25].

6 LIFE-LONG LEARNING

During the design project, it is hoped that more information regarding data detection and processing should be learned. Networking and networking protocols would be a part of the learning process of the wireless sensor network. SDRs are becoming more inexpensive and can yield significant discoveries in the monitoring of signals and frequency spectrum. Creating a user interface would set the groundwork for future applications and designs that applies the knowledge learned from the project to other frequency bands such as the ISM band [26].

7 BUDGET AND TIMELINE

7.1 Budget

The primary contributor to price in the project is the number of sensors preferred in the system. Each of the components individually are not very expensive; however, when bought in bulk, the overall cost of the project can be high. For the design, the preferred number of sensors to be used in the detection process is ideally ten with a minimum of seven sensors that can be used. In the table containing the itemized costs for the components, the preferred number of sensors was the consideration. The grand total for the project if the minimum number of sensors is selected would be about $579.47. The grand total for the project if the ideal number of sensors is selected would be about $815.93. With there being a difference of nearly $236.46, there are a great amount of possibilities to consider whether that be selecting several sensors that could be economically justifiable with the allotted budget given. More than likely each of the components would gradually be added to the system as it is perfected for smaller numbers of sensors.
The Adafruit Software Defined Radio (SDR) Receiver [27] is by far the most important component of the sensor because of the testing of the detection of unauthorized signals in the FM band. SDR receivers are convenient due to the accessibility of the waveforms from the USB connectivity. The SDR prevents having to organically find a way to interpret the received signals.

The Zigbee is important to the design because it is the medium in which the sensors can communicate with the Fusion Center. It would be counterproductive to have a wired interconnection system between the sensors because there are not many devices that are in practice today that are exclusively connected with a wired network to protect against outside network intrusion. The Wi-Fi band for the wireless sensor network is not ideal for the expansive cross-section that is planned for the design. The Bluetooth band could be used, but there is not much information or documentation that is provided to ensure that the sensor network can behave properly.

The Raspberry Pi is important to the design because it is the liaison between the SDR receiver and the Zigbee. The Raspberry Pi converts the signals received from the receiver to be transmitted via the Zigbee. The networking protocols would be programmed within the Raspberry Pi layer, so the Raspberry Pi plays an integral role in the final design. Since each of the receivers are wireless, they must be self-powered. The battery power supply would be the more economical option because it would be powered with conventional batteries, and there should not be any difficulty maintaining that power for the receivers. The breadboard allows connection between all the components, and the 3-D printer filament would allow for a uniform enclosure for the devices to protect from any damage.
### 7.2 Timeline

In order to ensure optimum efficiency throughout the year, a significant amount of time working on testing the various components within the design is projected. More testing allows for more understanding of each of the individual components of the system which would help overall continuity in the design.

To summarize the project schedule, the first phase is the Conceptual Planning phase of the project. The conceptual phase consists of majority of the preliminary research that needs to be evaluated when moving forward with the ideal system components. The second phase is the component testing. After much research, there needs to be the actual testing of each of the components to have a more intimate look at whether each of the components behave as anticipated from the research. Third phase entails the development of the sensor network beginning with only two sensors to experiment with the communication between the devices as information is both received and transmitted. The primary goal of the third phase is to not only establish a communication standard between the sensors but to also be able to distinguish what is being sent. The third phase sets up for phase four which entails the extensive programming of the Fusion Center. During the fourth phase, more sensors are introduced to the system to simulate

---

**Table 1: Basic itemized cost for 10 sensors**

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini Breadboards (6 pack)</td>
<td>2</td>
<td>$5.87</td>
<td>$11.74</td>
</tr>
<tr>
<td>Xbee 3</td>
<td>10</td>
<td>$20.00</td>
<td>$200.00</td>
</tr>
<tr>
<td>Raspberry Pi 3</td>
<td>10</td>
<td>$35.00</td>
<td>$350.00</td>
</tr>
<tr>
<td>Adafruit SDR Receiver</td>
<td>10</td>
<td>$19.82</td>
<td>$198.20</td>
</tr>
<tr>
<td>Battery Power Supply</td>
<td>10</td>
<td>$4.00</td>
<td>$40.00</td>
</tr>
<tr>
<td>3D Printer Filament</td>
<td>1</td>
<td>$15.99</td>
<td>$15.99</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td></td>
<td></td>
<td><strong>$815.93</strong></td>
</tr>
</tbody>
</table>

**Table 2: Basic itemized cost for 7 sensors**

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mini Breadboards (6 pack)</td>
<td>2</td>
<td>$5.87</td>
<td>$11.74</td>
</tr>
<tr>
<td>Xbee 3</td>
<td>7</td>
<td>$20.00</td>
<td>$140.00</td>
</tr>
<tr>
<td>Raspberry Pi 3</td>
<td>7</td>
<td>$35.00</td>
<td>$245.00</td>
</tr>
<tr>
<td>Adafruit SDR Receiver</td>
<td>7</td>
<td>$19.82</td>
<td>$138.74</td>
</tr>
<tr>
<td>Battery Power Supply</td>
<td>7</td>
<td>$4.00</td>
<td>$28.00</td>
</tr>
<tr>
<td>3D Printer Filament</td>
<td>1</td>
<td>$15.99</td>
<td>$15.99</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td></td>
<td></td>
<td><strong>$579.47</strong></td>
</tr>
</tbody>
</table>
the multiple places in which data would be collected in order to detect unauthorized signals.

Once the Fusion Center can receive multiple signals at once, the remaining focus is on the filtering itself and the optimization of the network. A more in-depth project schedule can be found in the Gantt chart found below.
8 TEAM MEMBERS CONTRIBUTIONS TO THE PROJECT

The proposed project contains many elements of unexposed technology that the entire group had the opportunity to learn. Previous courses at Northern Illinois University have only surfaced many important elements of this project which assisted in understanding the functionalities of each component. Work was distributed in a fair manner that utilizes each of the group members specific skills to complete essential roles.

8.1 Alexander Bermeo

Each sensor required a strong understanding of the software defined radio receivers, Raspberry Pi computing, and Zigbee wireless networking protocols. The task for Alexander Bermeo was to understand the fundamentals of the sensor to a detailed extent. Collecting the raw data samples by using an SDR required an understanding of the operations of the receiver without the preexisting applications with the device.

Operating the SDR receiver required a direct connection to a Raspberry Pi for the creation of the wireless sensor network. An extensive amount of time was dedicated to understanding the fundamentals of the Raspberry Pi and its operating system Linux. The Raspberry Pi essentially replaces the direct access to a computer for the SDR and allows for the same functionalities in collecting the raw sample data and storing the information.

Transferring the raw sample data from sensor to sensor and eventually reaching the base center for data fusion required a detailed understanding of the Zigbee networking protocols. Creating reliable data transfer while utilizing the XBee module is an essential requirement of the project, making the knowledge obtained from this role a high priority.
8.2 Randall Barnes
With every engineering design and project, it is important to understand the logistics and costs. With each decision, a best value plan must be selected in order to conserve cost while increasing system reliability. The proposed design could be implemented in widespread design. Researching the many alternative devices on the market to locate the appropriate device for the project required analysis of many scenario.

A wireless sensor network must contain functionalities that allow for operating without a direct connection to a wall outlet. Developing the most appropriate power supply for a sensor that contains multiple devices required extensive research to reach the aimed lifespan of each sensor.

Developing the aimed schedule for each major component of the project required a strong understand of the operations of each component. Researching each device of the project to obtain a projected timeframe on how long each device may take when prototyping allows every member of the group to complete their assigned task with ease. The projected schedule developed allowed for extensive time to create a final product to satisfy the required specifications of our client.

8.3 Cory Packard
The optimum design requires a fusion center computer that runs a Python program. The task for Cory Packard was to understand the data fusion program, detection algorithm, and methods to display the finalized output of the fusion center. Creating the data fusion program requires an in-depth knowledge of Python in addition to a strong understanding of detection theory.

Building the detection algorithm requires the utilization of multiple different aspects of detection theory. A large amount of time was dedicated to understanding the fundamentals of
Python, along with understanding Bayesian detection concepts, in addition to detection methods presented in other scholarly articles. One method used in order to detect unauthorized transmission requires knowledge of radio frequency fingerprinting, which allows for specific signals to be categorized based on different characteristics.

The final display of information requires a clean and easy to understand user-interface. The user-interface includes detection alerts, a frequency spectrum analyzer, power and intensity statistics. The program may include direction detection or heat-map display for an exact location of the unauthorized transmission. Programming the user-interface requires an extensive understanding of computer programming and Python. These design implementations are fundamental to the overall project and are crucial for success.
Advancements in technology have overcome multiple milestones but have also created new methods of intrusion. Protecting autonomous devices may require a new form of security that could allow for a reliable and secure operation. The report introduces a system that detects unauthorized transmissions in a frequency spectrum using a wireless sensor network. Conducting the research for similar products concluded that there are currently limited implementation methods for that type of security. The main reason for that is due to the high cost of owning a system that functions identically as the project proposed.

The application of the project is approached in a fashion to allow flexibility to the user while bearing a low price. Main features of the system do not contain expensive hardware or limitations on the number of sensors within the wireless network. An inexpensive FM receiver may be adjusted to observe and collect data from a frequency channel. The chosen microprocessor should contain enough capabilities to bundle the raw data and deliver the information with the assistance of the XBee module which allows for effortless connection to the wireless network. ZigBee allows for smooth communication between the sensors while executing efficient and reliable monitoring. Programming the fusion center and detector through Python may result in accurate decisions because of the chosen Bayesian methodology.

The components described in the report are targeted to meet the minimum criteria of functionality while maintaining a low price. Budget justification contains detail descriptions on why each component is chosen and the cost associated with it. The overall cost of the project falls within the required range, which may be suitable for practical implementation for personal use or company operation.
REFERENCES


11 ACKNOWLEDGEMENTS

The process of exploring the functionalities of software defined radio receivers required a general guide on how to approach the project. Guidance from Dr. Fonseca, the projects client, assisted in the understanding of the functionalities of each component. General knowledge about certain functionalities within the project led us to create the bridge required between theory and practice. Feedback from Witenberg Santiago Rodrigues Souza assisted on the development of the documentation for this project.
12.1 Updated Specifications

**Physical:**

- Receivers: Aluminum with Fiberglass
- Minicomputers: Fiberglass with Copper Foil
- Transmitter: Fiberglass Epoxy Resin with Copper Foil
- Mainframe: Carbon Fiber
- Power Source: Plastic

**Mechanical:**

- Receivers
  - Size: 0.024 x 0.023 x 0.099 m
  - Weight: 0.192 kg
- Minicomputers
  - Size: 0.152 x 0.203 x 0.063 m
  - Weight: 0.453 kg
- Transmitter
  - Size: 0.063 x 0.079 x 0.051 m
  - Weight: 0.0156 kg
- Mainframe
  - Size: 0.324 x 0.217 x 0.015 m
  - Weight: 1.089 kg
- Power Source
  - Size: 0.057 x 0.063 x 0.016 m
  - Weight: 0.13 kg

**Electrical:**

- Receivers
  - Maximum Input Voltage: 5.5V
  - Maximum Input Current: 0.5A
Minimum Output Voltage: 2.0V
Minimum Output Current: 0.04A
Sampling Rate: 1 kHz
Wireless:
  Frequency: 170 MHz
  Range: 10 m
  Protocol: Zigbee

Minicomputers
  Maximum Input Voltage: 3.3V
  Maximum Input Current: 0.05A
  Maximum Output Voltage: 5.0V
  Maximum Output Current: 0.05A
  Sampling Rate: 10 kHz
  Wireless:
    Frequency: 16 MHz
    Range: 10 m
    Protocol: Bluetooth

Transmitter
  Maximum Input Voltage: 12.0V
  Maximum Input Current: 0.035A
  Maximum Output Voltage: 5.0V
  Maximum Output Current: 0.01A
  Sampling Rate: 10 kHz
  Wireless:
    Frequency: 76.0 – 87.0 MHz
    Range: 10 m
    Protocol: Bluetooth

Mainframe
  Maximum Input Voltage: 20.0V
  Maximum Input Current: 3.3A
Sampling Rate: 1.9 GHz
Battery Life: 54000 s

Wireless:
  Frequency: 233 MHz
  Range: 10 meters
  Protocol: Intel Dual Band 8265 Wireless AC + Bluetooth

Power Source
  Maximum Input Voltage: 6.0V
  Maximum Input Current: 0.05A
  Maximum Output Voltage: 5.0V
  Maximum Output Current: 0.20A
  Battery Life: 25200 s

Environment:

Receivers
  Storage Temperature: 298.15 K
  Operating Temperature: 298.15 K
  Operating Environment: (indoors/outdoors)

Microprocessors
  Storage Temperature: 301.34 K
  Operating Temperature: 303.18 K
  Operating Environment: (indoors/outdoors)

Transmitter
  Storage Temperature: 291.75 K
  Operating Temperature: 291.75 K
  Operating Environment: (indoors/outdoors)

Mainframe
  Storage Temperature: 313.15 K
  Operating Temperature: 321.45 K
Operating Environment: (indoors/outdoors)

Power Source

Storage Temperature: 298.15 K
Operating Temperature: 298.15 K
Operating Environment: (indoors/outdoors)

Software:

User Interfaces: keyboard, mouse
Hardware Interfaces: LG monitor, spectrum analyzer
Communication Protocols: Zigbee, USB
Features: display of Frequency spectrum, voltage, current, location of transmissions, and map of sensors

Computer Requirements:

Operating System: Microsoft Windows 10
Processor: Intel Core i5-8250 CPU @1.6 GHz
Memory: 8 GB

12.2 Purchase Requisitions and Price Quotes

<table>
<thead>
<tr>
<th>ORDER LIST</th>
<th>VENDOR</th>
<th>PART DESCRIPTION</th>
<th>SPECIFICATIONS (size, color, etc.)</th>
<th>QTY</th>
<th>UNIT PRICE</th>
<th>TOTAL PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>Nooeelec NESDR Mini USB RTL-SDR &amp; ADS-B Receiver Set*</td>
<td>4.7 x 2.8 x 1.2 inches, black, 2.08 ounces, USB to MCX</td>
<td>6</td>
<td>19.95</td>
<td>119.70</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>Raspberry Pi 2 Model B*</td>
<td>5 x 4 x 3 inches, DIMM, 5 Volts, Linux, 0.9 GHz</td>
<td>4</td>
<td>$37.49</td>
<td>149.96</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>PNY 16 Gb Elite Class 2-Pack</td>
<td>5.2 x 3.9 x 0.3 inches, 0.64 ounces</td>
<td>2</td>
<td>$9.99</td>
<td>$19.98</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>XBee 2mW Wire Antenna - Series 2C</td>
<td>6.7 x 5 x 1 inches, 0.64 ounces, blue</td>
<td>4</td>
<td>$33.95 ($Sale: $26.95)</td>
<td>$135.80</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>DEYUE 40PCs PCB Double-Sided Prototyping PCBs Circuit Boards Kit</td>
<td>10 - 2 x 8 cm, 10 - 3 x 7 cm, 8 - 4 x 6 cm, 8 - 5 x 7 cm, 4 - 7, 9 cm</td>
<td>1</td>
<td>$9.99</td>
<td>$9.99</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>Generic 2mm 10 Pin XBee Socket Header</td>
<td>3.9 x 2.7 x 0.2 inches, 0.64 ounces, metal</td>
<td>1</td>
<td>$7.49</td>
<td>$7.49</td>
<td></td>
</tr>
<tr>
<td>Amazon</td>
<td>Power Supply 3A with Switch UL 2.5A, 2A, 1.5A, 1A 4 Pack</td>
<td>7 x 6 x 2.2 inches, 12 ounces, 5V 3A</td>
<td>1</td>
<td>$29.99</td>
<td>$29.99</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grand Total:</td>
<td></td>
<td></td>
<td></td>
<td>$472.91</td>
<td></td>
</tr>
</tbody>
</table>

*The group did not receive the Raspberry Pi or the SDRs due to the COVID-19 pandemic.*