Table of Contents

Figure Index

Table List

Table of Equations

1.0 Executive Summary

In the United States, driving has become an extremely prevalent aspect of life. Even if someone doesn't drive themselves, chances are very high that they are at least a regular passenger in a vehicle. Despite how necessary travel is in the lives of Americans; it is still potentially very dangerous. Members of the Truck Smart team personally know multiple people that were in a car accident caused by a truck's blind spot. One member of the team was actually in one such accident personally. This team wanted to use Senior Design as a chance to make the roads safer for everyone on it.

The Truck Smart system was designed with safety and convenience in mind. Truck Smart consists of two parts, one hub unit and six sensor units. The sensor units will be placed around the truck in key locations that are known as blind spots, or areas that are not visible to the truck driver. While the system is on, these sensors will continuously send the sensor data to the hub unit which is seated in the cabin at the location most convenient to the driver. The hub unit shows the result of the sensor data. Blind spots with vehicles or other obstructions will light up red, alerting the driver that there is something in the way. The LED for areas that are clear will remain off. What makes Truck Smart unique is the portability of the system. The trailers hauled by truck driver are not constant. They continuously drop off and pick up new ones. For this reason, sensors cannot directly be built into the trailers. Truck Smart solves this by providing portable sensors that be installed or uninstalled in just minutes, enabling the driver to easily transfer the sensors from the old trailer to the new trailer in a very short amount of time. All the sensors are also completely wireless, eliminating the hassle of cable management.

This report documents the Truck Smart design process. It will first describe the motivation and goals for the project. It will then go into detail about specifications and requirements such as dimensions and battery life. The research chapter will include the choices made for each system part and why it was made. Important decisions include topics such as why the following technologies were chosen: ATmega258P, ZigBee, XBee, ultrasonic. Next, the paper will discuss the various constraints (economic, sustainability, etc.) and standards (IEEE, DoT) that affected the design decisions of the project. This document will then discuss, in detail, the hardware and software design of the system. This includes various schematics, block diagrams, and data flowcharts. Specifically, the hardware design section will involve power management and PCB design. The software design section will explain the logic of the code for the two different units of the system and how the wireless network was implemented. Moving forward, this document will show and explain the design of the system's housing and how its design provides environmental protection. Afterwards, the paper will precisely explain the methods of testing employed onto the system. Finally, the administrative section shows how the budget was split up and a basic schedule of the development process in the milestone section.

2.0 Project Description

The motivation and goals for the project were one of the first conditions to be satisfied. With this information settled, the function, requirements, and specifications of the system could be based on the previously mentioned information.

2.1 Motivation

Across the United States and the world, there are thousands of accidents every year due to drivers not checking their blind spots before switching lanes or making a turn. With the technology nowadays, many car manufactures have integrated blind spot detection technology in the newer vehicles. In today’s market, there is no fully developed blind spot detection system to help truck drivers reduce or eliminate their blind spots. According to a study published by the Federal Motor Carrier Safety and Administration, of all the truck accidents that occur each year, 20 000 of them happen due to blind spots or because a truck driver failed to adequately survey his or her surroundings. The goal for our Senior Design project is to create a sensor-based system that will alert truck drivers on real time whether or not there is a car, pedestrian, bicycle, or any vehicle within a close proximity of their truck. With this project, our main objective will be to reduce the amount of accidents that happen every year and help save thousands of lives with our system.

2.2 Goals & Objectives

- The system will be portable, allowing drivers to easily move it from one trailer to another when they load and unload their trailer at their destinations.

- The system will have low power consumption. The only components consuming power will be the microcontrollers attached to sensors and an LCD or LED display for the output of the sensors. With the technology nowadays, these devices are usually very low powered making this an easy feature to accomplish.

- Our system will have a low cost. Making it a lifesaving system is our main goal along with not making a huge profit from it, in the event that the system is marketed in the future. We want all truck drivers or corporations to be able to afford it while keeping them safe as well as the surrounding drivers.

- The system must be accurate. Accuracy is extremely critical for this project considering there will be lives on the line, should it fail to display the correct information. We want our system to display real-time information, at all times, whether or not there will be a car next to the truck’s trailer.

- The system must be very easy to use. Truck drivers have very busy lives and they drive nonstop for days across the country on a daily basis. We want truck drivers to be able to use this system regardless of their technological background and with minimal effort. The system will be designed as a fully automated product and will display the output of the sensors on the screen after being powered.

- The system must be safe and efficient. We do not want the system to distract drivers from looking at the road. As a result, we would like the system to either be located within the driver's line of sight with the road so he or she does not look away from the road when trying to see the outputs of the sensors, or to play a sound if there is a car in any of the blind spots.

- A possible objective would be to have the system powered using solar or wind technology. This would benefit our application by taking advantage of green technology and burning less fossil fuel.

2.3 Function

The main function of our project is to save lives. After interviewing a current truck driver, Mr. Reynold Marrero, and asking for his feedback on whether or not this would be a useful product for him, we came up to the conclusion that this would be a very practical application. During the interview, the driver stated that the line of sight when he makes a turn or switches lanes is very limited due to the huge blind spots caused by the trailer. His concern was that, many times, cars pass him at a high rate of speed and he has a hard time seeing them. In the functionality of our project, a very important consideration is adaptability. Since trailers can have different sizes we want to be able to simply plug and play the sensors into the trailers regardless of the size. As a result, our best approach would be to use a wireless communication system between the sensors and the base that way we will not have to worry about the cables being too long or too short depending on the size of the trailer.

Another function of our system would be to make the driver's experience easy, while not incorporating more stress into the lives of the drivers who work very long hours with very little sleep. If a truck-driver has to be worrying about modifying his truck or spending extra resources into making the system adaptable to his truck, then this would create a negative impact on our application. We want the system to be easily adaptable, portable from one trailer to the other, easy to use, safe, and efficient.

2.3.1 Related Work

There is currently a product on the market made by the company, "Gosher" that provides similar functionality. However, the system is only for the truck's cabin, and not for the trailer itself. In addition, the high price of the system is almost doubled by the installation cost. Our goal would be to implement this system for trailers with a friendly user interface and easy adaptability.

2.4 Specifications

- The system shall assist lane changes by warning the driver if there is a vehicle within a designated blind-spot range, as determined by the driver.
- The system shall include distance detection sensors (IR/Sonar/Other) strategically placed around the vehicle, as well as a display placed in the cabin.
- The system shall include up to eight transmitters to send the data collected by sensors.
- The system shall include one centralized receiver to collect data and control the display.
- The sensors shall be battery powered with a lifespan of at least 18 hours.
- The display shall receive power from the cigarette lighter receptacle port.
- All sensors shall connect wirelessly to the display inside the cabin in order to make the system easily switchable between vehicles.
- The sensors shall have an installation time of under 10 minutes.
- Each sensor unit shall weigh under 3 kg and have dimensions of less than 115x115 mm.
- The sensors shall be contained in a rigid, weatherproofed body to guarantee a lifespan of at least 100,000 miles.
- The in-cabin unit shall house the receiver, microcontroller, and display.
- The in-cabin unit shall weight 2 kg or less and have dimensions of 200x100 mm.
- The system shall be “plug and play” requiring no configuration or programming on the user end.

2.5 House of Quality

The house of quality, seen in Figure 1, is a design tool that enabled the Truck Smart team to create and visualize the relationships between marketing requirements and engineering requirements.

Marketing requirements are a set of needs and wants from the perspective of the potential consumer. This includes things such as cost, which consumers want as low as possible. The cost value under the marketing requirements means the cost of the product that the end user will pay. Another marketing requirement is battery life; which consumers want as high as possible. The accuracy of the system comes down to how often the Truck Smart system will accurately warn the user of blind spot detection. This value needs to be as close to a hundred percent as possible. The portability of the system depicts the installation ease. The system should be as easy to install as possible. The durability of the system is how long the product will last. The higher this is, the more likely the customer would be to buy it. Finally, usability of the product should be as simple as possible.

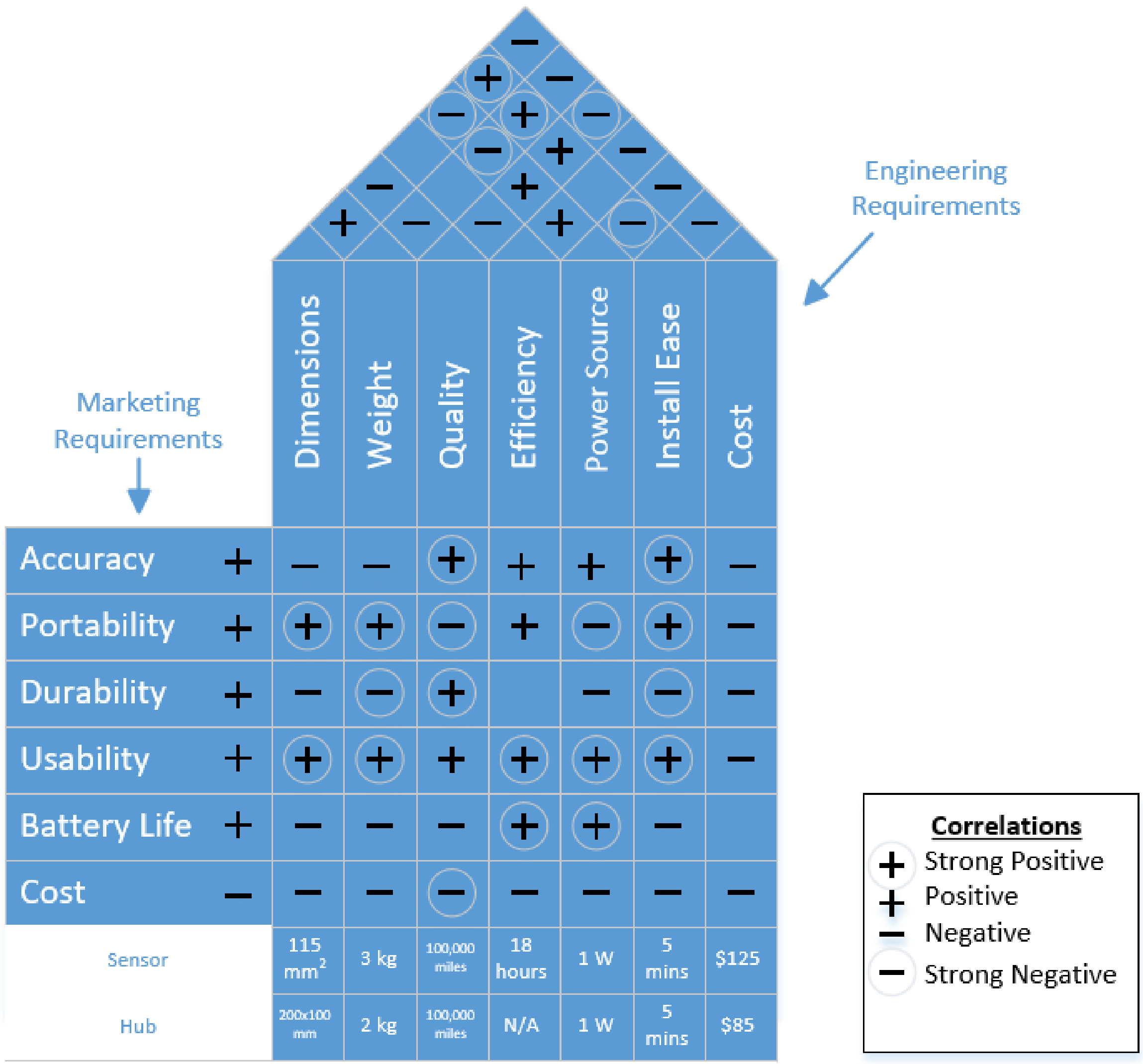


Figure 1 – House of Quality Specification

The engineering requirements once again list of the needs of the system. However, this time it is done from the perspective of the developer, or engineer, as opposed to the perspective of the consumer. The smaller the dimensions of the system, the more appealing it will be as it greatly increases the portability and usability of the system. However, this will negatively affect battery life since there will be less space to house the battery. The same logic also applies to the weight of the system. However, less weight will negatively impact the durability of the system as thinner construction material tends to be weaker. The quality of the system refers to the quality of the parts used in it. This is the difference between more expensive but longer lasting and more trustworthy brands. The efficiency of the system is based on how much power it uses to accomplish its task. Higher efficiency means much higher battery life, and therefore, better usability. The larger the power source it, the better the battery life but the lower the portability. This will also directly affect install ease.

2.6 Project Operation Manual

The Truck Smart system is a portable, interchangeable blind spot detection system with a very easy 3 step installation process shown below. The sensors are attached to the truck using magnets so no tools are required. As specified from the system requirement specifications in section 3.1, “the sensors shall have an installation time of less than 10 minutes”. To make this requirement feasible, a simple and intuitive design is necessary. “Common elements” and labels are keys to having a low installation time. On the peripheral sensors, users will need to turn them on via a small on/off switch. Once turned on, a label will be placed on the underbelly of the housing that identifies which peripheral sensor it is and where it should be placed. Magnets will be located on the underside of each sensor which will allow users to easily attach the housing to the sensor body.

The identifier on each sensor will be either a number or a color scheme. If the sensors are not placed in their correct locations, the displayed information from the hub will be inaccurate and cause a safety hazard. The numbering scheme is another simple, yet intuitive, method of sensor identification. As the driver exits the vehicle, the sensors are labeled numbers based on the order in which each peripheral sensor is encountered. Many other industries, including manufacturing and aviation, use similar simplistic interfaces in machinery instructions and checklists to make certain that users are able to comprehend each direction in the fullest sense. Without a proper understanding of the task, functionality will be inhibited by user error.

On the inside of the cabin, once the sensors are magnetically attached, the Truck Smart Hub will be provided as a “plug and play” system. The driver will plug the car port power adapter into the 12V cigarette lighter port of the vehicle and turn the Hub’s power switch to the “ON” position. The Hub’s software will then initialize and configure the system’s settings. A large portion of the start-up will be dedicated to creating a private ZigBee network, syncing with each sensor, checking for system faults, and notifying the user of the system’s status (either a successful sync or a failed sync, followed by the status’ associated flashing LED pattern). A button will be provided to re-sync the system with the peripheral sensors, in the event of a failed sync.

Within the cabin, there is an optimal placement of the Truck Smart Hub which will allow for safe, undistracted use of the system. In a study titled, “Managing In-

Vehicle Distractions", 48 drivers were presented with an array of in-vehicle tasks to complete while at the wheel. While many observations were made that were applicable to the Multiple-Resource Theory, the study demonstrated "a fundamental human performance limitation in the real-world driving context and has implications for driver response speeds when distracted". This conclusion implies that the Truck Smart system should not require the driver to carry out tasks while at the wheel. All processes, including syncing and fault reporting should be automated with no need for user intervention.

Another research study, "Effects of display position of a visual in-vehicle task on simulated driving" analyzed lane departure when compared to the eccentricity (angle of deviation) from the focal point (the road). From the behavioral data collected, researchers concluded that there is evidence of "clear detrimental effects with eccentricities of at least 35º". Therefore, a mount should be created for the Truck Smart Hub to allow the driver to maintain a low degree of eccentricity from the road.

2.6.1 Steps to Operate System:

1) Install hub unit in the cabin and connect the power supply. The system should be placed in such a way that it is easy for the driver to see while operating the vehicle. However, care should be taken that it does not draw too much attention and distract the operator from the road.

2) Place all the sensors in the corresponding spots on the outside of the truck and trailer as shown in the diagram below. Each sensor has a unique number; sensor 1 should be placed in the front, 2 and 3 on the right side of the trailer, number 4 in the back, and 5 and 6 on the left side.

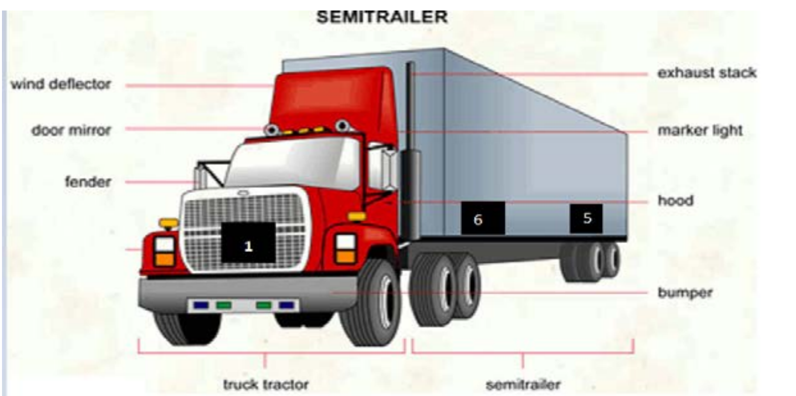


Figure 2 – Truck Sensor Placement Diagram (used with Permission from FixOnRoad)

3) Once the sensors are placed correctly, turn the hub on and wait for it to sync all six sensors. The six LEDs corresponding to each of the six sensors will blink red when the connection is successful.

3.0 Project Research

The first step in designing the Truck Smart system is to research every aspect of it. This includes the various hardware components it will use such as the microcontroller, transceiver, sensor and PCB. It also explores the various technologies that will be used such as the wireless networking and sensor type.

3.1 Microcontroller

The Truck Smart system will be using one microcontroller for the hub and one for each of its sensors. The MCUs (microcontroller units) are a key aspect of this system. The MCU in each sensor will allow for data transmission between the sensor and the XBee unit connected to it. The data will then be sent to the base unit where its MCU will process the data and output the results accordingly.

For the sake of simplicity, the same MCU will be used in each of the sensors and the hub. From an engineering standpoint, this will make both creating and maintaining the system a lot easier. Only one type of MCU will need to be understood and any worry of incompatibilities will be avoided. From a marketing standpoint, having only a single type of MCU will make purchasing hardware parts in bulk number easier. This will lower the cost to build the system which, in turn, will result in a more economically priced product. To work well for both the hub unit and the sensor unit, the chosen MCU should have a low-power functionality for simple tasks such as transferring data from the sensor to the hub and the performance necessary to process the information gathered by the sensors and output it to the user in various ways.

3.1.1 Microcontroller Options

There are a wide variety of microcontroller selections. The Truck Smart team's first choice was a TI (Texas Instruments) microcontroller due to the amount of advertising done by them on the campus and previous experience gained in various classes. The other options came from extremely popular developer boards such as the Raspberry Pi, which uses a Broadcom microcontroller, and the Arduino Uno, which uses an Atmel microcontroller. Research was done between all three brands among various price points and performance levels.

3.1.1.1 TI MSP43x Series

The MSP43x series is a family of microcontrollers created by Texas Instruments. These microcontrollers were among the first to be considered by the team due to the ample amount of prior experience. These chips also tend to have ultra-low power consumption along with a relatively low price tag. Moreover, not only does UCF host various informational seminars performed by Texas Instruments, UCF's Text Instruments Innovation Lab is staffed by professional advisors who are available to answer any questions five days a week. The MSP430 series is heavily marketed by Texas Instruments and so they provide a plethora of development resources in addition to an official community supported forum. Lack of support is not something to worry about with this series.

3.1.1.1.1 MSP430FG4618

This particular model, the MSP430FG4618, was chosen to be researched because every Computer Engineering and Electrical Engineering student at UCF has experience with it due to the Embedded Systems class which is part of the core curriculum. This MCU costs $17.54 per unit.

The MSP430FG4618 prioritizes extremely low power consumption. Running in active mode at 1 megahertz and 2.2 volts, this MCU consumed only 400 micro amps. This device supports 5 different power-saving modes. More importantly, it has a wakeup time from standby mode of only 6 microseconds. An interesting feature to note on this MCU is that it has a built in segment liquid crystal display driver. This would greatly simplify printing out simple status messages such as a low battery warning. This MCU has a total of 80 I/O pins. It also has a 12-bit digital-to-analog converter with synchronization.

Between the high pin count, integrated LCD driver, and various other features, this is very clearly a feature rich microcontroller. Nevertheless, there is a matching price tag to go along with all those features. While this is a very capable MCU, it is just too much for the relatively simple tasks of pulling data from a sensor and transmitting it. This microcontroller is no longer under consideration.

3.1.1.1.2 MSP430G2 LaunchPad

The LaunchPad is a kit designed to have everything needed for a developer to get up and running. This kit features another microcontroller that most UCF engineers are familiar with due to its use in the required class Engineering Analysis and Computation. The LaunchPad kit actually comes with two different microcontrollers and the LaunchPad Development board (MSP-EXP430G). This development board included an integrated emulator interface that allows real time programming and debugging through a USB. The most interesting aspect of the LaunchPad was found in the descriptions of the two microcontrollers it includes: “Typical applications include low-cost sensor systems that capture analog signals, convert them to digital values, and then process the data for display or for transmission to a host system.” This application is almost word for word exactly the goal of the Truck Smart system.

3.1.1.1.2.1 MSP430G2553

This is the first microcontroller included in the LaunchPad. It includes 16kB of flash memory and 512B of RAM. It is part of the ultra-low-power family and uses 230 micro amps at 1 megahertz in active mode. This MCU has a total of only 20 pins, the lowest of any microcontroller researched.

3.1.1.1.2.2 MSP430G2452

This is the second microcontroller included in the LaunchPad. It is very similar to the first, the main differences are that it has exactly has as much memory. More specifically, it includes 8kB of flash memory and 256B of RAM. Same as the G2553, this MCU also has a total of only 20 pins.

Since the only difference between the two microcontrollers included in the LaunchPad is memory size, only the G2553, the MCU with the greater memory, will be considered in future comparisons.

3.1.1.1.3 MSP432P401R

This microcontroller was selected to be researched after a team member received one at no cost while attending a Texas Instruments seminar aimed at Senior Design students. It is a low-power, high performance microcontroller. It uses the 32-bit ARM Cortex-M4 processor which has a frequency of up to 48 megahertz.

When it comes to power consumption the MSP432P401R is extremely efficient, using only 80 micro amps per megahertz. It also features a total of 84 GPIO pins, giving it a large amount of possible connections.

3.1.1.2 Broadcom 2835

The Broadcom 2835 is the MCU that lies at the heart of the Raspberry Pi, an extremely popular single-board computer. Initially it was thought that since the Raspberry Pi was so popular and successful in the world of development, its MCU would have support for developers and vice versa. However, it was found that the Broadcom 2835 would be impossible to buy in quantities less than hundreds of thousands. The Raspberry Pi Zero, which uses the 2835, could be bought and used for only $5.00 each but since it is essentially a pre-built computer it would not meet the requirements of Senior Design. Consequently, this unit was not considered any further.

3.1.1.3 Atmel megaAVR Series

The Atmel megaAVR microcontroller series offer high performance for extremely low power consumption using Atmel's picoPower technology. Atmel's products tend to also be the cheapest out of all the researched products which will lead to an affordable final product. However, the largest attraction of these boards lie in its popularity. Atmel's chips are used in various Arduino products, leading to an absurdly large amount of community support.

3.1.1.3.1 ATmega328P

The ATmega328P is a low-power microcontroller created by Atmel. It can be purchased at only $1.88 per unit from Atmel's website. This MCU used in the extremely prominent Arduino Uno. Thanks to this, there is a severe abundance of various tutorials and resources on the ATmega328P/Arduino Uno. Due to the popularity of this MCU, it can be trusted to be both reliable and relatively simple to program.

Given the operating conditions of 1 megahertz, 1.7 volts, and 25 degrees Celsius, the power consumption of the ATmega328P in active mode is 0.2 milliamps. One thing to note, however, is that this MCU has the special feature of supporting six sleep modes, including the power-down and power-save modes. The power consumption in these modes under the previously mentioned conditions are, respectively, 0.1 micro amps and 0.75 micro amps.

The ATmega328P has a total of 28 pins. There are 15 digital and 8 analog pins, for a total of 23 I/O pins. The analog pins can also be used as digital pins if necessary. For the sensor unit, these pins will be used by the XBee module and the ultrasonic sensor. For the hub unit, these pins will be used by the XBee module and the display. This will be more than enough of the pins needed for the Truck Smart components.

3.1.1.3.1.1 Arduino Uno

The Uno is a programmable circuit board used by numerous hobbyists. An extremely important consideration to take into account is that it is possible to burn the Uno's bootloader onto the standalone ATmega328P. It is also possible to purchase an ATmega328P with a pre-loaded Arduino bootloader. These are available for $5.50 per unit through the SparkFun website. Regardless of the method used, this means that any code made for the Arduino Uno will be fully compatible with the ATmega328P. This will allow the team to use an Arduino Uno for testing purposes while waiting on the PCB design. Afterwards, once the PCB is ready, the team will be able to migrate from the Arduino Uno to the ATmega328P without too much hassle.

Another attractive feature of the Arduino Uno is the existence of the XBee shield built specifically for the Uno. This will allow the team to easily connect the XBee transceiver to the Uno microcontroller while testing. The shield acts as an intermediary between the Uno and the XBee, allowing the two to communicate while still preserving all GPIOs.

3.1.1.3.2 ATmega2560

The ATmega2560 MCU is a bigger, stronger version of the ATmega328P. Starting at $10.45 per unit, it is more than five times the price of the 328P. In return for the price increase, the 2560 boasts more memory, higher performance, and more pins. Similar to how the 328P is used in the Arduino Uno, the 2560 is used in the Arduino Mega. The bootloader can even be burned in the same method. Therefore, there is no shortage of resources when it comes to learning how to use the 2560.

Despite being a larger and more powerful MCU, the 2560 actually has a lower power consumption than the 328P. According to the datasheet, under the conditions of 1 megahertz and 1.8 volts the power draw is only 500 micro amps. This MCU also supports the same power saving modes as the ATmega328P, making it highly competitive.

The ATmega2560's biggest feature is the amount of pins that it possesses. The ATmega2560 has a total of 86 programmable I/O lines. Of these, there are 54 digital I/O pins and 16 analog input pins. This amount of pins would allow for a multitude of add-ons for each unit.

Despite all these features, it was decided that this microcontroller would be overkill for the purpose of this project. The additional pins, while handy, would not be used in the current scope of the project. The additional memory capacity would remain unused as well. Furthermore, the larger form that comes with the additional performance would be a hindrance while designing the housing for both the sensor unit and the hub unit. For the aforementioned reasons, the ATmega2560 will not be considered any further.

3.1.2 Microcontroller Comparisons

An initial count of 8 microcontrollers were narrowed down to the top three potential choices. MCUs were eliminated based on factors such as price, user popularity and amount of reference material available, and performance. The performance category not only eliminated MCUs that were too weak, but also ones that were too powerful and unnecessary for the task at hand. Below are more specific comparisons between the top three choices.

3.1.2.1 Power Consumption

Battery life is a major concern of the Truck Smart system. Truck drivers tend to make long cross-country trips. The drivers should not have to worry about the sensors running out of power before they reach their destination. According to the Federal Motor Carrier Safety Administration, drivers are limited to 11 hours of driving per day. Nonetheless, the goal of the Truck Smart system is to have the sensors last at about 18 hours to account for any supplementary driving. To reach this goal, the power consumptions of the microcontroller should be as low as possible while still having enough performance to perform the tasks necessary.

Table 1 – Microcontroller Power Consumption Comparison

Microcontroller	Lowest Operating Voltage (V)	DC Current at 1 MHz (mA)	Power Consumption (mW)
MSP430G2553	1.80 V	0.23 mA	0.414 mW
MSP432P401R	1.62 V	0.08 mA	0.129 mW
ATmega328P	1.80 V	0.20 mA	0.360 mW

The power consumption seen in Table 1 was calculated by multiplying the operating voltage and the DC current. In an impressive showing, the microcontroller with the most performance also boasts the least power usage. The MSP432 is positively in a league of its own when it comes to low power consumption. On the other hand, the MSP430 and the 328P are extremely similar, bearing the same value for the lowest operating voltage. Despite that, the ATmega328P can be said to use about 15% less power. While the MSP432 is the clear winner in this category, the MSP430 can be said to be the clear loser.

3.1.2.2 Cost

System cost is an extremely important aspect of this project. A potential lifesaving system such as Truck Smart needs to be used by as many people as possible. One of the easiest ways to do that is to make the final product affordable. That can be done by keeping the production cost low.

Table 2 – Microcontroller Cost Comparison

Microcontroller	Unit Price ($)	System Price (x8)	System Price Increase (%)
MSP430G2553	$2.38	$19.04	0%
MSP432P401R	$6.20	$49.60	160.50%
ATmega328P	$3.30	$26.40	38.66%

Table 2 above shows that the MSP430G2553 is the cheapest option. However, while important, price is not the only factor. The ATmega328P is only about 39% more expensive. This is a small enough increase in price that the MCU might still be worth it based on other factors. On the other hand, the MSP432P401R is more than twice as expensive, actually being closer to three times as expensive. It would be hard to justify such a large increase in price.

3.1.2.3 Memory Size

When it comes to microcontrollers, memory can be split into 3 types. The first type is flash. Flash memory is non-volatile. That means it keeps the stored data even after a power cycle (turning off and on). Flash memory is where a microcontroller's programming is stored. The more flash memory that is available, the more complicated the code can be. The second type of memory is Electrically Erasable Programmable Read-Only Memory (EEPROM). This type of memory performs the same function as flash memory but is slower. Flash actually falls under the category of EEPROM but is typically referred to as flash to differentiate from non-flash EEPROM which tend to be a lot slower. The EEPROM capacity should not matter in the Truck Smart system and therefore will not be compared. The third type of memory is RAM. It is a volatile memory, meaning that any data stored in it is lost once power is turned off. It is used to store temporary data, such as the result of a calculation. Again, as with flash memory, the more RAM there is, the more complicated programs can be.

Table 3 – Microcontroller Memory Comparison

Microcontroller	Flash Memory	RAM
MSP430G2553	16 KB	0.512 KB
MSP432P401R	256 KB	64 KB
ATmega328P	32 KB	2 KB

As expected, Table 3 above shows that the cheapest option also has the least memory. The ATmega328P has double the flash memory and four times the amount of RAM as the MSP430. The MSP432 on the other hand, is on a whole different level. The MSP432 has as much flash as the MSP's squared along with 125 times as much RAM. There is absolutely no comparison here, the MSP432 is unparalleled in terms of memory.

3.1.2.4 General-Purpose Input/Output

A general-purpose input/output (GPIO) is a pin that can transfer either data or power. A GPIO can act as either an input or an output and is specified at run time. The sensor unit of the Truck Smart system will contain both a sensor and a ZigBee transceiver module. Since both of the aforementioned attachments will be powered through the chosen microcontroller, the pins need to have high enough of an output current to power both devices. In addition to the power output, there also needs to be enough pins to transfer the data to and from the devices. The sensor will require four pins in total and needs a working current of 15mA. The ZigBee transceiver, the XBee SC2, will require four pins and an operating current of 45 mA.

Table 4 – Microcontroller GPIO Comparison

Microcontroller	GPIO Pin Count	Max Output Current per Port (mA)
MSP430G2553	16	48 mA
MSP432P401R	48	100 mA
ATmega328P	23	100 mA

As seen in Table 4, the MSP432P401R is once again the clear winner in terms of performance. It offers three times as many GPIOs as the MSP430G2553 and more than two times as the ATmega328P. While this is a great feature, it is a bit too much for the purposes of the Truck Smart system which only requires a minimum of 8 pins. Between the ATmega328P and the MSP430G2553m the pin count is relatively similar with only a difference of seven pins. The important thing to note between these two MCUs is the difference in power capability. The ATmega328P can sink or source up to 100mA per port whereas the other can only manage 48 mA. This is a huge deal breaker as 48 mA would only be enough to power either the sensor or the transceiver, not both. Meanwhile, 100 mA offered by the ATnega328P is more than enough to power both.

3.1.2.5 Clock Frequency

The clock frequency is the rate at which the microcontroller executes an instruction. Simply speaking, the faster the clock frequency is, the faster the CPU can perform tasks. Users will rely on the Truck Smart system to relay real-time information from the sensor to the display in the hub unit concerning blind spots. The system cannot afford to spend extra time processing information before getting it to the user so the clock frequency is an important factor to consider.

Table 5 – Microcontroller Clock Frequency Comparison

Microcontroller	Clock Frequency (MHz)
MSP430G2553	16 MHz
MSP432P401R	48 MHz
ATmega328P	20 MHz

Table 5 shows that the MSP432P401R clocks at a frequency three times as fast as the MSP430G2553. What this means is that in the time it takes for the MSP432P401R to perform 48,000,000 clock cycles, the MSP430G2553 will only have performed 16,000,000. That is a huge difference of 32,000,000 clock cycles. The ATmega328P once again lies somewhere in between at 20 megahertz. In terms of pure performance, the MSP432P401R is the best followed by the ATmega238P.

3.1.3 Microcontroller Choice: ATmega328P

The previously discussed comparisons of the three microcontrollers show a very clear trend. The MSP432P401R is a high performance unit. It heavily and decidedly towers over the other two microcontrollers in all areas of performance. It has more than eight times as much memory and 32 times as much RAM as compared to the values of the next highest microcontroller. The MSP432 also has twice as many GPIO pins and a more than twice as fast clock rate. Most importantly, even with all these higher performance values, the MSP432 is also the most efficient unit when it comes to power consumption. For all that, as far as the performance requirements of the Truck Smart system goes, the extreme performance of this microcontroller is unnecessary. Many of the pins and much of the memory will go unused. The high performance of this MCU can unfortunately also be seen in its price. It is almost three times as expensive as the cheapest MCU. Due to the high price and overloaded feature set, the MSP432P401R is no longer considered for use in the Truck Smart system.

The dismissal of the MSP432P401R leaves two microcontrollers to choose between, the MSP430G2553 and the ATmega328P. These MCUs seem in be extremely similar in most areas. Both are low budget chips but the ATmega328P outperforms the MSP430G2553 in every single category except for price. The AVR MCU has twice as much memory, four times as much RAM, faster clock frequency, and higher GPIO count. The GPIO current output is a staunchly limiting factor of the MSP430 as it is not able to power both of the attachments that are necessary for the Truck Smart system. On the other hand, the AVR MCU has more than enough to power both. As far as cost is concerned, the ATmega328P is only about 40% more expensive than the MSP340. While 40% seems like a lot, the actual difference is only around a single dollar per microcontroller. This is a very small increase in price for a very large performance gain.

Another factor to consider is the availability of resources for the Truck Smart team to consult. While Texas Instruments does provide a popular forum to discuss their products, it is trumped by the fact that the ATmega328P microcontroller is used in the Arduino Uno. This fact makes it a much more popular choice among other developers which leads to a greater availability of resources.

For the formerly examined reasons, the team has decided to use the ATmega328P in the Truck Smart system. It provides the best balance of power, efficiency, price, and performance among all of the researched microcontroller units.

3.2 Wireless Communication

The Truck Smart system requires a means of communication to allow for easy and practical use. The following sections will discuss each type of communication technology that was considered, the pros and cons of the technologies, and an in-depth explanation of how the chosen technology operates. Having a low-level description of the communication interface's inner workings will enable optimization of the Truck Smart system design. These details will also be useful during the system integration phase.

3.2.1 Why use wireless technology?

The main goal when designing the Truck Smart system is that it must be adaptable and easy to plug and play into different truck trailers regardless of the size of the trailer. Truck trailers can be anywhere from 20 to 53 feet; therefore, the location of the sensors will vary depending on the size of the trailer; making it almost impossible to have a plug and play system that operates through wires. The system will be using anywhere from 6 to 8 sensors (3 on the left side, 3 on the right side, and with the possibility of adding 1 in the back). However, the number of sensors will depend on the trailer's size and the sensor's range. For example, it would not be efficient to use 6 sensors for a 20 feet long trailer.

If the system used wires to connect the sending microcontrollers (houses the sensor) to a base receiver (receives the output of the sensors and displays it), it would be an ineffective application. This would be due to the possibility of wires overstretching, breaking, not being long/short enough, and other issues such as the delay time from when the signal is sent across the wire until it is received would be significantly long. To make the system easily adaptable, wireless technology will be employed to transfer the sensor's output to the base microcontroller.

3.2.2 Communication Types

There are many options available for data transmission, for both wired and wireless systems. For practicality and ease of installation, wireless communication was chosen. From this decision, a set of questions arose: What protocol is most well-suited for this application? What transmission range does each type of communication provide the system? Does this technology allow for intuitive serial communication? Will a router be a necessary component of the system? Finding answers to these questions is a task that is deemed a necessity in order to overcome the engineering challenges that face the Truck Smart System.

The technologies considered for integration into the Truck Smart system includes Bluetooth transmit/receive modules, router-based private network Wi-Fi, infrared, RF, and ZigBee modules. These types of wireless communication technology all have the capability to sync with the Truck Smart Hub to pass small data packets. The methodology for how these data packets are created and handled is the unique variable that exists between each of these technologies.

3.2.3 Technology "Pros and Cons"

In the world of technology, there is always another competing product available for any application. Throughout the many industries that depend on technology, it has become a commonality to quantify the potential usefulness of a product through a means of "pros and cons". Depending on the market in which the comparisons are made, there are certain focal points of technologies that hold more weight. The retail market, in particular, has a heavy focus on the performance specifications of new products. Many companies go head-to-head in "spec battles" where products are advertised to be "the best" in a variety of categories, which include titles like "fastest processor", "highest mega-pixel camera", and "sharpest screen image". Many customers take these claims to heart and they will use earlier models of a product as a specification baseline for comparison before making a purchase. However, many of these assertions act as an attempt to take a majority share of the consumer market and may hold little weight.

3.2.3.1 Bluetooth Transmit/Receive Modules

Bluetooth technology operates on a "Personal Area Network", also known as PAN. PAN networks meet the requirements in terms of data transmission and transmission rates. However, the Bluetooth options available to integrate into the Truck Smart system offer a limited range of up to 40 feet for data transmission. With the maximum sensor range estimated at 63 feet, investing in Bluetooth technology opens the Truck Smart system to potential data transmission faults in the system. Some kinds of faults may include dropped data packets, device sync failures, and high battery consumption rates.

3.2.3.2 Router-based Private Network Wi-Fi

The well-known technology of Wi-Fi continues to be incorporated into more and more consumer electronic devices every year. While Wi-Fi networks have many advantages, including ranges up to 300 feet outdoors (according to speedguide.net), the lack of a practical means of implementing a mobile Local-Area-Network (LAN) adds much more time and effort to the Truck Smart installation procedure. Another disadvantage is the added cost of including a Wi-Fi router into the critical system components. Wi-Fi routers are also notorious for high power consumption and may inhibit the use of the cigarette lighter port as a power source.

3.2.3.3 Infrared Communication

Infrared (IR) communication is a technology that depends on light in order to operate. The infrared wavelength spectrum ranges between 750 nm to 1 mm. The lower portion of the infrared band borders the visible red spectrum, hence the name “infrared”. The infrared spectrum is unable to be seen by the human eye, and does not impact many (if any) mobile communication systems. These characteristics make IR communication much less technologically intrusive than its frequency based counterparts. However, the downside to light-based technology is the need for a direct line-of-sight connection between the system’s transmitter (LED) and receiver. The Truck Smart system’s peripheral sensors will be placed in a Tractor Trailer’s six major blind spot areas. Since these areas are known “blind spots”, it would be an extremely difficult task (possibly requiring mirrors) to implement line-of-sight IR technology as a means of communication between the system’s peripheral sensors and the Truck Smart Hub (located inside the truck cabin). Therefore, IR has been deemed an impractical technology to be used for Truck Smart data transmissions.

3.2.3.4 Radio Frequency

Many technologies harness the power of Radio Frequency (RF) to communicate with devices near and far. RF has the benefit of being very versatile in the design aspects and can be integrated into a system without much hassle. Ranges for RF can reach extremely long distances (at lower frequencies). RF can also propagate through a variety of different mediums including air, water, solids, and even space. Table 6, provided by National Instruments, shows a list of the applications for certain bands of RF and the band designations that reside within each range (as deemed by the United States National Telecommunications and Information Administration 2016).

This frequency allocation table is based off the Frequency Allocation Radio Spectrum chart, which designates the bands that are government exclusive, non-government exclusive, and shared government/non-government frequencies. As the frequencies increase, the applicable distance decreases. This characteristic of RF is due to the attenuation of radio waves into the environment. As the mass (in terms of particle movement) of propagation decreases, it becomes easier for the wave to dissipate into its surroundings.

In the event RF was to be integrated into the Truck Smart system, the frequency allocation table is critical to adhere to, in order to prevent frequency band intrusion and remain legal. The wireless Wi-Fi network discussed in the earlier sub-section operates in both the 2.39-2.4 GHz and 5 GHz LAN unlicensed spectrums.

Table 6 – Frequency Allocations and Applications

Frequencies in GHz	Allocated Purposes
0.216–0.220, 0.235–0.267, 0.4061–0.45, 0.902–0.928, 0.960–1.215, 1.215–2.229, 2.320–2.345, 2.360–2.390, 2.7–3.1, 3.1–3.7, 5.0–5.47, 5.6–5.925, 8.5–10, 10.0–10.45, 10.5–10.55, 13.25–13.75, 14–14.2, 15.4–16.6, 17.2–17.7, 24.05–24.45, 33.4–36, 45–46.9, 59–64, 66–71, 76–77, 92–100	Radar, all types
2.390–2.400	LANs (unlicensed)
2.40–2.4835	Microwave ovens
45.5–46.9, 76–77, 95–100, 134–142	Vehicle, anticollision, navigation
10.5–10.55, 24.05–24.25	Police speed radar
0.902–0.928, 2.4–2.5, 5.85–5.925	Radio frequency identification (RFID)
3.7–4.2, 11.7–12.2, 14.2–14.5, 17.7–18.8, 27.5–29.1, 29.25–30, 40.5–41.5, 49.2–50.2	Geostationary satellites with fixed earth receivers
1.610–1626.5, 2.4835–2.5, 5.091–5.25, 6.7–7.075, 15.43–15.63	Nongeostationary satellites, mobile receivers (big LEO, global phones)
0.04066–0.0407, 902–928, 2450–2500, 5.725–5.875, 24–24.25, 59–59.9, 60–64, 71.5–72, 103.5–104, 116.5–117, 122–123, 126.5–127, 152.5–153, 244–246	Unlicensed industrial, scientific, and medical communication devices
3.3–3.5, 5.65–5.925, 10–10.5, 24–24.25, 47–47.2	Amateur radio
6.425–6.525, 12.7–13.25, 19.26–19.7, 31–31.3	Cable television relay
27.5–29.5	Local multipoint TV distribution
12.2–12.7, 24.75–25.05, 25.05–25.25	Direct broadcast TV (from satellites)
0.928–0.929, 0.932–0.935, 0.941–0.960, 1.850–1.990, 2.11–2.20, 2.450–2.690, 3.7–4.2, 5.925–6.875, 10.55–10.68, 10.7–13.25, 14.2–14.4, 17.7–19.7, 21.2–23.6, 27.55–29.5, 31–31.3, 38.6–40	Fixed microwave (public and private)

However, the integration of general RF into the Truck Smart system is not as straight forward as one might believe. The use of RF as a data transmitter requires a protocol that allows both the transmitter and receiver to “agree” with a designated standard and encrypt/decrypt messages through this standard.

The integration of general RF as a method of data transmission would require designing a protocol specific to the Truck Smart system. An RF signal cannot carry data without being modified. Wave modification in the RF world is called “modulation”. Once the wave is modulated in a standardized fashion, the hub would have a component to demodulate a received waveform, then a "software overlay" would decrypt the received message based on the protocol. Designing a proprietary protocol for the system is an unnecessary task since are many protocols already available and integrated into the transmit/receive technology (such as Wi-Fi, Bluetooth, and ZigBee). This leads to the next discussion regarding a more practical option that takes advantage of RF technology in an efficient and scalable manner.

3.2.3.5 ZigBee Wireless Technology (Selection)

The introduction of ZigBee wireless technology into the consumer electronics market, wireless communications are more advanced than ever before. ZigBee technology has introduced the new concept of "Digimesh", which creates a network (similar to the one shown in Figure 3) that has no single points of failure. Devices using ZigBee as their "Wireless Language" have interoperability to their advantage, removing the need for router-based networking. Each device has the capability to route itself to another local device.

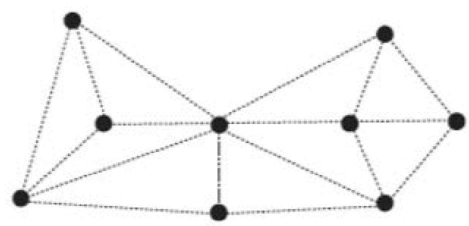


Figure 3 – Digimesh Network

The ZigBee component available to be used for the Truck Smart system is the XBee module. This low-power module is able to be integrated into microcontrollers seamlessly and combines RF technology (discussed in the previous section) with the ZigBee standardized protocol. Having a pre-existing protocol available means the Truck Smart system design does not have to account for proprietary RF communication development. The Digimesh capability of the ZigBee devices means there is no need for a router to be included in the system design. The XBee module available has an RF power output of 1 mW. The component itself consumes approximately 165 mW. The output power gives the XBee module a transmit/receive range of over 100 feet, which exceeds the requirement of 63 feet for the Truck Smart system. There is a wide range of available ZigBee technologies available for integration, all of which are small in size and do not require many special provisions.

The downside to the available XBee module is the low data transmission rate. According to Digi International, data transmission rates are "dictated by the system". The relation is noted that "Higher data rates allow the communication to take place in less time, potentially using more power to transmit". Since the XBee module has a very low output, it can be expected that the transmission rate follows suit in a negative linear relationship. Luckily, the data to be passed from the peripheral sensor to the Truck Smart hub will be in the form of small packets containing basic information such as strings of integers and system unique identification string hashes. The maximum amount of data to be included in these strings will be the same number of as the maximum addressable byte in the SRAM of the Truck Smart system (dependent upon the ATmega32P chip). There are 2 kilobytes of SRAM (or 2048 bytes). At 1 byte/character, the maximum passed string length cannot exceed 2048 bytes (2kB). The maximum data transmission rate provided by the 1 mW XBee modules happens to be 250 kbps. When calculating the time necessary for data transmission of a 2 kB string of data at 250 kbps, the transfer time is approximately 0.0625 seconds. Even with system overhead, this transmission time is very reasonable for the Truck Smart system since there are no large pieces of data being passed to, or from, the hub.

3.2.4 Inner-workings of Communication

The premise for any RF communications technology is the capability to produce electromagnetic waves in a periodic and accurate fashion. The theory of electromagnetic fields assumes that (in air) these waves travel at speeds near the speed of light. Knowing this speed, defined by the constant $c = 3.00 * 10^8\, m/s$, both wavelength and frequency are inversely dependent upon each other. The well-known equation describing this relationship is $\lambda = \frac{c}{f}$ where λ is the wave's length and f is the wave's frequency. These wave parameters are used during the design phase of an RF system. The circuitry must be biased to meet the necessary frequency requirements and standards, and the wavelength must be known to calculate the design length for the antenna. Without a proper length of antenna, the waveform will not modulate through the medium (air) effectively when transmitting, and a receiving antenna will not detect the waveform. When integrating ZigBee technology into the Truck Smart system, the proper antenna must be used to ensure a functional transmit/receive relationship between the peripheral sensors and hub.

3.2.4.1 Antenna Types and Applications

There are many types of antennas, each type having a different set of applications. According to elprocus.com, an antenna type is defined by the following characteristics; Antenna Gain, aperture, directivity, bandwidth, polarization, effective length, and the polar diagram. As one can assume, the more gain an antenna has, the more effective it is. To design directional antennas, gain must be distributed to certain vectors of the antenna to produce the desired pattern of radiation. The radiation pattern that is formed defines the directivity of the antenna. There are many variations of unique directional antennas, some of which include dipole, short dipole, monopole, loop, and many more. These given types are derived from a single family (wire antennas) of many other antenna families, which include log periodic, travelling wave, microwave, and reflector antennas. The aperture of an antenna is what makes it effective and efficient. The aperture is strictly dependent on a known input/output wavelength.

While there are many antenna types, families, and characteristics (well into the hundreds), to stay within the realm pertaining to the Truck Smart system, the discussion will focus on the basic wire antenna family of the monopole type.

As a rule, the monopole antenna must be mounted upon a ground plane. Unlike the concept of grounding in typical electronics, an antenna ground plane acts as a reflector for electromagnetic waves. Ground planes are usually much larger than the receiving wavelength to ensure maximum reflectivity. The Truck Smart system's peripheral sensor ground plane (68.6 mm x 53.4 mm) will suffice for the low energy signal being received.

The peripheral sensor antenna length should match the wavelength to be received. However, the wavelength for a 2.4 GHz signal is 12.5 cm, which is too long and impractical to integrate into the Truck Smart antenna design. Luckily, the laws of physics allow for fractions of a wavelength to be used to capture the transmitted wave's harmonics. There are different ways to calculate antenna length depending on the type of antenna being used. The fraction of wavelength is also a critical design aspect that provides the potential to save material and space. Since an electromagnetic wave that is resonating through a medium has a finite amount of known harmonics, the laws of physics allow engineers to take advantage of the available harmonics. For the Truck Smart system, the most practical length to use will be a 32 mm length antenna, or approximately 1/4 of the original wavelength. This length will enable the antenna to receive small harmonics of the transmission, which will then be compensated for gain once the signal is processed.

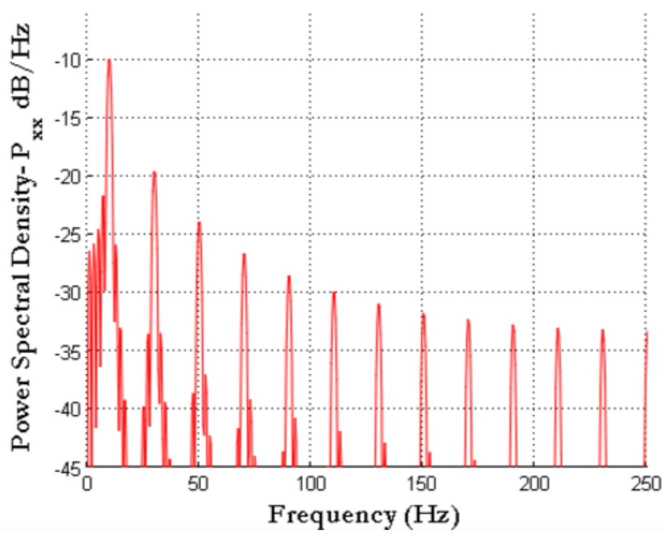


Figure 4 – Decaying Harmonics

3.2.5 Structure of the System

The use of wireless technology will allow to simply have a housing unit (contains the sensor, microcontroller, Arduino shield, and wireless transceiver) *refer to Figure 5 attached to specific point in the trailer.

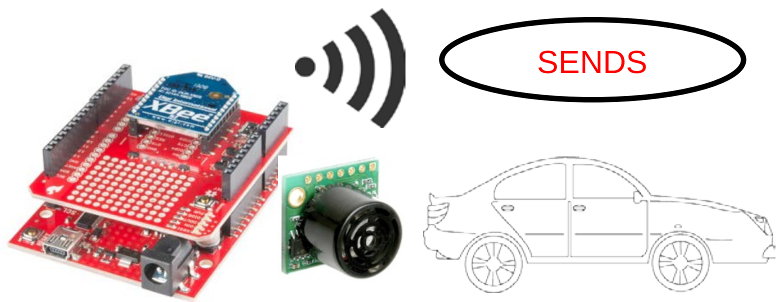


Figure 5 – Sending System Structure and Components

This unit will be responsible for capturing the output of the sensor (whether there is a vehicle, pedestrian, bicycle, etc.) next to the trailer, and send this data in real time to the base *refer to Figure 6*.

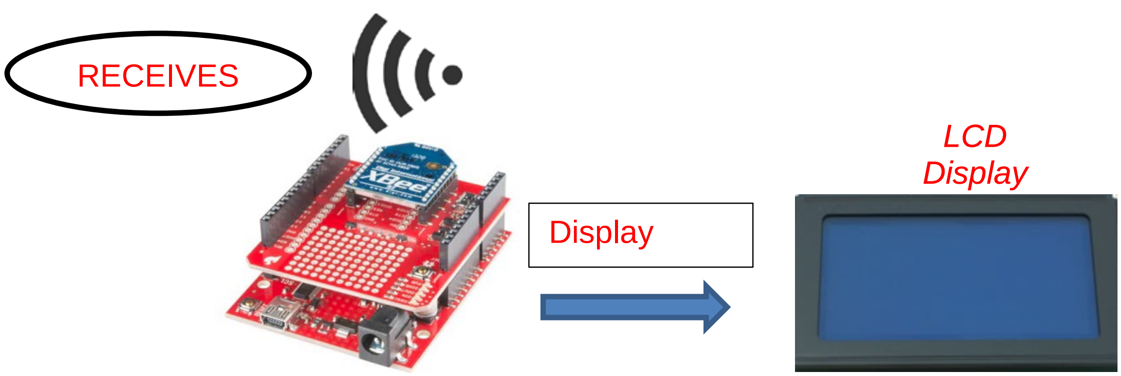


Figure 6 – Receiving System Structure and Components

The base will consist of another microcontroller containing a wireless transceiver that uses the same "ZigBee" protocol as the sender. In addition, the base will be connected either to a display, or a simple LED system that will output the sensor's data. Most of the trailers attached to 18-wheeler trucks are made from some type of metal material. This makes it more compatible to attach a magnet in between the housing that contains the sending components and the trailer. The use of a magnet will allow the driver to simply plug or unplug the housing containing the system's components. The base/receiver will be placed in the driver's cabin. The sensors will be distributed throughout the trailer. For example, for a 52 feet trailer, the distribution will consist of 3 sensors 17 feet from each other. For a 30 feet trailer, the system will distribute 2 sensors, 15 feet from each other, and so on. The final layout of the system is displayed in Figure 7. It displays the location of the sensors (from 1 to 3) on one side of the trailer. The second set of sensors will be in parallel where the displayed set is, but on the opposite side.

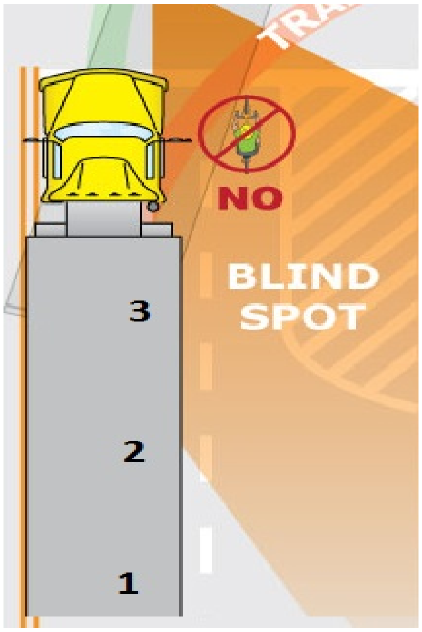


Figure 7 – Placement of the System

3.2.6 Basics of Wireless Communications

Wireless communications have become a breaking innovation in the technological world. Humans use wireless communications for basic activities such as sending a text message, accessing the web, printing, and thousands of other interactions. For the Truck Smart system, the use of wireless communication will allow to have a more flexible product that can be easily adaptable to the different scenarios and locations in which it will be operated.

Wireless communications are used in 4 basic operations. These include computer networking, data communication, transmission media, and for distributed systems (De Castro 2). Out of these 4 uses, the system will employ data communications to achieve its goal. Basic wireless communication works through transmitting and receiving electromagnetic waves in open space across a well-defined channel. This channel operates at a specific frequency and has a limited bandwidth capacity. Bandwidth is how much data can be sent across a specific channel. All forms of wireless communications use a specific protocol. A protocol is "a set of rules and conventions that govern how devices on a network communicate" (De Castro 14). These rules and conventions can represent aspects such as how the physical data is built, how the data is formatted for transmission, how the data is sent, or how the system deals with errors. The Truck Smart system will use the ZigBee protocol.

To begin the description of how a wireless network operates, the concept of a bus network needs to be introduced. A bus network is “a common channel that connects all devices” (De Castro 15). For example, in a wireless home network, the bus network would be located in the router. This allows for all devices in the network such as cell phones, laptops, Blu-ray players, smart TVs, etc. to talk to each other and access the internet at the same time. The Truck Smart system will not operate specifically using a bus network since it does not have a router, but will employ what is called a “star network”. A star network is when all devices are connected to a central device or hub. In the system, the sending devices are the microcontrollers that contain the attached sensors. The data will be sent using the XBee transceivers. Finally, the hub/base receiving device will be composed of another microcontroller using the same XBee transceiver, operating in the same channel, and employing the same exact protocol as the sending devices. The base will decode the sent data and display it. The data being sent from the sensors will travel through a communications channel. A communications channel allows for the data to be sent across the communication system. The bandwidth used in the system will be very small since the data being sent will be a simple string of 0s and 1s that will represent whether there is a vehicle in the blind spot or not.

Wireless transmission media can include microwaves, Bluetooth, WI-FI, radio signals and others. The Truck Smart system will employ radio signals since this is the way ZigBee - enabled devices operate and communicate. The radio signals coming from each of the sensors will be broadcasted using the XBee transceiver over the network across a short distance of approximately 20 to 60 feet.

3.2.7 What is ZigBee?

To begin with, ZigBee is “a low power, inexpensive, wireless mesh network standard that is employed throughout many applications that utilize wireless sensor networking and control” (Gislason 2). It was conceived back in the early 1990s and standardized in 2004. While other wireless technologies such as WI-FI or Bluetooth focus on streaming high definition media, or send data at extremely fast speeds; ZigBee aims for low data rates, devices that can be used for decades, and can run for years from a single power supply (Ex: a battery). Other advantages of using ZigBee include high reliability, cost – effective, highly secure, and an open global standard. To meet the low power and low cost criteria, ZigBee employs just one constraint in their devices: low data rates. The Truck Smart system will not be able to stream voice or video using ZigBee technology if at some point it integrated cameras to transmit live video streams wirelessly. However, it will be able to send the data from the sensors in real time since this data uses a very small amount of bandwidth on the network.

The reason behind the low data rates constraint is due to the main purpose of ZigBee devices which is to focus on wireless network and control. For example, the Truck Smart system should not be sending data when the truck is parked or when there are no vehicles in the blind spots for a very long period of time. This would result in an unproductive use of a limited power supply such as a battery or solar panel. When the devices are not in use for a very long period of time, ZigBee employs a very low power mode which reduces current draw by over 30% compared to when it is in full use. This enables to extend battery life and usability. The ZigBee standard can be used in very simple 8 – bit microcontrollers due to its small complexity and structure.

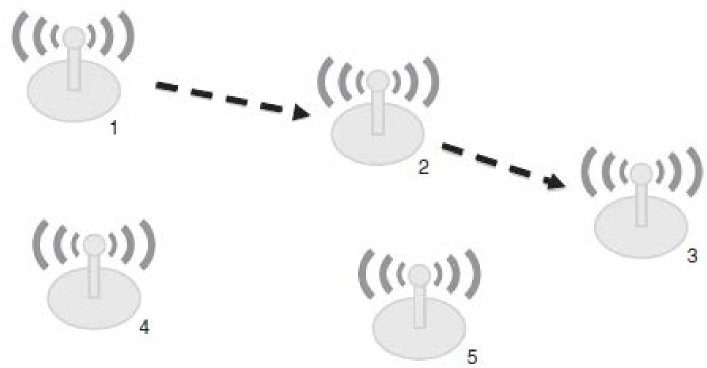


Figure 8 – Concept of the Mesh Network used by ZigBee

ZigBee can support anywhere from 2 to 2000 meshes or devices in a multi-hop mesh network. This allows for the employment and implementation of very big systems using this protocol. ZigBee employs the IEEE 802.15.4 wireless specification for communication and an "end-to-end acknowledgments to verify data made it to the destination" (Gislason 5). The use of Offset-Quadrature Phase-Shift Keying (O-QPSK) and Direct Sequence Spread Spectrum (DSSS), a combination of technologies that provides excellent performance in low signal-to-noise ratio environments and allows for very high reliability regardless of location. This means that the system will still be fully operational even if the vehicle is under a tunnel or in an isolated location. Figure 8 represents the concept behind a mesh network and how the data is transferred from one point to another. The data being sent from the 1st device goes to the 2nd device. From there the 2nd device receives the data, stores it, reads the data for itself, encodes it, and finally sends it to the 3rd device and so on until all data reaches the base/hub device. Devices 4 and 5 can still send data wireless to the base. Even though they might not appear as part of the mesh network, the system will automatically find the best way to read the data from these devices. For example, it might send the data from 4 to 5 since they are next to each other and then send it from 5 to 3. Regardless of the automatic combination chosen by the system, ZigBee would make sure that the data from those devices is sent and received as well.

3.2.8 The ZigBee Protocol and Modules

The ZigBee protocol uses what is called a "Carrier Sense Multiple Access Collision Avoidance" (CSMACA) to increase reliability (Gislason 5). Right before beginning data transmission from one channel to the other, it waits for the channel to make sure it is clear for transmission. If there is another device sending data to the same channel, it will wait its turn and won't send the data at the same time in order to prevent corrupt data. ZigBee uses a 16-bit CRC on each packet, called a Frame Checksum (FCS). This ensures that the data bits are correct. Each packet is retried up to three times (for a total of four transmissions). If the packet cannot get through after the fourth transmission, ZigBee informs the sending node so something can be done about it (Gislason 5).

The use of mesh networking in ZigBee allows for multiple error detection and fails mechanisms. For example, in the Truck Smart system, where there will be a total of 6 sending devices (3 in each side of the trailer) and 1 receiving hub, the data from device 3 located in the far back, transfers from 3 to 2, from 2 to 1, and finally from 1 to the base. If device 2 were to fail, ZigBee will automatically detect the failure and route the data directly to device 1 or use any of the other 4 devices that are still functional. This is done through a mechanism called "reliable broadcasting" and multicasting. The data coming from one device is distributed across all nodes in the network. Through multicasting this same message can be sent to any given group of nodes. Even though the signal is routed from one device to the other, the additional devices also receive this signal but they do not transfer it to the next mesh. Instead they store this data, and in case there is a failure on the routing, then they transfer the data to the next adjacent device until it reaches the hub.

Figure 9 – XBee Pro S2C Transceiver

Using 802.15.4 radios to transmit data, ZigBee devices can operate in both 2.4 GHz and 915 MHz frequencies (Gislason 8). In either case, the RF boards must be certified by the government to ensure that they do not interfere with other wireless devices. There are several options when it comes to the physical devices or modules that employ the protocol. A module is a single board with ZigBee components installed that is ready to communicate out of the box. There are several companies that sell these boards, the Truck Smart application will employ the Digi Key – XBee Pro S2C *refer to Figure 9*.

In terms of security, the ZigBee protocol employs the National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES). This standard, AES-128, is a block cipher that encrypts and decrypts packets in a manner that is very difficult to crack (Gislason 11). The use of 128 – bit encryption, makes it very difficult for hackers to illegally access data being sent across a network using ZigBee devices. In addition, ZigBee implements authentication which detects when malicious data has been injected into the network and discards it. Transceivers communicate at a rate of approximately 250 kilobits per second (kbps). Transceivers can send or receive data, but not both at the same time. A ZigBee network can have up to 16 channels in the 2.4 GHz frequency.

3.3 Wireless Transceivers

A wireless transceiver is both a transmitter and a receiver combined into once device. Various transceivers were researched and the best match with the requirements of the Truck Smart system was chosen. The most important of aspect of the transceiver is that it needs to support the chosen wireless communication method of ZigBee. Below you will the top four choices and their corresponding pros and cons. This section will then end with the final choice and an explanation of why it was chosen.

3.3.1 Freescale MC13213

The Freescale MC13212 is part of Freescale's 2nd generation ZigBee platform devices. It consists of a small 8-bit microcontroller with the ZigBee transceiver embedded on the board *refer to Figure 10*. It operates in a 2.4 GHz radio frequency and has very low power consumption. It has a 60 KB flash memory and 4 KB of RAM. The device can be used for multiple applications such as residential and commercial automation, industrial control, health care, and consumer electronics. It can operate in temperatures between -40° to 85° C (Freescale 3). It contains seven general purpose input/output pins and three low power modes. The embedded wireless transceiver can support up to three types of networks which include star, mesh, and tree.

The device supports two data transfer modes that include packet mode and streaming mode. Packet mode is when the data is buffered in on-chip RAM while streaming mode buffers the data word by word. The device does not employ any form of data verification after transmitting from one point to the other. In addition, to use this device for the Smart Truck application, the integration team will be limited to using that one specific type of microcontroller. As a result, in order to use the wireless transceiver; the programmer is required to use the embedded microcontroller as well. Therefore, this device is not a good choice for the application since it would limit the team to a single implementation and architecture while not employing any type of data verification after transmission. This might cause for data to be corrupt once received and result in multiple errors in the system.

Figure 10 – The Freescale MC 13213 with the embedded wireless transceiver on the board

3.3.2 Panasonic PAN802154HAR00

Based on the Freescale™ ZigBee Sensor Application Reference Design (SARD) development platform, the Panasonic PAN802154HAR00 is a communication device that incorporates both a microcontroller as well as a wireless transmitter similarly to the Freescale MC13213. The module uses "the Freescale's 802.15.4 transceiver (MC13193), micro controller (GT60) and is licensed to use all released Freescale ZigBee Protocol stack layer software" (Panasonic 1). The device operates in the 2.4 GHz band and is fully compliant with the IEEE 802.15.4 standard.

Figure 11 shows the physical layout of the device. The RF section of the board is on the far right embedded in what appears to be a small metal box. Inside this box there are two antennas for sending and receiving data from/to another device. Shielding the RF section on the board helps prevent leakage and improves performance as well.

Figure 11 – Panasonic PAN802154HAR007

The device has a lot of similar features and functionality when compared to the Freescale MC and the XBee Pro S2C, however its main disadvantage is that when in use has a DC current draw of 100 mA. For the Truck Smart system this amount of power consumption is unacceptable due to the very limited power supply. As a result, this device is not a good choice for the system.

3.3.3 Freescale MC13202

The Freescale MC 13202 *refer to Figure 12* is a wireless radio-only device that can be used with the HCS08 family of microcontroller units. It supports peer-to-peer star as well as mesh networking making it suitable for the Truck Smart system. The device is a small chip that attaches to the input pins on the microcontroller. The module can support anywhere from simple point-to-point systems to complete ZigBee networks such as in the Truck Smart system. The device is used mainly in the industrial, scientific and medical fields. The module communicates with the MCU through a four-wire serial peripheral interface (SPI) connection and an interrupt request output.

Figure 12 – Freescale MC13202

Even though the device's power consumption does not exceed more than 42 mA when in full use, it has a very small communication range of approximately 20 feet. In addition, it is only supported by the microcontrollers in the HCS08 family limiting the devices that can be used for the system. Due to both these limitations, the MC13202 is not a good choice when implementing the system since it would limit the developing team to using a limited compatible set of resources. The small range supported automatically disqualifies it as a possible solution for the Truck Smart system since the system requires a wireless transceiver that can support up to 70 feet.

3.3.4 XBee Pro S2C (Final Selection)

To have the sensors communicate to the base, the system will use the XBee Pro S2C *refer to Figure 9*. This device is a simple transceiver (can send or receive data) that uses the ZigBee protocol for communication between itself and other devices on the network. It uses an RF Module in the 2.4 GHz frequency. Its data rate is 1 Mbps and operates with a current of 31 mA for receiving and 45 mA for transmitting (Datasheet 1). It uses a voltage supply in between 2.1 and 3.6 Volts. The supported serial communication interfaces are UART and SPI. When not in use, it consumes a small current of 1uA. It has a range of 200 feet indoors and 4000 feet outdoors. The supported range enables to easily implement the Truck Smart application since the wireless components will not be sending data over more than 70 feet.

The choice for the communications interface in between the XBee Pro and the microcontroller will be very important due to the fact that they support different data transfer bandwidths. For example, UART supports up to 1 MB/s maximum burst while SPI can support up to 5 MB/s. For the Truck Smart system where there will be approximately 5 to 8 devices sending data wirelessly at the same time using UART will be the most efficient choice. Supporting 1 MB/s is more than the required amount since the sensor's data is a small string of 0s and 1s that gets updated every 2 to 3 seconds. Using UART instead of SPI allows for a better use of the limited resources as well as simplifies the architecture of the system since all sending device will be using the same protocol.

The device can operate within a temperature range from -40º C to +85º C making it suitable for the application's requirements. Truck drivers drive across the country on a daily basis. Sometimes they operate in Florida where it can be 40º C during the summer, or in Alaska where the temperature can be as low as -20º C during the winter. For the initial testing phase, the system will be using the Arduino Uno microcontroller. In order to connect the XBee transceiver to the microcontroller, it will use the Arduino shield that simplifies the connection in between these devices.

The shield is a simple board with a slot to plug in the transceiver. The shield plugs directly into the input pins in the microcontroller for a simplified installation.

Due to its low power consumption, excellent rage support, as well as simplified architecture and usability, the XBee Pro S2C will be the device used to handle all the wireless communications for the Truck Smart system.

3.4 Blind Spot Detection Sensors

Blind spot detection sensors are nothing more than various kinds of distance detection sensors. The main idea behind them is to send out a signal of some kind (IR light, Ultrasound, Radar, etc.) and record the time it takes for the signal to bounce back from surface of the nearest object and return to the sensor. The distance of that nearest object the signal bounced off of can be calculated based on the time it took for the signal to return. If there isn't an object in the line of sight of the sensor, the signal (light or sound) would eventually dissipate in the environment and not return to the sensor; in which case it can be determined that there aren't any objects nearby. Other applications of these sensors include navigation systems, counting devices (e.g., wait watcher, product assembly), surveillance system, edge detection, various robotic systems, and military applications. Robustness, lightweight, inexpensive and fast response time makes these sensors suitable to be used in the development of navigation aids.

3.4.1 Sensor Choice (IR vs. Ultrasound vs. Radar)

Selecting the right sensor for this system is very crucial because the reliability and accuracy of the sensor directly affects the performance of the system. A false detection or failure to detect a vehicle can result in an accident. Therefore, the team carefully took the pros and cons stated below in consideration to pick out the right sensor.

Table 7 – Ultrasonic vs. Infrared: Pros and Cons

Infrared	Ultrasonic
Cheaper in cost	Lighting doesn't affect the sensibility (works well in daytime as well as night time)
Faster response time	Color and transparency of the object doesn't affect the accuracy
smaller in size	Significantly longer range than IR
Better for soft surfaces as the texture of a surface doesn't affect the absorption of light	Overall a better accuracy than IR

Table 8 – Ultrasonic vs. Radar: Pros and Cons

Radar	Ultrasonic
Much quicker response time than US	Cheaper in cost (approximately 3X cheaper)
Less affected by temperature	Significantly lower power consumption compared to radar sensors
More rigid	Smaller in size
The life expectancy is less prune to constant vibrations or moving	More suitable for wireless applications compared to radar due to the smaller data size of the output.

In conclusion, IR sensors are cheaper, faster, and smaller than US but they are only accurate for indoor applications and therefore aren't suitable for this system. On the other hand, radar sensors are more accurate and rigid than US but higher power consumption, higher cost, and the need for a bigger bandwidth make them less attractive for this system. The team has decided to use ultrasonic sensors based on these considerations.

3.4.2 Ultrasonic Sensors

The "sonic" in ultrasonic refers to sound, and "ultra" means that humans cannot hear it (but bats and dogs can). Ultrasonic sensors are based on measuring the properties of sound waves with frequency above the human audible range. They are based on three physical principles: time of flight, the Doppler effect, and the attenuation of sound waves. Ultrasonic sensors are non-intrusive in that they do not require physical contact with their target. They can measure distances from 0 to 2.5 meters, with a precision of 3 cm.

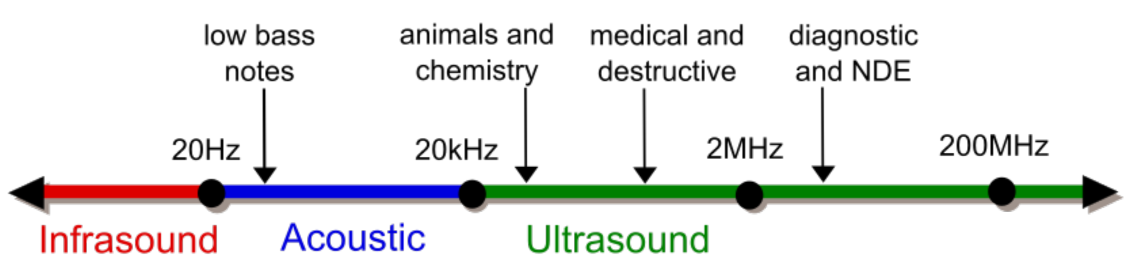


Figure 13 – Sound Spectrum

Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front (refer to figure 13). One opening transmits ultrasonic waves (like a tiny speaker), the other receives them (like a tiny microphone). Speed of sound is approximately 341 m/s in air; ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object. The equation used to calculate the distance is:

$$Distance = (Time * Speed\ of\ Sound)/2$$

Equation 1 – Sensor Distance Calculation

Where Time = the time between when an ultrasonic wave is transmitted and when it is received. It is divided by 2 because the sound wave has to travel to the object and back so the actual distance is only half of this number.

Besides distance detection and obstacle avoidance, other uses of ultrasonic sensors include medical imaging, measurement of dynamically changing depths, measurement of dynamically changing heights, counting number of units, etc.

Figure 14 – Typical Distance Measurement Ultrasonic Sensors

3.4.2.1 Echolocation: Ultrasonic Sound Waves in Nature

The idea to use ultrasonic sound to measure distance came from animals like Bats and Whales. Bats use a technique called echolocation to navigate and find food in the dark. Echolocation is the use of sound waves and echoes to determine where objects are in space. To echolocate, bats send out sound waves from their mouth or nose. When the sound waves hit an object they produce echoes. The echo bounces off the object and returns to the bats ears. Bats listen to the echoes to figure out where the object is, how big it is, and its shape. A bat can tell how big an insect is based on the intensity of the echo. A smaller object will reflect less of the sound wave, and so will produce a less intense echo. The bat can sense in which direction the insect is moving based on the pitch of the echo. If the insect is moving away from the bat, the returning echo will have a lower pitch than the original sound, while the echo from an insect moving toward the bat will have a higher pitch. Using echolocation, bats can detect objects as thin as a human hair in complete darkness.

3.4.2.2 Ultrasonic Signal Generation/Detection

Ultrasound is most commonly generated as a direct conversion from electrical energy. This is accomplished by applying a rapidly oscillating electrical signal to a piezoelectric crystal attached to a mounting. The charge causes the crystal to expand and contract with the voltage, thereby generating an acoustic wave. The waves are later detected by a piezoelectric receiver, which converts the waves back into voltage using the same method.

In the figure below, it is visualized how a chirp is emitted from the "speaker." It bounces off of an object. The echo returns to the microphone. The time it takes to travel to the object and back is used to figure out the distance.

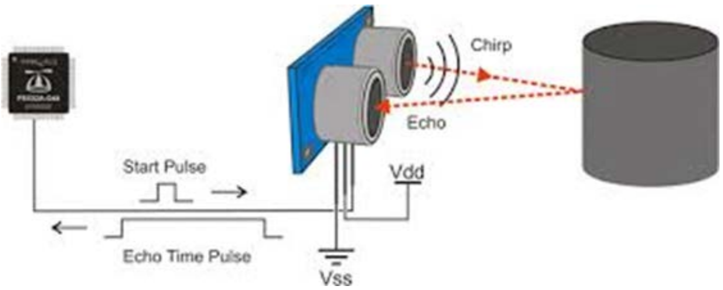


Figure 15 – How an Ultrasonic Signal is Sensed

3.4.2.3 Measuring Principles

Three different properties of the received echo pulse may be evaluated, for different sensing purposes. They are: Time of flight (for sensing distance), Doppler shift (for sensing velocity), and Amplitude attenuation (for sensing distance, directionality, or attenuation coefficient). All three methods make use of different physical principles, but they all employ the same measuring procedure. In each case, an ultrasonic sound wave is created, received, and evaluated.

1. Time of Flight: an ultrasonic transmitter emits a short burst of sound in a particular direction. The pulse bounces off a target and returns to the receiver after a time interval, T. The receiver records the length of this time interval, and calculates the distance travelled based on the speed of sound.
2. Doppler shift: When a wave reflects off of a moving object, its frequency is shifted by an amount proportional to the velocity of the object. This fact can be exploited in ultrasonic sensing by having the receiver measure not the time of flight but the frequency of the returning echo pulse.
3. Amplitude attenuation: Ultrasonic sound attenuates much faster than audible sound when propagating through air. By measuring the intensity of the returning pulse, an estimate of the distance travelled can be made. Attenuation may also be caused by an increased angle between the target and receiver, which may even deflect the echo somewhere else and not be heard at all.

Another measurement principle is pulsed vs. continuous measurement. Time-of-flight-based sensing requires emitting a pulse and waiting for it to return. This waiting time limits the speed with which successive measurements can be made, without risking confusion. However, Doppler- and attenuation-based sensing devices do not have the same restrictions: a constant wave of ultrasound may be emitted, and the received wave's attenuation or frequency continuously analyzed. This may make measurements speedier, effectively increasing the sensitivity of the sensor.

3.4.2.4 Ultrasonic Sensor Constraints

The main advantage of ultrasonic sensors is that measurements may be made without touching or otherwise impeding the target. In addition, depending on the distance measured, measurement is relatively quick (it takes roughly 6ms for sound to travel 1m). However, many factors such as temperature, angle, and material may affect measurements.

Here is a list of most notable constraints in ultrasonic sensing:

1. **Minimum Sensing Distance**: Ultrasonic proximity sensors have a small unusable area near the face of the sensor. If the ultrasonic beam leaves the sensor, strikes the target, and returns before the sensor has completed its transmission, the sensor is unable to receive the echo accurately. This unusable area is known as the blind zone. The outer edge of the blind zone is the minimum distance an object can be from the sensor without returning echoes that will be ignored or misread by the sensor.

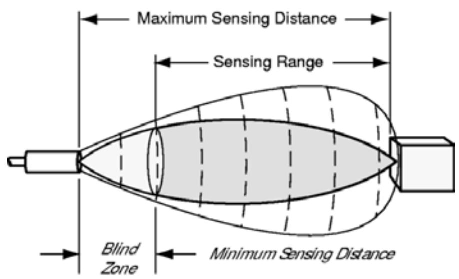


Figure 16 – Blind zone of an ultrasound sensor

2. **Maximum Sensing Distance (based on the material of the object):** Target size and material determine the maximum distance at which the sensor is capable of seeing the object. The harder an object is to detect, the shorter the maximum sensing distance can be. Materials that absorb sound (foam, cotton, rubber, etc.) are more difficult to detect than acoustically reflective materials, like steel, plastic, or glass. If detected at all, these absorbent materials can limit maximum sensing distance.

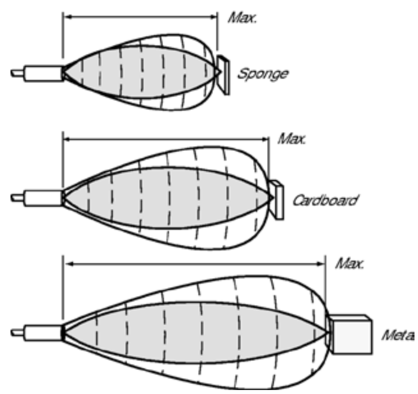


Figure 17 – Maximum Sensing Distance of an Ultrasound Sensor

3. **Effective Beam:** When the transducer vibrates, it emits ultrasonic pulses that propagate in a cone-shaped beam. This cone can be adjusted, usually via potentiometer, to widen or extend the sensing range.

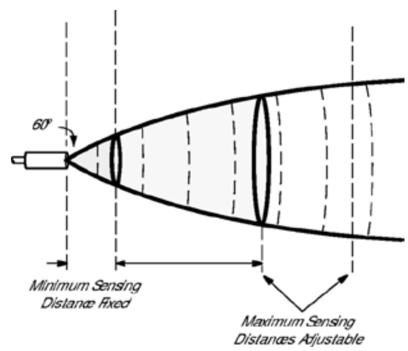


Figure 18 – Adjustable Effective Beam of an Ultrasound Sensor

4. **Spacing Considerations**: Spacing between sensors is determined by their beam angles. The sensors must be spaced so they do not interfere with each other. This interference is sometimes called "crosstalk." When more than one ultrasonic sensor is in use, some experimentation may be needed to determine optimal spacing for the application.

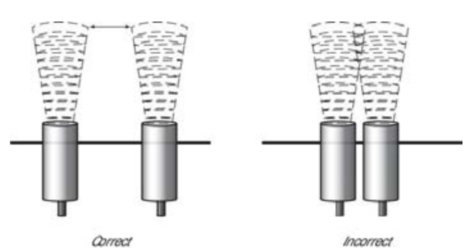


Figure 19 – Spacing Considerations

5. **Angle:** For the transmitted wave to echo back to the receiver, the target surface must be perpendicular to the transmitter. Round objects are therefore most easily sensed since they always show some perpendicular face. When targeting a flat object, care must be taken to ensure that its angle with respect to the sensor does not exceed a particular range.

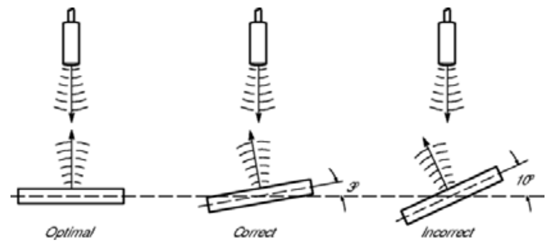


Figure 20 – Optimal Angle for an Ultrasound Sensor

6. **Temperature:** The surface temperature of a target can also influence the sensing range. Radiated heat from high temperature targets distorts the sound beam, leading to shortened sensing range and inaccurate readings. Ultrasound sensors may need to be recalibrated to make accurate measurements in a new environment (or, an on-board temperature sensor may need to be incorporated).

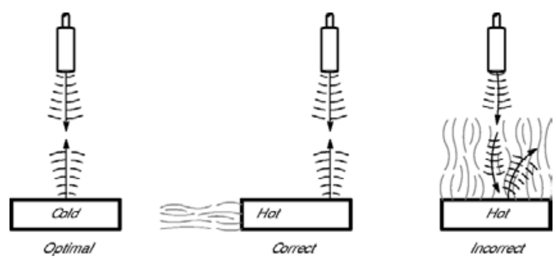


Figure 21 – Effect of Temperature on Ultrasonic Waves

3.4.2.5 Ultrasonic Advantages and Disadvantages

Ultrasonic Advantages

1. An ultrasonic sensor's response is not dependent upon the surface color or optical reflectivity of the object. For example, the sensing of a clear glass plate, a brown pottery plate, a white plastic plate, and a shiny aluminum plate is the same.

2. Ultrasonic sensors with digital (ON/OFF) outputs have excellent repeat sensing accuracy. It is possible to ignore immediate background objects, even at long sensing distances because switching hysteresis is relatively low.

3. The response of analog ultrasonic sensors is linear with distance. By interfacing the sensor to an LED display, it is possible to have a visual indication

of target distance. This makes ultrasonic sensors ideal for level monitoring or linear motion monitoring applications.

Ultrasonic Disadvantages

1. Ultrasonic sensors must view a surface (especially a hard, flat surface) squarely (perpendicularly) to receive ample sound echo. Also, reliable sensing requires a minimum target surface area, which is specified for each sensor type.

2. While ultrasonic exhibits good immunity to background noise, these sensors are still likely to falsely respond to some loud noises, like the “hissing” sound produced by air hoses and relief valves.

3. Proximity style ultrasonic sensors require time for the transducer to stop ringing after each transmission burst before they are ready to receive returned echoes. As a result, sensor response times are typically slower than other technologies at about 0.1 second. This is generally not a disadvantage in most level sensing and distance measurement applications. Extended response times are even advantageous in some applications. Transmitted beam style ultrasonic sensors are much faster with response times on the order of 0.002 or 0.003 seconds.

4. Ultrasonic sensors have a minimum sensing distance.

5. Changes in the environment, such as temperature, pressure, humidity, air turbulence, and airborne particles affect ultrasonic response.

6. Targets of low density, like foam and cloth, tend to absorb sound energy; these materials may be difficult to sense at long range.

7. Smooth surfaces reflect sound energy more efficiently than rough surfaces; however, the sensing angle to a smooth surface is generally more critical than to a rough surface.

Despite these disadvantages, ultrasonic sensors are practical for vehicle blind spot detection purposes because the sensors aren’t affected by these disadvantages in typical road conditions. For example, blind spot detectors are always sensing for a hard surface (other vehicles), loud noises produced by air hoses and relief valves are non-existent on open roads, minimum sensing distance is negligible for vehicular applications, the operable temperature range for ultrasonic sensors is well within the typical road conditions in habitual environments, etc. Some of the other constrains can be overcome with proper adjustments of the sensor. One example of this would be to do enough trial and error to determine the angle at which the sensor should be mounted on the vehicle to maximize the target surface area mentioned in disadvantage 1.

3.5 Printed Circuit Board

Printed circuit boards (PCBs) are the most common method of assembling modern electronic circuits. A PCB mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate.
Comprised of a sandwich of one or more insulating layers and one or more copper layers that contain the signal traces and the powers and grounds, the design of the layout of printed circuit boards can be as demanding as the design of the electrical circuit. Most modern systems consist of multilayer boards of anywhere up to eight layers (or sometimes even more). Traditionally, components were mounted on the top layer in holes that extend through all layers. These are referred as through hole components. More recently, with the near universal adoption of surface mount components, you commonly find components mounted on both the top and the bottom layers. PCBs are very popular because they greatly improve the durability of electronic products by reducing the chances of short circuits and incorrect wiring.

3.5.1 Composition of a PCB

PCB has alternating layers of different materials that are laminated together with heat and adhesive such that the result is a single object. The layers can be seen in the figure below.

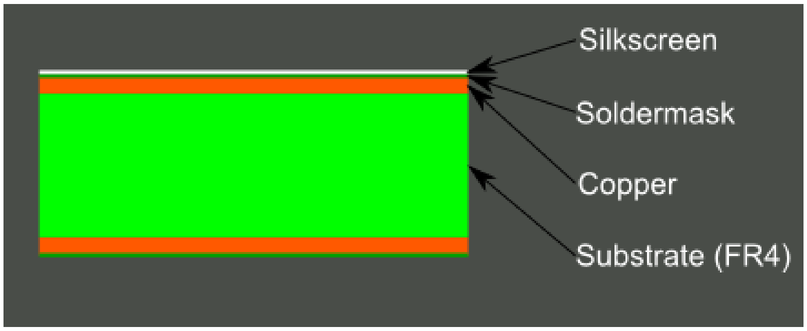


Figure 22 – Composition of a PCB

Substrate
Substrate is the solid core that gives PCB its rigidity and thickness. FR-4 (or FR4) is a grade designation assigned to glass-reinforced epoxy laminate sheets, tubes, rods and printed circuit boards. FR-4 is a composite material composed of woven fiberglass cloth with an epoxy resin binder that is flame resistant (self-extinguishing). "FR" stands for flame retardant, and denotes that safety of flammability of FR-4 is in compliance with the standard UL94V-0. FR-4 glass epoxy is a popular and versatile high-pressure thermoset plastic laminate grade with good strength to weight ratios. The material is known to retain its high mechanical values and electrical insulating qualities in both dry and humid conditions. These attributes, along with good fabrication characteristics, lend

utility to this grade for a wide variety of electrical and mechanical applications. There are also flexible PCBs built on flexible high-temperature plastic.

Copper

The next layer is a thin copper foil, which is laminated to the board with heat and adhesive. On common, double sided PCBs, copper is applied to both sides of the substrate. In lower cost electronic gadgets, the PCB may have copper on only one side. When we refer to a **double sided** or **2-layer board,** we are referring to the number of copper layers (2) in our lasagna. This can be as few as 1 layer or as many as 16 layers or more. The copper thickness can vary and is specified by weight, in ounces per square foot. The vast majority of PCBs have 1 ounce of copper per square foot but some PCBs that handle very high power may use 2 or 3-ounce copper. Each ounce per square translates to about 35 micrometers or 1.4 thousandths of an inch of thickness of copper.

Solder Mask

The layer on top of the copper foil is called the solder mask layer. This layer gives the PCB its green color. It is a thin lacquer-like layer of polymer that is usually applied to the copper traces of a printed circuit board for protection against oxidation and to prevent solder bridges from forming between closely spaced solder pads. A solder bridge is an unintended electrical connection between two conductors by means of a small blob of solder. PCBs use solder masks to prevent this from happening. Solder mask is not always used for hand soldered assemblies, but is essential for mass-produced boards that are soldered automatically using reflow or solder bath techniques. Once applied, openings must be made in the solder mask wherever components are soldered, which is accomplished using photolithography. This layer helps the user to solder to the correct places and prevent solder jumpers. While it's recommended to get the PCBs printed using professional equipment, it is possible to take a blank copper plate and use a solder mask layer to make homemade PCBs. This requires simple household materials like Sodium Carbonate solution, a UV light source, and a laminator.

Silkscreen

The white silkscreen layer is applied on top of the solder mask layer. It's normally used on the component side to identify components, test points, PCB and PCBA part numbers, warning symbols, company logos and manufacturer marks and allows for easier assembly and indicators for humans to better understand the board. Silkscreen labels are often used to indicate what the function of each pin or LED. While it's possible to use Silkscreen on solder side of the board as well, it should only be utilized if necessary as it increases the manufacturing price by a lot. In addition to component identification, Silkscreen is also used to put manufacturer's mark and identifier on the circuit board. Although there are no strict standards for where these markings appear, manufacturers will generally try to place them in non-critical areas. Silkscreen is most commonly white but any ink color can be used. Black, gray, red, and even yellow silkscreen colors are widely available. Conventional silk-screening requires polyester screens stretched across aluminum frames. Solvent resistant emulsions are used to coat the screens prior to imaging. Both the screen and panel must be registered precisely to ensure proper alignment. This method of screening requires the following equipment: Laser photo plotter to produce the initial film, UV printer, spray developer and curing ovens. Screens can be cleaned and reclaimed to be used again although not for jobs which require high resolutions.

3.5.1.1 PCB Terminology

This section lays out some common PCB terminology that is used in the design section.

- **Annular ring** - the ring of copper around a plated through hole in a PCB
- **DRC** - design rule check. A software check of your design to make sure the design does not contain errors such as traces that incorrectly touch, traces too skinny, or drill holes that are too small.
- **Drill hit** - places on a design where a hole should be drilled, or where they actually were drilled on the board. Inaccurate drill hits caused by dull bits are a common manufacturing issue.
- **Pad** - a portion of exposed metal on the surface of a board to which a component is soldered.
- **Panel** - a larger circuit board composed of many smaller boards which will be broken apart before use. Automated circuit board handling equipment frequently has trouble with smaller boards, and by aggregating several boards together at once, they process can be sped up significantly.
- **Plated through hole** - a hole on a board which has an annular ring and which is plated all the way through the board. May be a connection point for a through hole component, a way to pass a signal through, or a mounting hole.
- **Slot** - any hole in a board which is not round. Slots may or may not be plated. Slots sometimes add to add cost to the board because they require extra cut-out time.

3.5.2 Design Recommendations for Better Reliability

PCB is a very crucial component of an electronic system so it is important to design a reliable, long lasting PCB. While each PCB is different and requires unique design approaches, this section lays out some basic design precautions that can be taken to improve performance and reliability of the system. Some of them are general knowledge that most designers already know and some are more advanced. A few of these techniques have been consistent for 25 years and are still in use today. In addition to increasing reliability, these techniques also help reduce dimensions of the board making the overall electronic system even more compact. While this section only gives a brief overview of the recommended techniques, chapter 4 of this report gives more of an insight of how these techniques help overcome the common PCB constrains.

Technique #1: Keep traces as short and direct as possible
Longer traces mean higher resistance and inductance in the on board data and power lines; keeping them short can drastically improve performance of the PCB. This applies particularly in analog and high-speed digital circuitry where impedance and parasitic effects will always play a part in limiting your system performance.

Rule #2: Group related components and test points together
Keep all the related components together because they are most likely connected to each other. Grouping them together will ensure shorter paths between them keeping the traces together. This will also make trouble shooting a little easier during the testing phase. This technique especially applies to ICs on the board because ICs require bypass capacitors and resistors and grouping them together makes the most of noise cancellation properties of the bypass capacitor.

Technique #3: Use a power plane to manage the power lines and ground
This technique helps ensure that power flows as effectively as possible with minimal impedance or voltage drop, and that ground return paths are adequate. It is also recommended to run multiple supply lines in the same area of the board. Whenever possible, for multilayer boards, it's recommended to use a whole layer as the ground plane. Also, there shouldn't be any overlap between analog and digital ground planes, or analog and digital power planes.

Technique #4: Pick a right grid
Find a grid spacing that suits as many of your components as possible and use it throughout. Although multiple grids may seem appealing, a little additional thought at the early stages of the layout can avoid spacing difficulties and will maximize board use.

Technique #5: Use decoupling capacitors wherever necessary
Decoupling capacitors play a major role in shielding the on board IC from high frequency noise. Capacitors are very inexpensive and robust and should be used whenever possible. Since decoupling capacitors can work on a wide range of values, keeping all of them in a standard range helps keep the inventory simple.

Technique #6: Lower thermal resistance of the circuit by using enough heat sinks
Heat sink is an external low thermal resistance device attached to the IC to aid heat removal. Using a heat sink with larger area and weight will better aid the heat dissipation. In addition, natural air ventilation inlets and outlets should be provided and using forced air may be necessary if several watts are dissipated in a confined space.

Technique #7: Panelize your PCB by replicating it on a larger board
Using a size that best suits the equipment used by the manufacturer will improve the cost of prototypes and manufacturing. Start by laying out the board as one panel and repeat the design a several times within the preferred panel size.

Technique #8: Use automatic signal routing CAD software with extreme caution
Even though automatic routing feature is available in most CAD software, critical signal paths should be routed by hand to avoid undesired coupling and/or emissions. Properly designed multilayer PCBs can reduce EMI emissions and increase immunity to RF fields, by a factor of 10 or more. Almost all PCB design software have a Design Rule Check function. This function checks the design for a lot of different validations and should be used frequently during design.

Technique #9: Use the Silkscreen carefully
Silkscreen is used on a PCB to identify components, part numbers, and wiring symbols and allows humans to better understand the board. Since it's a crucial part for understanding the circuit, proper fonts and orientation should be used. Full use of silk screening on both sides of the board streamlines production and can reduce re-work.

Technique #10: Generate the PCB manufacturing data and verify it before sending it out to be fabricated
It's advised to create your own Gerber file before sending it to the manufacturer. This way any errors previously missed can be caught before it's forever set in fiberglass, resin, and copper.

Technique #11: Place the mounting holes strategically
The mounting holes for the PCB should be planned well in advance and should be clear of tracks and components. The holes should be coated with dielectric material if metal screws or connectors are being used. PCB manufacturers usually charge more for multiple hole sized, therefore, it is recommended to keep the mounting holes consistent.

The techniques mentioned above are the basic strategies followed in most PCB designs. Besides these there are some other techniques that can be useful to some designs. One of not so common technique is to consolidate passive component values. This technique is used because for low current circuits, like the Truck Smart system, it's often possible that a wide range of values can be used for some passive components (like resistors) to produce the same results. Consolidating on a smaller range of standard values makes the Bill of Material simpler. Good PCB layouts also isolate critical analog paths from sources of high interference (I/O lines and connectors, for example). High frequency circuits (analog and digital) should be separated from low frequency ones. If the system has separate analog and digital ground and power planes, the analog ground plane should be underneath the analog power plane, and similarly, the digital ground plane should be underneath the digital power plane. Using the techniques listed above helps produce reliable, long lasting PCBs.

3.5.3 PCB Design Software

Electronic Design Automation (EDA) software is needed to develop PCBs once designing and prototyping of the circuit is done. This software contains a schematic editor for designing circuit diagrams. Parts can be placed on many sheets and connected together through ports. The software also allows back annotation to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic. Properly using these functions makes it easier to make well-functioning PCBs. Some of the most popular EDAs are Altium, Eagle, DipTrace Starter, and ORCAD. While most of this software is quite pricey, ORCAD, Eagle, and DipTrace do offer free versions of their software for non-commercial use. However, the free versions have limited number of allowed layers. Things to consider while picking out software are Community support, Ease of use, Capability, and Portability. Even though Eagle does not have as appealing of a user interface as its competitors, the large community of users makes it very appealing. There are plenty of online tutorials and libraries available for Eagle and therefore the team has decided to use Eagle for this project. The table below lists different PCB design software packages and their prices. Please note that all of these software packages have a lot more versions but only some are shown in this table.

Table 9 – PCB Design Software

Software Package	Specs	Price
Eagle Standard	99 sheets, unlimited layers	$820
DipTrace Standard	1000 pins, unlimited layers	$345
Eagle Free Version	16 layers, 4.0” X 6.0” board	$0.00
DipTrace Free Version	2 layers, 500 pins	$0.00
OrCAD Free Version	2 layers, 100 pins	$0.00

3.5.3.1 PCB Design Using Eagle

PCB design in EAGLE is a two-step process. The first step is to design a schematic (creating a .sch file), and the second step is to create a board layout in .brd format based on that schematic. Eagle's board and schematic editors work hand-in-hand. Designing a schematic first is important because it helps catch errors beforehand. Once this is done, a Gerber file is created and sent to the PCB manufacturer for printing.

Procedure to create a schematic in Eagle is listed in this section. Start by creating a new project by right clicking on the project directory in control panel and selecting "New Project". Then a new schematic can be added to the project folder by right-clicking the folder and selecting schematic. A new schematic window will pop up when this is done. Then to add parts, "Add" tool in Navigation bar is used. This tool opens us a library navigator with a list of libraries. Needed libraries can be expanded to select the desired parts it holds. When a part is selected from the list on the left side, window on the right updates shows its schematic symbol. This tool also has a search functionality that comes in very handy when navigating through dozens of libraries. Wildcards can be added to the search function by adding a * at the beginning and end of a search query. This will return the whole series related to that product instead of just one package. The next step of the process is to add a frame to the design. This isn't a critical component but it keeps the schematic looking clean and organized. When the schematic is saved, the frame's title updates automatically. The next step of the process is to add general power inputs and ground. All six sensor PCBs are the same and they all use a 0.5 mm Barrel Jack (part name: POWER_JACKPTH), 0.1 microfarad Ceramic coupling capacitor, generic voltage supply, and two grounds.

All of these are found in the standard built in Eagle libraries except for the Barrel Jack; for which "SparkFun-Connectors" library was downloaded. The next step is to add the microcontroller and the supporting circuit. For the Truck Smart system, an ATmega328, four 0.25 Watt resistors, three LEDs, and a 0.1 microfarad Ceramic coupling capacitor are needed. All the components listed above are then to be connected like the diagram shown in hardware design section of this report. This is done using the net tool. To use this tool, select the tool from the toolbar; while the tool is selected, left clicking on a component will start a wire. If the schematic starts getting messy because of the wires, net stubs can be used as well to connect various components. This technique comes in extremely handy when multiple ground pins need to be connected together; it keeps the schematic clean. Keep in mind that every component in the schematic should have two editable text fields: name and value. And each component should have a unique name (e.g. R1, R2, LED4, etc.). The "name" tool from the toolbar can be used to change name of a component. The "show" tool is very useful when verifying that pins across your schematic are connected correctly. When the show tool is used on a net, every pin connected to the net lights up.

Once the schematic is designed and verified, it can be transferred to a board layout using Eagle's board designer. Board designer is where the dimensions of the board come together, parts are arranged, and connected by copper traces. The conceptualized, idealized schematic that was designed in previously becomes a precisely dimensioned and routed PCB in the board editor. To begin this process, switch from the schematic editor to the related board by clicking the Generate/Switch to Board command in the toolbar. Doing this opens a new board editor window; all the parts in the schematic should already be in this layout stacked on top of each other, ready to be placed and routed. The gold lines, called air wires, connect between pins and reflect the net connections that were made on the schematic. There should also be a faint, light-gray outline of a board dimension to the right of all of the parts. Once the board is displayed, the first job in this PCB layout should be to arrange the parts, and then minimizing the area of PCB dimension outline. When rearranging the parts, care should be taken so that the parts don't overlap each other to prevent short circuits. Once all the components are placed strategically, routing those components can be started. The main goal behind routing is to turn each of those gold air wires into top or bottom copper traces. Care should be taken to be sure that the traces don't overlap each other; if the traces do overlap each other, they should be on different sides of the board. Vias can be used if the trace needs to be moved to a different layer mid-way. After the routing is complete, Eagle's Design Rule Check function should be used and annotations should be done as parts of final touch ups. Once this is done, the design is ready to be transformed into a group of Gerber files and sent to the fabricator. For the each of the PCBs used in the Truck Smart system, seven Gerber files will be generated.

3.5.4 PCB Design Constraints

High quality PCBs make it possible for modern electronics to last many years between breakdowns. However, there are a handful of factors that affect the longevity and performance of a PCB. These factors are: leakage resistances, IR voltage drops in trace foils, vias, and ground planes, the influence of stray capacitance, and dielectric absorption (DA). In addition, the tendency of PCBs to absorb atmospheric moisture means that changes in humidity often cause the contributions of some parasitic effects to vary from day to day. All of these factors can be accounted for by proper grounding, partitioning, decoupling, thermal management, and tracing of a PCB. These different methods of making a PCB more reliable are discussed in this section.

3.5.4.1 Partitioning

Any PCB with both Analog and Digital signals should have them physically separated as much as possible to prevent crosstalk. Even though it's hard to achieve, crosstalk can be prevented by preventing different signals from interfering with each other. High level analog signals should be separated from low level analog signals, and both should be kept away from digital signals. For example, TTL and CMOS digital signals have high edge rates, implying frequency components starting with the system clock and going up from there. And most logic families are saturation logic, which has uneven current flow (high transient currents) which can modulate the ground. In addition, sharing drivers between clocks of different frequencies in the same package will produce excess jitter and crosstalk and degrade performance; therefore, only one frequency clock should be passed through a single package. The ground plane can act as a shield where sensitive signals cross.

Another thing to consider is to allocate extra ground pins to a PCB. The contact resistance of a single pin of a PCB connector is in the order of 10 mΩ when the board is new. However, as the board gets older, the contact resistance is likely to rise and the board's performance might be compromised. Therefore, it is recommended to allocate extra PCB connector pins. Using multiple ground pins helps keep down the ground impedance at the junction between the board and the backplane. There should also be several power connection pins for similar reasons. The figure below is a good example showing PCB layout where all necessary portions are isolated.

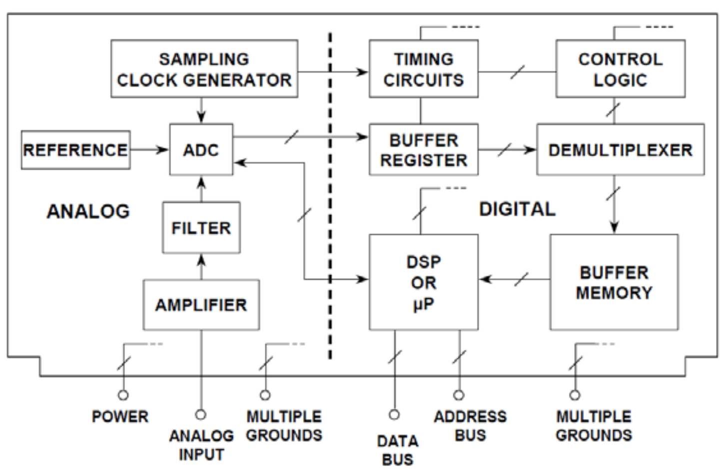
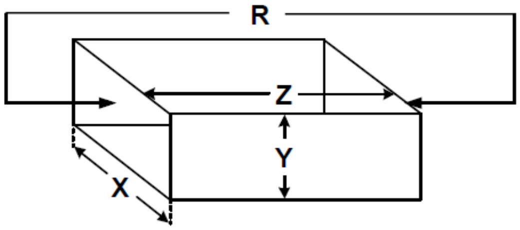


Figure 23 – Example PCB Partitioning

.

3.5.4.2 Tracing Resistance

Even though line resistances are disregarded in Macro electronic circuits, for high precision PCBs, even the relatively small trace resistances can have a drastic effect on the performance. PCB traces are most commonly made of copper – which at 25°C has a resistivity of $1.724X10^{-6}$ Ω/cm. The thickness of standard 1-ounce PCB copper foil is 0.036 mm, which based on the calculations below, accounts for a resistance of 0.48 mΩ/square. Total resistance of a linear trace can be calculated by effectively stacking a series of such squares end to end. If the line length is Z and the width is X, the line resistance R is simply Z/X times resistance of a single square.



$$R = \frac{\rho Z}{XY} = \frac{1.724e{-6}}{0.0036}\frac{Z}{X}\ m\Omega$$

Equation 2 – PCB Trace Resistance

Figure 24 – An Equivalent Single Square of Copper Tracing

PCB trace resistance can be a serious error when conditions aren't favorable. For example, consider a 16-bit ADC with a 5 KΩ input resistance, driven through 5 cm of 0.25 mm wide 1 oz. PCB track between it and its signal source. The track resistance of nearly 0.1 Ω forms a divider with the 5 kΩ load, creating an error. The resulting voltage drop is a gain error of 0.1/5 k (~0.0019%), well over 1 LSB (0.0015% for 16 bits). And this ignores the issue of the return path! It also ignores inductance, which could make the situation worse at high frequencies.

One widely used method of removing the voltage drop effects of tracing resistance is to use "Kelvin" or "Voltage Sensing" feedback. In this modification, a long resistive PCB trace is still used to drive the input of a high resolution ADC, with low input impedance. In this case, however, the voltage drop in the signal lead does not give rise to an error, as feedback is taken directly from the input pin of the ADC, and returned to the driving source. This scheme allows full accuracy to be achieved in the signal presented to the ADC, despite any voltage drop across the signal trace. The use of separate force (F) and sense (S) connections (often referred to as a Kelvin connection) at the load removes any errors resulting from voltage drops in the force lead, but, of course, may only be used in systems where there is negative feedback. It is also impossible to use such an arrangement to drive two or more loads with equal accuracy, since feedback may only be taken from one point.

From the discussion above, it can be concluded that when dealing with high precision circuits, even the minor things like PCB trace resistances cannot be ignored. There are various solutions that can address this issue, such as wider traces (which may take up excessive space), and may not be a viable solution with the smallest packages and with packages with multiple rows of pins, such as a ball grid array (BGA), the use of heavier copper (which may be too expensive) or simply choosing a high input impedance converter.

3.5.4.3 Tracing Inductance and Capacitance

While usually unaccounted for in purely DC circuits, inductance becomes an issue at high frequencies even for small pieces of copper or traces. Since the Truck Smart system will be using high frequencies for the sensor-hub communication, accounting for unwanted inductance is very necessary. The inductance of a PCB track can be approximated by the inductance of a strip conductor, given by:

$$\text{Strip Inductance} = 0.0002L\ [\ln\frac{2L}{(W+H)} + 0.2235\ \frac{(W+H)}{L} + 0.5\ ]\ \mu H$$

Equation 3 – PCB Strip Inductance

Using the formula above, a 1 cm of 0.25 mm PCB tracing's inductance can be approximated to be 9.59nH. Even though this inductance is relatively low, it can produce an impedance of 0.46 Ω at 10MHz and produce a high error for microelectronic circuits. Fortunately, the effects of inductance on a PCB can be minimized by placing the outward and return paths close together. By placing these paths together, the overall area enclosed by conductors is reduced which also limits the magnetic field created by it, limiting unwanted interactions with other circuits on board. The PCB schematic in the hardware design section of this report reflects this technique of limiting the enclosed area.

While inductance affects the instantaneous performance of a PCB, dynamic effects caused by stray capacitance can lead to permanent damage of a PCB. Stray capacitance is caused when two conductors aren't screened from each other or short-circuited together. Hence, a multilayer PCB is affected by multiple layers of stray capacitance and limiting it is necessary for high performance circuits. Stray capacitance becomes really big of an issue when high resolution data converters are connected to high speed data bus with a sharp edge rate. In this case the noise is easily connected to the converter analog port via stray capacitance across the device. The same issue also limits performance from other broadband monolithic mixed signal ICs. Tracing capacitance can be approximated by the formula:

$$0.00885 * \mathrm{Er} * (\mathrm{A/d});\ \text{where Er} = \text{Dielectric constant}$$

Equation 4 – PCB Trace Capacitance

Using this formula, the capacitance between conductors on opposite sides of the board is calculated to be about 3 pF/cm^2. Since stray capacitance is linked to lead and component placement, an efficient way to minimize it is to use correct PCB layout. While proper component placement is effective for stray capacitance, noise coupling effect of an actual capacitor on board can be reduced by using Faraday shield. Faraday shielding a capacitor means inserting a grounded conductor between the noise source and affected circuit. To be fully effective, the Faraday shield must completely block the electric field between the noise source and the shielded circuit. For high resolution data converters, COMS buffers are used as a Faraday shield.

3.5.4.4 Grounding

While the basic concept of grounding is relatively simple, implementing it is very involved making it one of the most difficult aspects of PCB design. For linear systems the ground is the reference against which we base our signal. Unfortunately, it has also become the return path for the power supply current in unipolar supply systems. Improper application of grounding strategies can destroy high accuracy linear system performance. Proper grounding is essential for all analog designs, including PCBs. Fortunately, certain principles of quality grounding, namely the use of ground planes, are intrinsic to the PCB environment. This factor is one of the more significant advantages to PCB based analog designs, and appreciable discussion of this section is focused on this issue. Some other aspects of grounding that must be managed include the control of spurious ground and signal return voltages that can degrade performance. These voltages can be due to external signal coupling, common currents, or simply excessive IR drops in ground conductors. Proper conductor routing and sizing, as well as differential signal handling and ground isolation techniques enable control of such parasitic voltages.

Today's signal processing systems generally require mixed-signal devices such as analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) as well as fast digital signal processors (DSPs). Requirements for processing analog signals having wide dynamic ranges increase the importance of high performance ADCs and DACs. Maintaining wide dynamic range with low noise in hostile digital environments is dependent upon using good high speed circuit design techniques, including proper signal routing, decoupling, and grounding. To further complicate the issue, mixed-signal ICs have both analog and digital ports, and because of this, much confusion has resulted with respect to proper grounding techniques. In addition, some mixed-signal ICs have relatively low digital currents, while others have high digital currents. In many cases, these two types must be treated differently with respect to optimum grounding. A few commonly used methods to overcome the grounding issue are listed in this section below:

1. Star Ground

The "star" ground philosophy builds on the theory that there is one single ground point in a circuit to which all voltages are referred. This is known as the *star ground* point. Note that the star point does not need look like a star; it may be a point on a ground plane. The key feature of the star ground system is that all voltages are measured with respect to a particular point in the ground network, not just to an undefined "ground". This star grounding philosophy is reasonable theoretically, but is difficult to implement practically. For example, if a star ground system is designed, drawing out all signal paths to minimize signal interaction and the effects of high impedance signal or ground paths, we often find implementation problems.

2. Separate Analog and Digital Grounds

When the power supplies are added to the circuit diagram, they either add unwanted ground paths, or their supply currents flowing in the existing ground paths are sufficiently so large, or noisy (or both) so as to corrupt the signal transmission. This particular problem can often be avoided by having separate power supplies (and thus separate ground returns) for analog and digital portions of the circuits. However, these separate grounds can later be joined at the star point.

3. Ground Planes

To implement a ground plane, one side of a double-sided PCB (or one layer of a multilayer one) is made of continuous copper and used as ground. The theory behind this is that the large amount of metal will have as low resistance as is possible. It will, because of the large flattened conductor pattern, also have as low an inductance as possible. It then offers the best possible conduction, in terms of minimizing spurious ground difference voltages across the conducting plane. Note that ground plane concept can also be extended to include voltage planes. A voltage plane offers advantages similar to a ground plane, i.e. a very low impedance conductor, but is dedicated to a one (or more) of the system supply voltages. Thus a system can have more than one voltage plane, as well as a ground plane. Ground planes also allow the transmission of high speed digital or analog signals using transmission line techniques (micro strip or strip line) where controlled impedances are required.

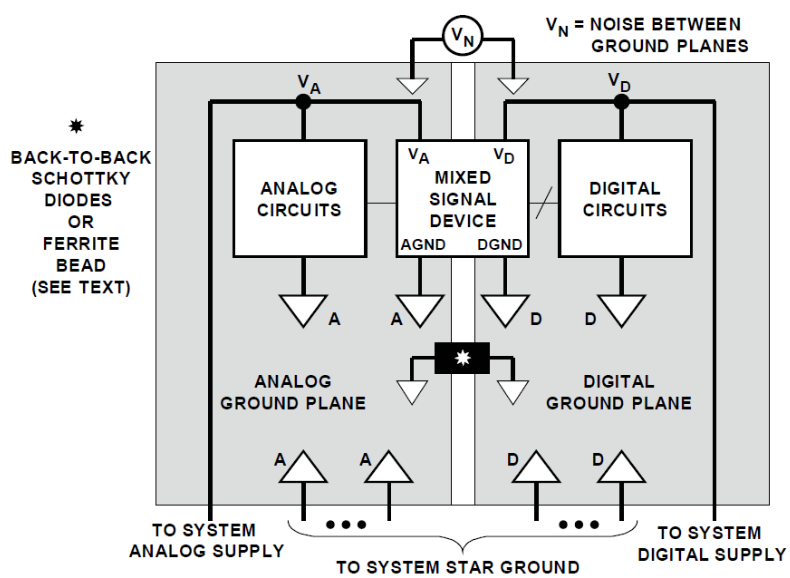


Figure 25 – Advanced Grounding Pattern

An alternative grounding method for a mixed-signal device with high levels of digital currents is shown in the figure above. The AGND of the mixed signal device is connected to the analog ground plane, and the DGND of the device is connected to the digital ground plane. The digital currents are isolated from the analog ground plane, but the noise between the two ground planes is applied directly between the AGND and DGND pins of the device. For this method to be successful, the analog and digital circuits within the mixed signal device must be well isolated. The noise between AGND and DGND pins must not be large enough to reduce internal noise margins or cause corruption of the internal analog circuits.

3.5.4.5 Decoupling

It is highly recommended to use decoupling capacitors for connecting individual analog stages together on a PCB (these stages are usually provided directly over power pins so this capacitor will need to be connected on the power trace). Decoupling shunts all the high frequency noise away which results in less noise at the output. This is achieved efficiently using capacitors because the power supply rejection ratio drops with the frequency. Decoupling should be done using a small capacitor (typically 0.01 μF to 0.1μF) with good high frequency characteristics; surface mount ceramic capacitors are ideal for this application. The figure below shows a good technique of connecting a decoupling capacitor to the PCB. As shown in the figure, a typical 0.1 μF ceramic capacitor should go directly to the ground plane rather than going to the ground of the IC. Decoupling capacitors should be used in all ICs operating at a frequency higher than 10 Mega Hertz. The Ferrite beads shown in the figure below are optional and add extra High Frequency noise isolation.

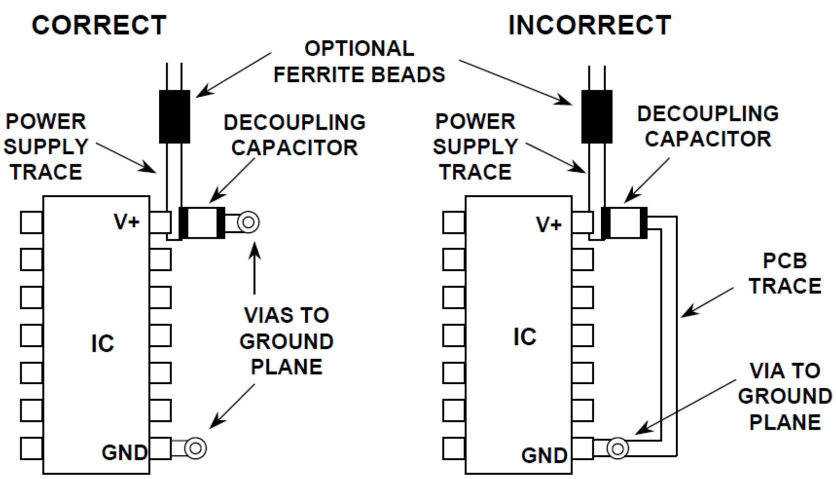


Figure 26 – Decoupling Capacitor Connection in a PCB

In addition to the high frequency capacitors, it is also recommended to use power capacitors for ICs that have instantaneous current requirements. The purpose behind this is to provide a local reservoir of charge so that instantaneous current demands can be provided from a local source, rather than the power supply that's relatively far away and is subjected to tracing resistance. This functionality is achieved by using larger electrolyte capacitors between 10 μF to 100 μF.

3.5.4.6 Propagation Delay

Table 10 – Clock Frequency vs Path Length

Clock Frequency	Path Length
1 MHz	70 Inches
10 MHz	10 Inches
100 MHz	0.7 Inches

Propagation delay is defined as the length of the time taken or the quantity of interest to reach its destination. It's caused when multiple ICs are connected to one clock using physically long signal path. This delay can cause timing skew between outputs of clocked devices. This issue is illustrated in the figure below. In portion A of the figure it can be seen that path length from the clock source to the clock input of U_1 is significantly shorter than that of U_n. This deficiency causes skewed outputs as shown in part B of the figure. Part C of the figure shows path length equalization technique to deal with this issue. In general, path equalization is necessary when the difference between the lengths of the signal path from the clock to the closest clocked device is greater than the values listed in table 10.

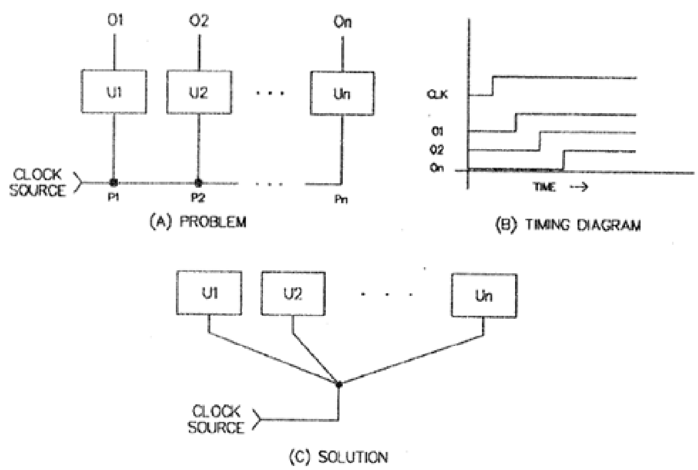


Figure 27 – Eliminating Timing Skew by Equalizing Clock Path Lengths

3.5.4.7 Thermal Management

PCBs are generally small and compact circuits so even relatively small amount of power dissipation can cause high on-board temperature. All semiconductors have a specified upper limit for junction temperature (T_J) that shouldn't be exceeded. And since semiconductor lifetime is inversely related to the operating temperature, the cooler the ICs are, the longer their lifetimes will be. Therefore, proper thermal management is very crucial for PCBs.

Thermal resistance (θ) is the unit used to calculate the heat dissipation of an IC; it's measured in °C/Watt. A device with a thermal resistance θ equal to 100°C/W will exhibit a temperature differential of 100°C for a power dissipation of 1 W, as measured between two reference points. The two reference points are usually the junction and air near the IC. This temperature is used to predict the internal temperature. θ_{JC}, θ_{CA}, and θ_{JA} are thermal resistances between Junction – Case, Case – Ambient, and Junction – Ambient respectively. The effective temperature differential can be calculated using the equation $\Delta T = P \times \theta$; where T is in °C and P in Watts. These thermal resistances are highly package dependent and are analogous to electrical resistances. Hence, copper frame packages offer lower θ (higher performance) than aluminum or steel.

Heat sinking is the most common method used for lowering an IC's temperature. Heat sink is an external low thermal resistance device attached to the IC to aid heat removal. Heavy copper blocks with large areas are preferred (if practical for the application). While externally attached heat sink devices are commonly found in older electronics, modern PCBs often use low thermal resistance copper traces as heat sinks and that's what will be used for the truck smart system as well. These copper traces can be very useful because higher power dissipation is possible for ICs that are better able to transfer heat from the chip to PCB. The figures below show temperature vs. max power dissipation graphs without a heat Sink (left) and with a heat sink (right). Devices meant for heat sink attachment usually have a θ_{JC} dramatically lower than the θ_{JA}. In this case θ is composed of more than one component. The glue used to embed the heat sink on IC must be electric insulator to ensure safety.

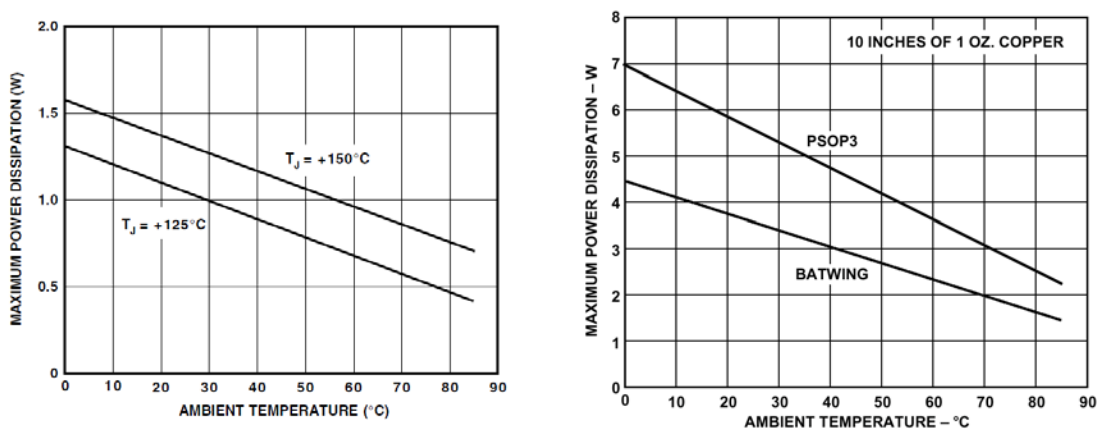


Figure 28 – Temperature vs Max Power Dissipation with and without Heat Sink

In addition to using heat sinks, providing sufficient ventilation inlets and outlets allows heat to freely move away from hot PCB surfaces. Placing high power dissipating PCB planes vertically aids convection airflow across heat sink areas.

4.0 Design Constraints and Standards

This section deals with the impact of realistic design constraints and how it affects the design of the Truck Smart system. It will also identify and review any standards related to the technology used in the system design.

4.1 Constraints

The Truck Smart project includes various hardware and software design constraints that needed to be addressed while designing the system.

4.1.1 Economic Constraints

The cost of the designing the system is a major constraint due to various reasons. One of the biggest reasons is that, unlike in previous and future semesters at UCF, Senior Design this current semester did not receive any general funding from any companies. Whereas for earlier projects businesses such as Lockheed or Boeing would provide up to a thousand dollars per group for senior design, the Truck Smart team will have to entirely fund the project themselves. This severely inhibits the testing capabilities of the group and means that the team will have to do a lot of research as opposed to regular trial and error to determine which components, such as sensors (Sonar/IR/Electromagnetic/other), would work the best for the system. Buying different kinds of components to test them will increase cost of the prototype which will burden the team financially.

Another economic constraint is the fact that the cost of manufacturing needs to be kept as low as possible for marketing potential reasons. The Truck Smart system needs to be affordable to its users and a higher cost to make the product will result in a higher MSRP. This will negatively affect the marketability of the product in a colossal way. In an ideal situation, the team would use the highest quality parts with the maximal amount of features to design the system. But due to this economic constraint, a balance must be achieved between cost and quality.

4.1.2 Environmental Constraints

Renewable energy and environment friendly technology is a rapidly growing market. In addition to the potential sales pitch, everyone is responsible for respecting the environment. The Truck Smart system has no part or function that will extensively affect the environment in a negative way. However, the system will need to use electricity to work. In the original design, this power is planned to come from a combination of the truck's battery and secondary batteries built into the system. While not a primary goal, research will be done into sources of alternative energy such as solar or wind. If, after extensive research, it is found that the system would be able to be powered by alternative energy without any negative consequences such as cost or time, it will be implemented into the system.

4.1.3 Social Constraints

The goal of the Truck Smart system is to increase the safety of every driver on the road. To accomplish this, the system should be readily available to as many people as possible. This will be done by making the product as affordable as possible and not limiting it to only select users (such as only to those living in a certain area or to only to those part of a certain organization).

Another social constraint comes into play when designing and testing the installation ease of the system. A very important feature of this system is that it shall be easily mountable on the truck body. Figuring out a technique to do so using readily available materials would be a challenge. The members of the Truck Smart system are all above average height and would have no trouble reaching a higher part of the truck. When creating the installation method, people of shorter stature should be considered to make sure it is equally accessible for everyone.

4.1.4 Political Constraints

After completing thorough research of possible political constraints, it was decided that any potential political constraint was not actually relevant to the Truck Smart system.

4.1.5 Ethical Constraints

The foremost purpose of this product is to increase the safety of everyone on the road. No corners will be cut on any aspect of the system that will in any way negatively affect the end user, including drivers around the end user. The Truck Smart system will not use materials that are potentially toxic or affect the lifetime of the product to lower production costs. Furthermore, no features will be cut from the product solely for financial reasons.

When it comes to patent protection, comprehensive research will be done on existing patents to make sure the Truck Smart system does not infringe on any of them. Other protected designs or concepts will not be copied or used without giving properly attributed credit if applicable.

4.1.6 Health and Safety Constraints

The health and safety of the end users of the Truck Smart system and those around them have the utmost priority. The product’s main purpose is to alert drivers about the current status of their blind spots. Careful consideration needs to be taken that this information is updated as quickly as possible. Even something as small as just a one second delay could be disastrous and life threatening as vehicles on highways routinely travel over a base speed of 70 miles an hour. This means in that one second vehicles over a hundred feet, more than the length of entire trucks. This calculation does not even take into account people that speed to more than double the speed limit, traveling almost at 140 miles an hour. Not only will this greatly affect the safety of the end user, but it will also directly affect the safety of the general public even more so. If a truck driver thinks it is clear to switch lanes while in reality there is a vehicle in their blind spot, it is that vehicle that is in the most danger. A truck pushing you out of your lane could cause you to veer off the road into uncharted territory or could even push you into the way of other oncoming traffic. For these reasons, the delay of information transmission from the sensor to the hub unit must be as short as possible.

The same dangers mentioned above in the previous paragraph could also be caused by faulty equipment such as a sensor that stopped working or a wireless transceiver that malfunctioned. To avoid these scenarios and protect the safety of the driver and those around him, only parts that are reputable and have been thoroughly tested by the community will be used in the Truck Smart system. However, regardless of the quality of the parts used, malfunctions are an unavoidable truth when it comes to mass manufacturing. To account for this, the system will continue to monitor sensor data outputs and will notify the user as soon as it stops receiving data from a sensor unit, whether the problem is due to a malfunctioning sensor or a faulty wireless transceiver.

Each part of the Truck Smart system will be need to powered by some source, whether by solar or otherwise, and will need a battery to store this energy. A danger that comes with batteries is the explosion a malfunctioning battery can produce. To avoid this, the team will learn as much as possible about battery technology and implement the method that will least likely result in a malfunction. Toxic material that could potentially poison the user or any of the workers creating the product will also be avoided.

While an idea for an audible alert for blind spot warnings was discussed, it was decided that it would actually be counterproductive and serve only to distract the user on busy roads. This resulted in a system that has no audible sounds whatsoever, meaning this project is not limited by any audio constraints such as loud noises that could result in hearing loss. Another health and safety constraint that does not apply are the extra considerations that need to be taken when creating products for infants or children. The Truck Smart system is solely intended for use by only adults as only adults will be driving the large truck.

4.1.7 Manufacturability Constraints

When creating the Truck Smart system, one manufacturing constraint to take into account is the availability of the chosen materials. For example, when deciding which sensor or microcontroller to use, the team will take into consideration how long that particular piece of technology has been available to consumers and how much longer it is expected to be available. Another factor to consider is the amount of parts the manufacture has in stock. If there are thousands available, it should not be an issue. But if there are only a hundred or fewer remaining, getting enough parts to mass manufacture the system could be an issue.

Another constraint to consider is the availability while prototyping. Certain types of material will be out of reach of the Truck Smart team either due to lack of equipment or due to a lack of funds. To account for this, the housing of the various system parts will be 3-D printed at one of the various labs provided by the University of Central Florida. This will enable to team to create a representative prototype while also adhering to the manufacturability constraint.

4.1.8 Sustainability Constraints

The sustainability goal of the Truck Smart system is to be able to guarantee a life span of at least 100,000 miles under the assumed normal operation conditions. The sustainability constraints for this lie in the consideration of environmental factors and the consideration of the survivability of the business.

With the exception of the hub unit, the system will be completely exposed to the elements. To be able to function successfully on a truck, the system should be able to withstand the blazing hot temperatures found in the southern United States that reach over a hundred degrees Fahrenheit. It should also be able to withstand the freezing temperatures found in the northern United States that reach below zero degrees Fahrenheit. When deciding which parts to use in the system, the temperature of the operating conditions need to be taken into account for this constraint. Beyond temperature, the housing of the sensor units also need to be able to successfully withstand winds over a hundred miles an hour. For reference, that is almost the speed of a category 3 hurricane in the Saffir-Simpson wind scale. This wind comes from the fact that the trucks using the system will regularly be traveling at rapid highway speeds. In addition to high and low temperatures and winds, the system also needs to be able to survive in extreme weather conditions. This includes things such as rain, snow, and hail. The system will have to be waterproof while not getting in the way of the sensor. The material of the housing also needs to be strong enough to withstand at least light hail. Lastly, the sensor units should not protrude out too far from the truck. If it does hang too much off to the side, something could possible hit it and damage. For example, a car or another truck that was driving too close could hit the sensor unit with their side mirror. So to successfully overcome this constraint, the housing needs to be designed not only to protect the fragile electrical components from this weather, but also designated in a way that the housing will not be damaged and will last the expected lifetime of the system (100,000 miles).

The hub unit will be situated inside the truck cabin alongside the driver. For this reason, temperature and extreme weather conditions do not apply to the hub unit. However, there are still various sustainability environmental constraints to consider. A truck cabin can be an extremely busy area as it basically functions as the trucker's home away from home. The hub unit needs to be designed so that it can be placed in the cabin of any truck, regardless of cabin design. It also needs to be located in such a way that it is clearly visible to the user while driving but also in a that it does not impede any of the driver's other functions. It must not be placed in a location that pulls too much attention away from the road, a balance must be found. To overcome this constraint, careful research and consideration must be taken while designing the hub unit.

Another sustainability constraint is the survivability of the product itself, not in terms of how long it will work but in terms of how long the product will remain a viable business item. When it comes to transportation, safety and security is not only among the top considerations for consumers, but also for the businesses themselves. Truck Smart is a product that makes the roads safer not only for trucks but also for every vehicle that has to drive near a truck. This appeals to every consumer, both truckers and drivers of regular cars. Moreover, the advancement of technology and self-driving cars have reached a point where automation of both trucks and cars are possible. Truck Smart is an affordable solution that will stall the progress of automation by enabling the trucking industry to increase the safety of their drivers.

4.1.9 Time Constraints

The time constraints could actually be considered the most important type of constraint. In the end, the number one priority of the Truck Smart project is to have it completed by the end of Senior Design 2. This gives the team two semesters, roughly eight months, to complete the entire project. The time constraint should be first and foremost when taking any design ideas into consideration. There are potentially a multitude of features that could be added to the Truck Smart system, such as solar power capability. However, features such as these that add extra research and development time will not be considered primary goals, only secondary. This means that the features will attempt to be added only if there is enough time after the primary features have been added.

To assist in conquering the time constraint, a schedule will be devised and followed to the best of the team's ability. This schedule can be seen in the milestone section under the administrative chapter. When managing time, the PCB is important to consider. It is not enough to only consider time to create and order the PCB. There must be additional time scheduled for PCB testing and more time for redesigning and reordering the PCB as necessary.

4.1.10 Testing/Presentation Constraints

The Truck Smart development team is severely limited when it comes to testing and showcasing the product. The reason is because the system is designed for use by large trucks and one will be needed to properly test and showcase the functionality of the device.

The team has contacted a trucker who has agreed to let them use a truck for testing. However, the team will have a limited access to this truck, making it harder to schedule the testing periods. The trucker will need to be available in the same time frame that the product is ready for real world testing but also needs to be a decent enough time before the product presentation date. This is to ensure that there is enough time to work out any problems that are discovered while testing. Another method of testing includes the use of multiple vehicles on a safe and closed course. The device functions would temporarily be adjusted to work with the much smaller dimensions of a mid-size car and tested in this form. While not perfect, this method of testing is readily available. To present the product, the team will build a smaller scaled truck trailer using wooden planks. The device calculations will again be adjusted to match the smaller dimensions of the display.

4.2 Standards

There are many sets of standards and government regulations to be aware of prior to designing the Truck Smart system. Wireless communications and general RF signals are subject to frequency spectrum allocation to prevent interference between users, systems, and applications. There are also standards for microelectronics and power supplies.

4.2.1 Power Supply Standards

The Power Supply Safety Standards (as provided by CUI) encompass every aspect of a system, from the components allowed to be used for a certain circuit classification, to circuitry insulation for shock prevention. The circuit classifications are generally defined in Table 11.

Table 11 – Circuit Classifications

Circuit Type	Type Definition
Hazardous Voltage	Any voltage exceeding 42.2 VAC peak or 60 VDC without limited current circuit.
Extra-Low Voltage (ELV)	Voltage in secondary circuit not exceeding 42.4 VAC peak or 60 VDC. Circuit should be separated from hazardous voltage by basic insulation.
Safety ELV	Secondary circuit unable to reach hazardous voltage, even while experiencing a fault. During fault, ELV limits apply. Absolute limits of 71 VAC and 120 VDC must not be exceeded. Must be double-insulated from hazardous voltages. Considered safe for operator access.
Limited Current Circuit	Circuit designed to ensure that under fault condition, the current able to be drawn is not hazardous. AC frequencies < 1 kHz, steady state shall not exceed 0.7 mA peak AC or 2 mA peak DC. Above 1 kHz, limit of 0.7 mA is multiplied by the kHz frequency, not to exceed 70 mA. For accessible parts not exceeding 450 VAC & VDC, maximum circuit capacitance is 0.1 uF. For accessible parts not exceeding 1500 VAC & VDC, maximum stored charge is 45 uC and available energy should not exceed 350 mJ.

The circuitry design for the Truck Smart system will fall under the Extra-Low Voltage (ELV) circuit type. A maximum voltage of 7.4V will reside at the battery terminal and will be attenuated throughout the remainder of the circuit. Basic insulation will be included on the wiring from the battery. The printed circuit board (PCB) will include plastic shielding to prevent short circuits on any surrounding metal within the Truck Smart peripheral sensor housing.

Figure 29 – Conformité Européenne

Meeting these standards is an important factor to consider in the design phase of the Truck Smart system. In the event Truck Smart is brought to the public market, there is a liability held by the designers to produce a product that functions properly by providing Truck Drivers with a “sixth sense” of situational awareness. Therefore, when microelectronic standards are met, there is an assumption made by the consumer that Truck Smart has been well-engineered and the system does not pose a factor of risk during operation. This assumption can be formalized by the manufacturer through the use of a variety of consumer goods standards markings. One of the well-recognized markings in the consumer electronics industry is the “Conformité Européenne”, also known as the “CE” marking (shown in Figure 29).

4.2.2 IEEE 802.15.4 Standard

The Institute of Electrical and Electronics Engineers (IEEE) is a professional association whose objective is “the education and technical advancement electrical and electronic engineering”. This organization is responsible for create a wide array of worldwide standards for electronics. Specifically relating to the Truck Smart project, IEEE 802.15 is a standards committee which specifics wireless personal area network standards. The technology used in the Truck Smart system, ZigBee, falls into a subsection known as 802.15.4, Low-Rate

Wireless Personal Area Networks. 802.15.4 focuses on technology that is low-power, low-speed, and low-cost. This is exactly the kind of device that is needed for the Truck Smart system.

4.2.2.1 Operating Frequencies

Table 12, seen below, shows the frequency options open to the Truck Smart project. Currently, development is focused solely on North America. However, these standards need to be taken into consideration if the product is to ever be deployed on a global scale. Based on this table, there are 2 ranges of frequencies available for use and both will be extensively covered during the research and testing phase to choose the best option for the Truck Smart project. Following IEEE 802.15.4 standards, the 902-928 MHz range is known as 915 MHz and the 2400 – 2483.5 MHz range is designated as 2450 MHz or 2.45 GHz.

Table 12 – Frequency Availability

Region	Frequency (MHz)	Channels	Bandwidth (MHz)	Data Rate (kb/s)
North America	902 – 928	10	2	40
Europe	868 – 868.6	1	0.6	20
Global	2400 – 2483.5	16	5	250

4.2.2.2 Network Security

According to IEEE, implementing device security is option. However, the standard to follow is that if the device does not implement security, it must not have “a mechanism to perform a cryptographic transformation on incoming and outgoing frames.” On the other hand, if the device does have security, it must implement it on both outgoing and incoming frames. The Truck Smart system is indeed planned to have network security enabled. For this, the IEEE standard states that certain values need to be set to a specific value. More specifically, “SecurityLevel” needs to be set to “secAutoRequestSecurityLevel”, “KeyIdMode” needs to be set to “secAutoRequestKeyIdMode”, and finally “KeySource” needs to be set to “secAutoRequestKey Source.

4.2.3 DoT Lane Widths Standard

The most critical component of the Truck Smart system is the sensor that is used for blind sport detection. For this sensor to work properly, it will need to be calibrated to work at certain distances. For example, if the sensor detects an object that is 20 feet away, it should not flag that as an object in the blind spot area. However, how close much the object be for it to be in the blind spot? This is where the lane width standard comes in. It is a standard this is set by the American Associate of State Highway and Transportation Officials (AASHTO).

This is an organization that coordinates with the Department of Transportation to set standards on road design in the United States.

Table 13 – Standard Lane Widths

Type of Roadway	Lane Width (feet)
Local	9 – 12
Collector	10 – 12
Arterial	11 – 12
Freeway	12
1-Lane Ramps	12 – 30

As seen in the above table, lane widths can range anywhere from 9 to 30 feet. However, that is including one lane ramps. Since it is a one lane road, blind spots should not be an issue as there should not be any vehicles adjacent to the user. For that reason, the dimensions of the one lane ramp will be excluded when taking road widths into account for sensor detection. The widths of relevant lanes now range from 9-12 feet. Thanks to this information from the Department of Transportation and the AASHTO, the sensor calibration can be expected to work on all roads in the United States that follow regulation.

4.2.4 UL 94 - Standard for Tests for Flammability of Plastic Materials for Parts in Devices and Appliances

Even though Truck Smart system will be an ELV – Extra Low Voltage system, the PCB is a thin, plastic based material so standards should be followed to ensure flame resistance of the system. Due to this reason, the PCBs used in this system are designed in accordance with the UL 94 standard. UL 94 is a plastic flammability standard released by Underwriters Laboratories of the United States. The standard provides a method for rating the ignition characteristics of plastic materials. Two UL 94 ratings that code officials commonly run across are HB and V (V-0, V-1, or V-2). These ratings are established using small-scale tests in which approximately 5 by ½ inch samples are subject to a ¾ inch, 50W Tirrell burner flame ignition source. To achieve a HB rating, test samples, placed horizontally, burn slowly across the sample when the test flame is applied to the end of the sample. To achieve a V rating the test samples, placed vertically with the test flame impinging on the bottom of the sample, must extinguish within specified times, not burning to the top clamp or dripping molten material which ignites a cotton indicator. The tests in the Standard for Tests for Flammability of Small Polymeric Component Materials should be used to evaluate small components, like the PCBs in this system, which contain materials that cannot be fabricated into standardized specimens in the minimum use thickness and subjected to applicable preselection tests in UL 94. Test procedures in UL 1694 are applicable to small components with an overall volume of less than 2500 mm3 (0.15 in3). UL 1694 is generally not applicable to small components with an overall volume greater than 2500 mm3 (0.15 in3).

The FR4 Substrate of PCBs used in this system meets the requirements listed by UL 94 standard. Meeting this standard in addition to the power supply standards listed in the previous section makes the Truck Smart system safe for public use in normal road conditions.

4.2.5 IPC PCB Standards

IPC, the Association Connecting Electronics Industries, is a trade association whose aim is to standardize the assembly and production requirements of electronic equipment and assemblies. It regulates a lot of standards that commercial PCBs need to follow to maintain reliability and longevity of the products. These standards apply to all printed circuit board types including single-sided, double-sided and multilayer. This range of standards includes general documents, design specifications, material specifications, performance and inspection documents, and flex assembly and material standards. Out of these standards IPC produces, IPC-221B focuses on printed board design; which lays out the generic requirements for PCB design and component mounting. The requirements and how they affect design of Truck Smart system are listed more in depth in the PCB research section of this report. The IPC-2220 standard covers PCB design in CAD software. This series of standards is built around the IPC-221B standard and covers the generic requirements for printed circuit board design in the software. IPC-2223 covers Sectional Design Standard for Flexible Printed Boards. Other standards in this range also focus on tracing and spacing between the PCB boards, the current bearing requirement of the traces, conductor thickness, material properties, and tolerance rules.

These IPC standards should be followed because they help the design achieve desired results. These standards gain control over end product quality and Reliability which are the cornerstones of competing in the marketplace and critical to the product’s reputation and profitability. By implementing IPC standards throughout the manufacturing process, better performance, longer life and compliance with lead-free regulations can be ensured. Working from an established IPC standard helps everyone "speak the same language" — the language of the global electronic industry. In addition, using IPC standards eliminates confusion for employees, because they know they need to perform to an established industry standard. And lastly, by ensuring that the design complies with IPC standards allows the designer to produce electronic assemblies that meet stringent quality tests down the line, minimizing costly delays, rework and scrap.

5.0 Project Design

The system's design is distributed into 4 major sections. These sections include software implementation, power distribution, sensor application, and printed circuit board design.

5.1 Hardware Design

The system's hardware components include the microcontroller, wireless transceivers, printed circuit board. The XBee shield will be replaced with a PCB design that will incorporate all the components into a single system.

5.1.1 System Design and Schematics

The following diagrams represent a layout of the system and its components as well as the assigned implementations for each of the members. The roles of the members were decided not only by their majors, but also by topic interest and previous experience.

5.1.1.1 Block Diagrams

The hardware block diagram begins by describing the relationship in between the battery-powered microcontroller, and sensor. The use of batteries allows for the system to be portable from one trailer to another. The sensor's reading is sent across the network to the hub device located in the truck's cabin. In order to send the data wireless, the use of the XBee S1 transceiver is required. This device is affordable, reliable, and efficient. On the bottom section of the hardware block diagram, the receiving device consists of another microcontroller with a wireless transceiver attached to it. The data coming in from multiple sensors is decoded and displayed to the driver. The LED display is the system's output source for the all of the sensor's statuses.

The team of four was split into two subsections, the computer engineering (software) team and the electrical engineering (hardware) team. The main objective of the software team is to enable communication between the hardware and the microcontroller and to successfully transfer that information wirelessly to the hub unit using the wireless transceivers. The hardware team has two main objectives. One is to design and implement the power supply of the system. The other is to create and design the printed circuit board the final product will use in place of the developer's board used in the prototype. Since the teams were split evenly, each task will be completed and managed using the buddy system. One member will take point and lead the task while the secondary member will provide as much support as possible and do whatever is needed to ensure the task is completed on schedule.

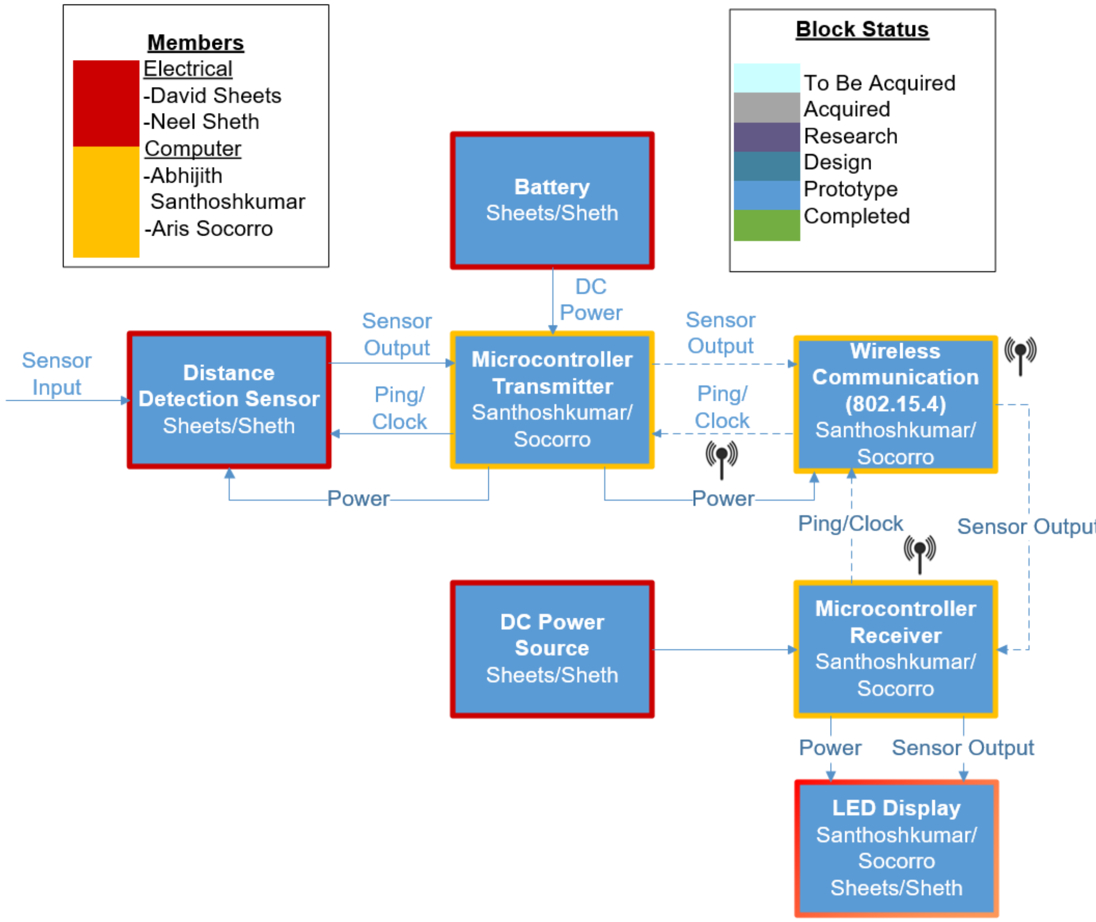


Figure 30 – Hardware Block Diagram

The software block diagram begins by assigning the network parameters such as private area network identification (PAN ID), network security key, trust center key, configuring the coordinator and router devices, and assigning the API module as wireless encoding standard. After the network has been set up, the sending and receiving algorithms are applied to the corresponding devices. Each of the algorithms uses a different structure and set of libraries. For the sending software configuration, the algorithm begins by taking the sensor's reading. If the reading states that there is a vehicle in the blind spot, then the sensor's reading is sent across the network using the XBee's libraries functions. The receiving algorithm implements functions from the same library but for different purposes such as decoding API frames.

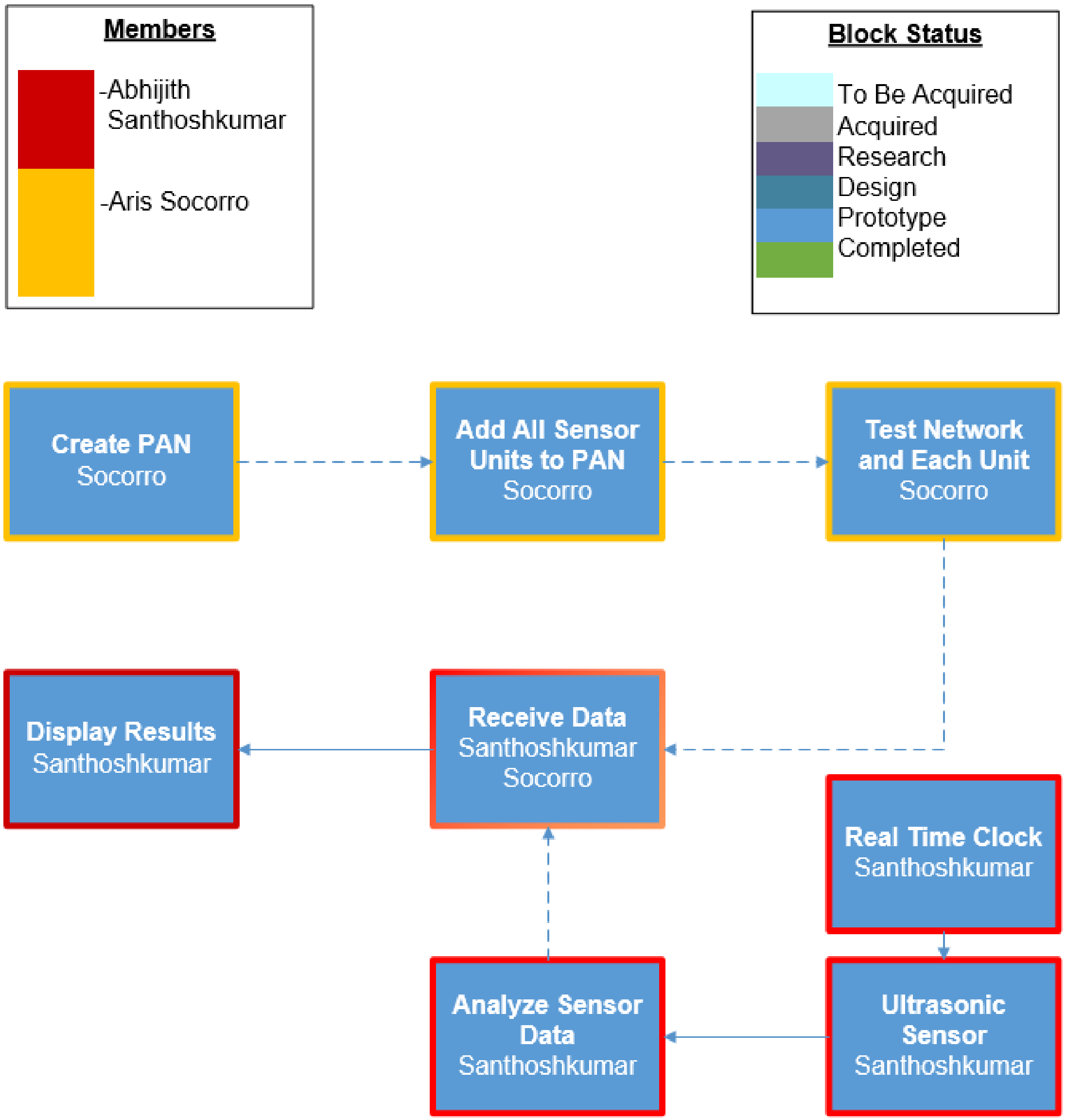


Figure 31 – Software Block Diagram

5.1.1.2 Prototype Build

As of early December 2016, the designing team has performed dozens of hours of research and development individually, as well as together as a group. A test bed has been set up using two Arduino Unos. The goal is to design a printed circuit board that contain the embedded microprocessor in the Arduino Uno, as well as directly support the XBee wireless transceiver, while being able to power the sensors at the same time. The reason for using the chip contained in the

Arduino Uno as the one used in the PCB, is to be able to load any Arduino code and the device will be able to understand and perform the functionality within the algorithm. This will be a challenging task and should the team not be able to accomplish it before the final project's deadline, there is a backup plan that will include using the regular Arduino microcontrollers as the sending and receiving devices. However, buying an Arduino Uno for each of the sending devices and the base will make the project more expensive than projected. The team has performed an extensive research on sensors and has decided that the HC - SR04 ultrasonic sensor will be the one used for the time being. This choice is based on the sensor's easy compatibility with the Arduino devices, its long range of up to 13 feet which is a lot more than the required distance for the system, and the many available libraries and functions that will help the programmers reduce the code complexity and increase flexibility at the same time.

The developing team has begun testing several critical components such as the sensors, the XBee USB explorer device (which allows for modifying the networks parameters for all the transceivers), the XBee S1 wireless transceivers, the XBee Arduino Shield, and the Arduino Uno microcontroller with all the devices attached to it. All the initial tests have been successful and the system is performing successfully considering that it is at its initial stage of development. The team was able to successfully test the sensor using an LED installed in a breadboard. Whenever a person approached within close proximity of the sensor, the blue LED in the breadboard blinked. In addition, the team successfully sent packets in a network that consisted of a coordinator and a router device. This network was implemented by the team members during a group meeting. The network operated in a fully encrypted environment. To fulfill this task, the team made use of the previously described network key as well as employing the trust center key to distribute the encrypted network key to the single router device. The network was fully encrypted and only the router and the coordinator devices were able to send or receive data within each other while no other device was allowed to join the network unless it knew the network encryption key.

Even though the network only consisted of 1 router device sending wireless data to a single coordinator module, it represented an excellent starting point to get the team familiar with how the data is handled when sent across the network and how to set the required parameters and algorithms to achieve this task. Moving forward, the team's next goal will be to integrate multiple sensors into the system and have all of them sending data to a single coordinator device. The biggest challenge lays in organizing and displaying six to eight data packets all coming in at the same time. The wireless transceiver's range appears to not be much of a constraint since the module used was the least expensive and the team was able to send data at over 250 feet. The following figure shows all the components before being merged together as one system. From left to right the components are 2 Arduino Unos, 2 XBee Shields, 2 XBee S1 Wireless Transceivers, 2 HC – SR04 Ultrasonic Sensors, 1 XBee USB Explorer Board, 1 breadboard used to light up the LEDs if the sensor activates, and 1 set of batteries used to initially test how long the system can last while attached to a truck.

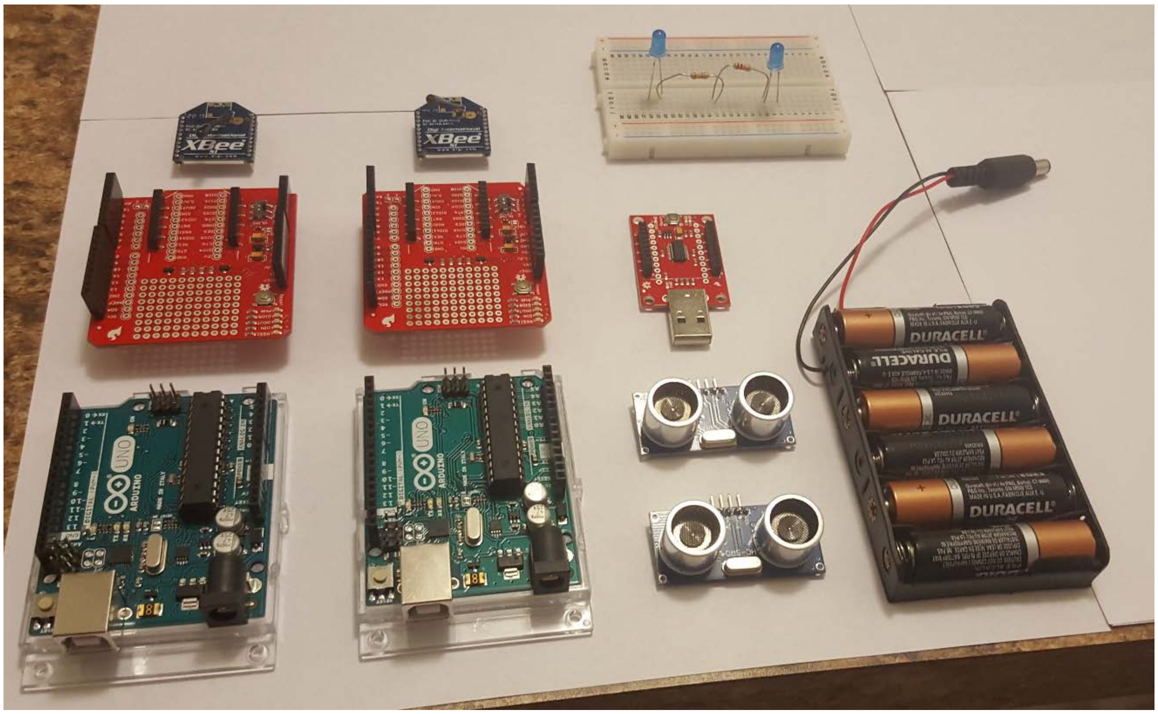


Figure 32 – Current Components Used to Setup a Test Bed

The following figures shows the shield, microcontroller, transceiver, and sensor, all attached together as one component using a breadboard. These components all together represent the single system required to collect the sensor’s data and send the data wireless across the network.

The following two figures represent the two terminals used to send a wireless transmission in a network composed of a single router and coordinator. The sending device’s terminal is shown on the left side of the screen while the terminal corresponding to the receiving device (the Arduino Uno), receiving the message containing that a sensor is active, is shown on the right side.

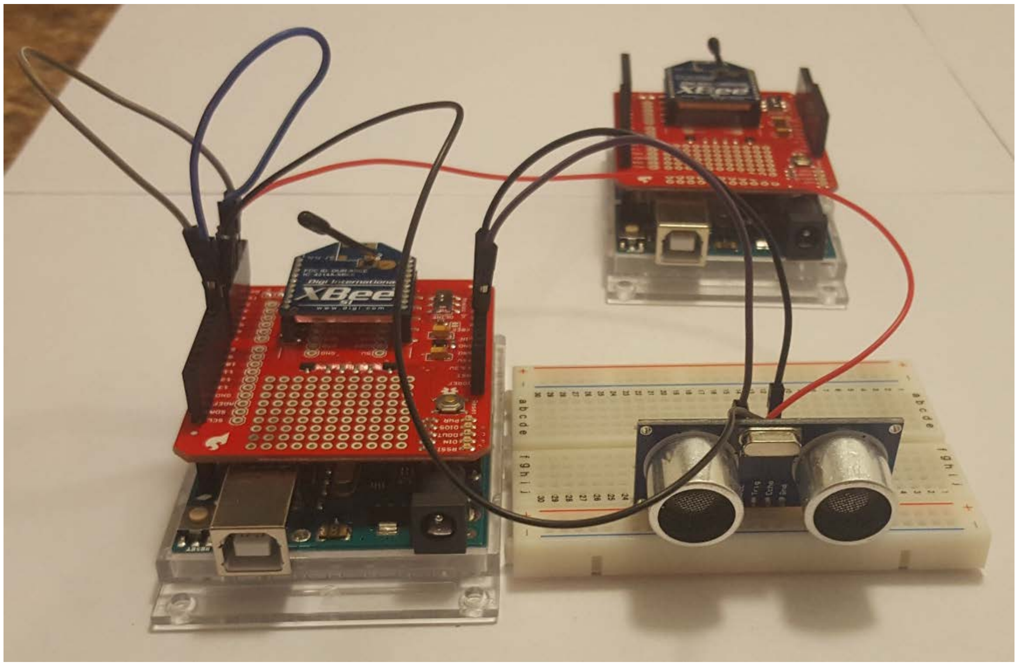


Figure 33 – Sending Device and Coordinator After Merging Components

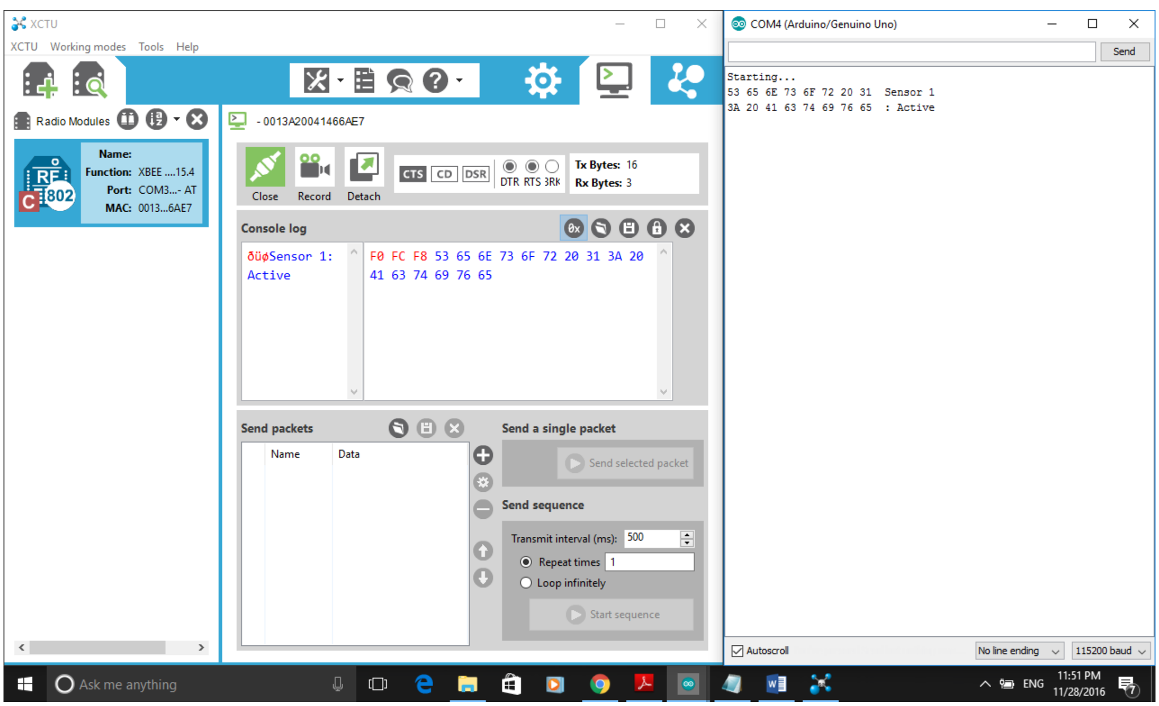


Figure 34 – Wireless Data Transmission in Between a Router and a Coordinator

5.1.1.3 System Integration & Schematic Profile

The biggest step to designing a fully operational Truck Smart system is integrating all components into a seamless design. For this system, there will be two schematics; one to represent the peripheral sensors and another to represent the receiver hub. Integration will require knowing the input/output layouts of each component, the power requirements for each component's operation, and which support components are necessary to integrate the ATMega328P chip into the system. Once this set of information has been gathered, each component's schematic traces will need to be analyzed and prepared to incorporate new devices. Any unused sections of the test-bed microcontroller will be removed to cut costs and make the task of PCB design less complex.

There are many tutorials available to guide through the "bread boarding process" of placing the microcontroller's components on a much more versatile workspace. Once the microcontroller has been converted to a modifiable platform, knowing which pins act as the general-purpose input-output (GPIO) ports is necessary before continuing to program the device. Figure 35 contains mapping to the actual test-bed (Arduino Uno) functionality. This diagram keeps the guesswork of pin finding out of the equation. The diagram will also be useful during the programming phase since each output pin will have a constant variable assigned to it to identify its function.

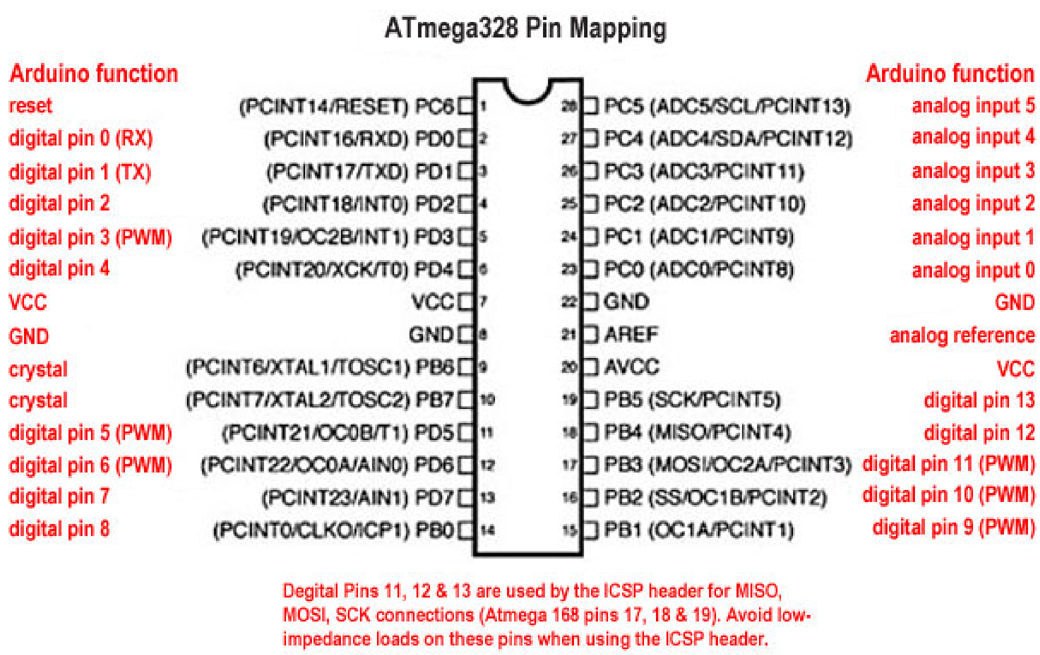


Figure 35 – Atmega328 Arduino Pin Out (used with permission from Atmel)

Now that the ATmega328 chip outputs have been mapped to their GPIOs, the next step is to map the XBee shield to its corresponding GPIOs and find which pins are being used by the transceiver. Tables 14 and 15 in the following sections reference the pin numbers located on the inside graphic of the chip in Figure 35.

The sensor schematic includes many capacitors between the inputs/outputs of supplied power and ground. These capacitors serve the purpose of “de-coupling” the power supply. Digital switching and power supply switching happen very quickly within the circuit. When switching occurs, rapid voltage and current changes can cause noticeable internal voltage spikes, more commonly known as noise. The de-coupling capacitors act as a filter to prevent the noise from saturating the circuit. The decoupling capacitors have also been placed across each terminal of the 16 MHz crystal to filter noise. The Signal-to-Noise Ratio (SNR) is a method to quantify how much noise exists within a circuit, compared to an existing signal. Normally, the SNR is expressed in terms of decibels. If the ratio is greater than one, the signal is more prevalent than noise. According to Meraki.com, the recommended SNR for data networks is 20 dB or higher, and 25 dB or higher for voice communication networks. The SNR can be a good measurement of the quality of transmission or internal data flow of a circuit. The addition of decoupling capacitors will increase the overall SNR of the Truck Smart sensor and, therefore, add to the reliability of the system.

The XBee transceiver requires only five of the twenty pins to operate for the Truck Smart application. The Tx and Rx pins are necessary for passing the transmitted and received data, to and from the ATmega328. The 3.3V Vcc and reset pins should both be high to provide the power which will be converted to electromagnetic waves. In the event the reset goes low, the XBee module will reset itself and an opportunity for data loss can arise during a transmit/receive operation. Table 14 (above) provides the ATmega328 pins for each component involved in the Truck Smart peripheral sensor.

The voltage regulators chosen for the circuit include the 7805T 5V regulator, and the LM 1117T 3.3V regulator. The 5V regulator is necessary for the ATmega328 chip Vcc. The 3.3V regulator is necessary for the XBee module. For the sensor, a 7.4V max input will be expected at the input of the 7805T. For the Truck Smart Hub, a 12V input is expected. The 7805T input voltage range peaks at 15V, which is not expected to be exceeded. The echo and trigger lines to tie into the ATmega328 have been chosen as digital pins 12 and 13 (see figure 35). Constant output variables will be used in a clock function to provide timing to the trigger. The echo will be set as an input pin and read the incoming return signal provided by the HC-SR04 sensor. An LED has been tied into the input line of the XBee module. This LED will confirm the XBee is receiving the necessary turn-on voltage required to operate. The LED has been tied to a voltage divider to provide a 2.1V input with 20mA across the diode. These values are the recommended inputs for green LEDs.

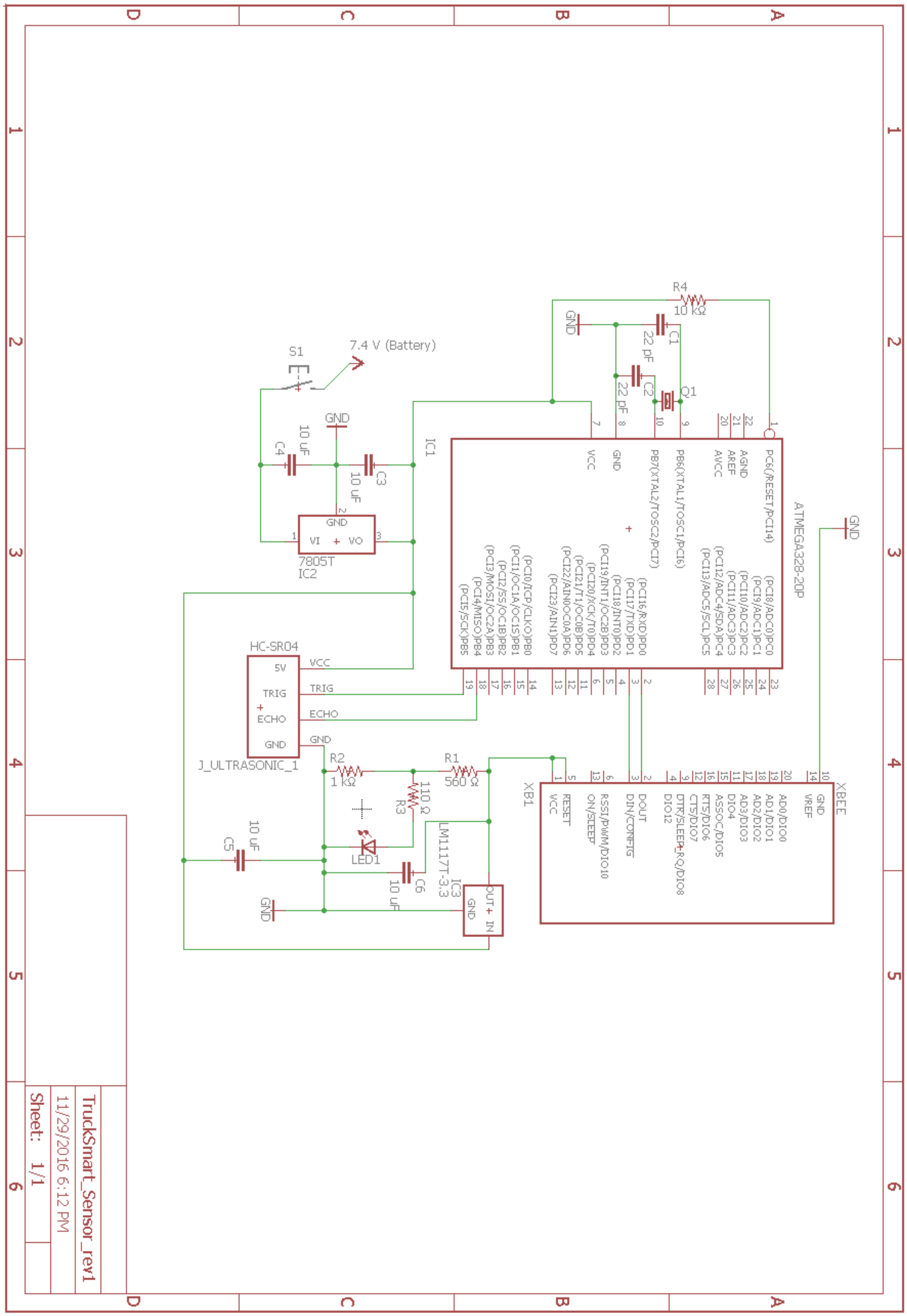


Figure 36 – Peripheral Sensor Schematic

Table 14 – Peripheral Sensor ATmega328 Pinout

Truck Smart Peripheral Sensor ATmega328 Outputs	
ATmega328 Pin #	Connection Description
1	Reset tied with resistor to Vcc
2	Rx XBee data receiving line
3	Tx XBee data transmit line
7	5V Vcc
8	Gnd
9	16 MHz crystal oscillator with de-couple junction
10	16 MHz crystal oscillator with de-couple junction
18	HC-SR04 Sensor echo line (Digital in)
19	HC-SR04 Sensor trigger line (Digital out)

Table 15 – Truck Smart Hub ATmega328 Pinout

Truck Smart Hub ATmega328 Outputs	
ATmega328 Pin #	Connection Description
1	Reset tied with resistor to Vcc
2	Rx XBee data receiving line
3	Tx XBee data transmit line
4	Sensor 1 LED output
5	Sensor 2 LED output
6	Sensor 3 LED output
7	5V Vcc
8	Gnd
9	16 MHz crystal oscillator with de-couple junction
10	16 MHz crystal oscillator with de-couple junction
11	Sensor 4 LED output
12	Sensor 5 LED output
13	Sensor 6 LED output
14	Sensor 1 LED fault line
15	Sensor 2 LED fault line
16	Sensor 3 LED fault line
17	Sensor 4 LED fault line
18	Sensor 5 LED fault line
19	Sensor 6 LED fault line

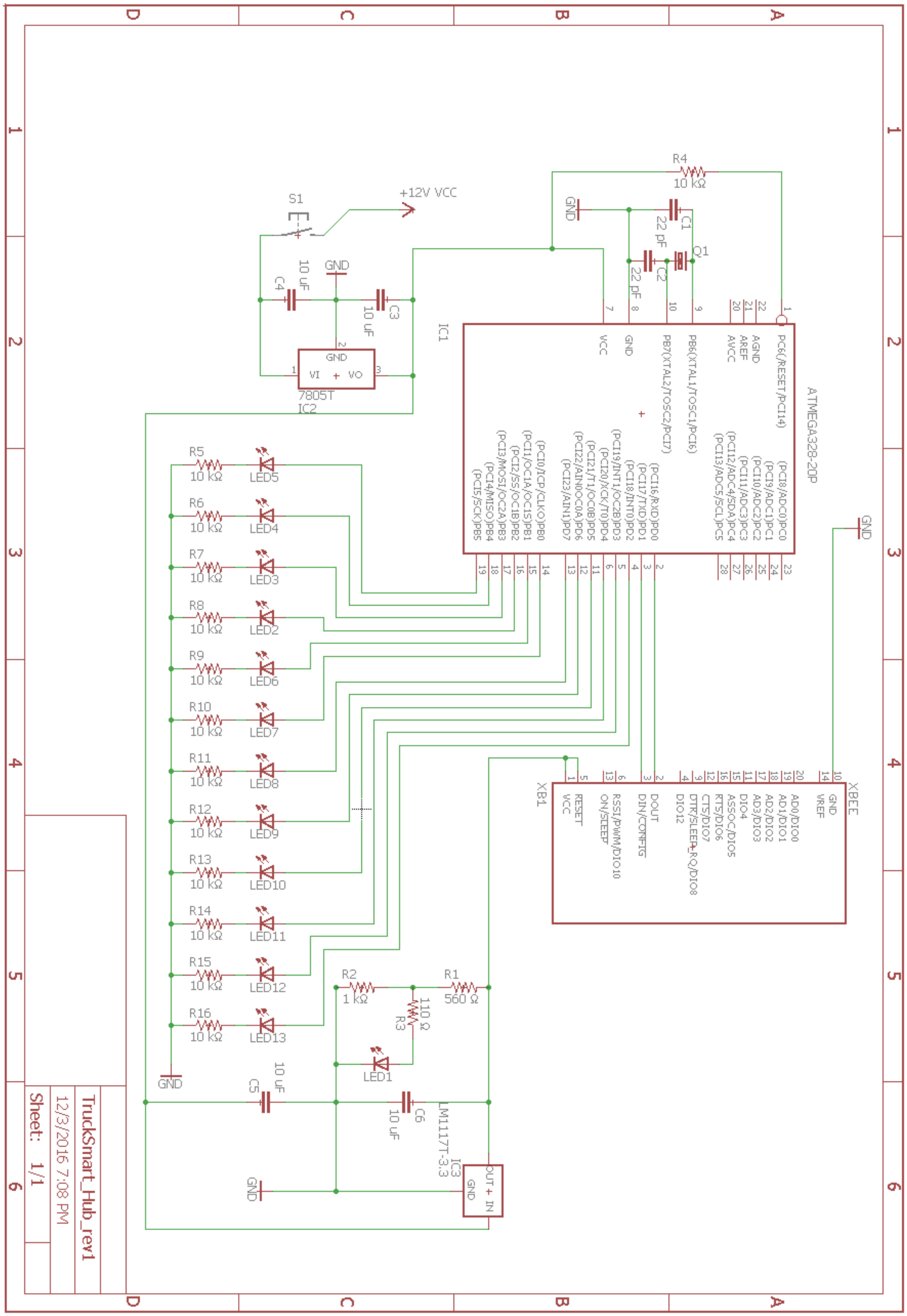


Figure 37 – Truck Smart Hub Schematic

The Truck Smart Hub schematic (shown in Figure 37) has a similar layout to the peripheral sensor schematic, due to using the same ATmega328. However, the major difference between the two is the LED scheme. There are 13 LEDs designed into the Hub to serve the purpose of providing obstacle warnings, error notifications, and overall system health. The design of the LED layout had to account for the output pin limitations of the ATmega328 and provide a logical manner to report errors and faults (using 6 red LEDs). To manage this, both sets of LEDs (notification and fault reporting) were given direct lines to the chip's digital output pins. Table15 provides the full scope of the ATmega328 outputs for the Truck Smart Hub.

5.1.2 Power Supply & Distribution Methodology

The Truck Smart power supply will be organized into three functional sections. These sections will be used to describe the concept of each piece of the system in a detailed, low-level manner while still maintaining an over-arching view of the entire infrastructure. Having a pre-designated layout for the process of establishing the technical details of the power supply, as well as any other major system segment, is a critical step in the integration of a system. Every electronic device, no matter the application, is dependent on a power supply of some type. A flawed design in a power supply, over-powering or under-powering a component, and lack of heat dissipation within a system are a few pitfalls, of many, that can prove detrimental to a system. The three functional sections to be discussed will address the design aspects and standards, power management and component distribution requirements, as well as system failure provisions and heat dissipation of the Truck Smart system.

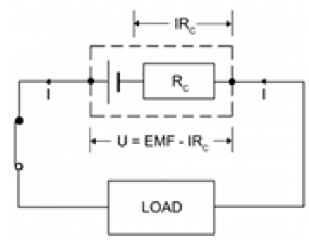
5.1.2.1 Power Supply Design Aspects

The functionality of a power supply comes from its capability to provide a constant output voltage (DC) to the circuit. This output voltage should be independent of both the system's load resistance, and the power supply's input voltage. The potential difference in output/input voltages is dissipated mainly as heat. In order to prevent damage to the circuit and powered components, the design should not exceed the specified operational input voltage range of the component(s).

5.1.2.1.1 Voltage Regulators

In terms of power supply, the main component of concern for the Truck Smart system is the **voltage regulator**. According to "Linear and Switching Voltage Regulator Fundamentals" published through Texas Instruments, the three basic types of linear voltage regulator designs include the Standard NPN Darlington Regulator, the Low Dropout (LDO) regulator, and the Quasi LDO regulator. There are many other different types of voltage regulators, both linear and non-linear, that are intended for applications requiring AC/DC power conversion and DC regulation. Since the Truck Smart system voltage inputs will only concern DC power, via batteries for the peripheral sensors and 12V DC input from the truck cabin cigarette lighter port, the recommended voltage regulator type for this application (as provided by the Texas Instruments published document specified above) is the LDO regulator.

The LDO regulator is the preferred component for the application based on the parameters of maximum operational load current, input voltage classification (DC), voltage input/output precision, idle current, and included circuit features intended to provide easy integration into the Truck Smart system. The system inputs are only of the DC type, therefore there is no need for wave rectification to be included in the voltage regulation module (as there would be for systems requiring AC inputs), and the LDO regulator meets this criterion. "Linear and Switching Voltage Regulator Fundamentals" states that, "In battery-powered applications, LDO regulators are usually the best choice because they utilize the available input voltage more fully (and can operate longer into the discharge cycle of the battery)". When using a battery of any type, the terminal voltage decreases over the course of the battery's discharge. This change in terminal voltage is caused by the growing difference between the Electro Motive Force (EMF) and the battery's internal resistance (relative to the circuit's current flow). Figure 1 (provided by www.itacanet.org) encompasses the circuit-battery relationship as it relates to the battery's terminal voltage.



$$U = E_{EMF} - IR_C$$

Figure 38 – Terminal Voltage Conceptualization

Since this terminal voltage is a variable prone to change, measures must be taken to harness as much potential energy over the course of the battery's lifetime as possible. Terminal voltage drop and terminal voltage tolerance are two influential factors to be taken into consideration when deciding on which battery to integrate into the Truck Smart system.

5.1.2.1.2 System Components Overview and Battery Selection

There are a number of battery options available on today's market. For the Truck Smart system, the preferred design should allow a fully active system to operate for at least 18 hours. To calculate the available online duration of a system before a complete battery discharge, the consumption characteristics of each active component should be known.

Table 16 – Peripheral Sensor Power Consumption

Truck Smart Peripheral Sensor	Operating Voltage (V)	Operating (Idle) Current (mA)	Active Current (mA)	Current Draw (mA) @ 0% Idle	Power (mW)
HC-SR04 Range Sensor	5	15	15	15	75
XBee RF Module	3.3	50	45	50	165
ATmega328P Chip @ 16Mhz *	3.3	12.3	21.8	21.8	71.94
Total Load Draw	3.593778802	77.3	81.8	86.8	311.94
Regulator Efficiency (est.)*		75%		*estimations based on ATmega328P chip test data provided by www.gadgetmakersblog.com. Chip can be run 3.3 V @ 8Mhz w/ active current estimate of 12.3 mA.	
Battery Capacity (mAh)		1000			
Lifetime est. (hours)		8.640552995			

The components for a single battery-powered peripheral Truck Smart sensor include the HC-SR04 Ultrasonic ranging module, the XBee RF module, and an ATmega328P Chip. Table 1 below shows the estimated calculation for a 1000 mAh battery (equivalent to capacity of one AAA battery). Using these available estimates, the independent variable of the equation is the battery capacity. To achieve the preferred system operational period of 18 hours, the required battery capacity is estimated to be 2084 mAh. For the purpose of practicality and a provisional power consumption buffer, the sensor battery to be integrated will be a Kingtoys© 7.4V 2500mAh Lithium Polymer battery pack. This battery is estimated to provide a 21.6-hour operational lifespan before needing recharged.

5.1.2.2 Power Management and Distribution

A stable circuit design to manage and distribute power to key components of the Truck Smart system is, undeniably, a critical infrastructure. For the peripheral sensors, a linear voltage regulator will be placed in the input line of the circuit. Therefore, the first stage of the circuit will act as a “voltage step-down” from the 7.4V down to 5V. There will be added resistors to limit current to operational specifications provided by each component’s data sheet. During the biasing of these resistors, standards (discussed in the previous section) will be taken into consideration to ensure ELV and limited current characteristics are met. The 5V line can then be separated into two paths; Path 1 can provide input voltage to the microcontroller variant. The XBee module will be dependent on this microcontroller. Both the XBee module and the ATmega328P Chip require 3.3V input. Another voltage step-down regulator is included in the microcontroller's circuit path; therefore, any further design on Path 1 should deal with ATmega328P Chip integration. Path 2 should provide a direct 5V feed directly into the HC-SR04 Range Sensor V_{CC} terminal. The sensor also requires a digital trigger clock signal and an echo feedback signal. These signals will be provided, processed, and translated through the microcontroller. Once translated, a predefined logic scheme will then prompt a transmission via the XBee module.

The Truck Smart Hub will act as a "sync and receive" device. The power line to the hub is provided by the cigarette lighter port, which will provide a 12V output. A voltage step-down LDO regulator will be placed in the input path, prior to providing input to the Arduino Uno Microcontroller variant.

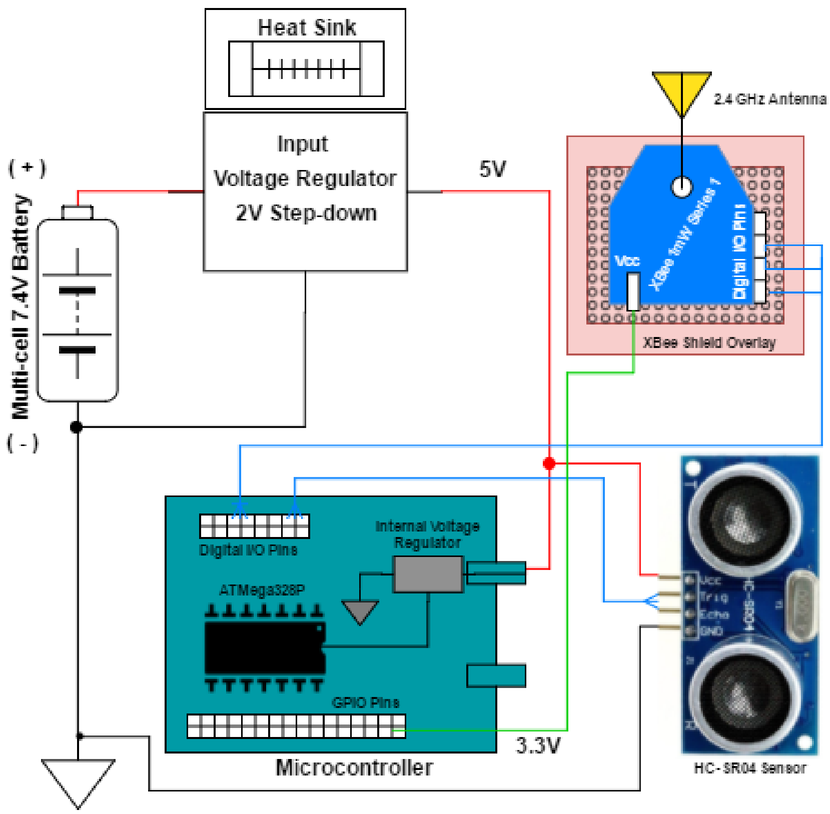


Figure 39 – Peripheral Sensor Power Schematic

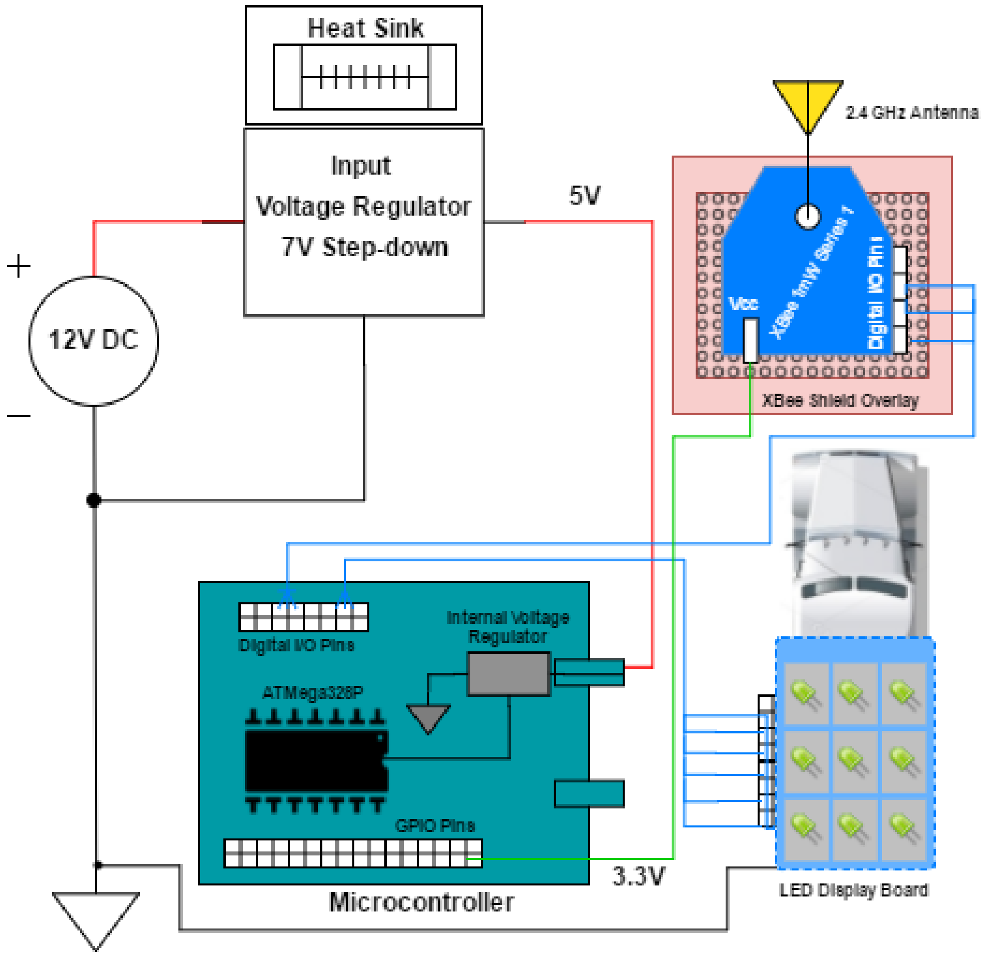


Figure 40 – Hub Power Schematic

5.1.2.3 System Failure Provisions and Heat Dissipation

From the estimations made in Table 1, the Truck Smart system power draw is anticipated to be in the realm of 300 mW. Over the course of time, this power will begin to heat the critical components of the system. To add to the engineering challenge, the Truck Smart system is required to be weatherproofed to withstand the elements of nature. The element of immediate concern would be the sun. The design of the sensor peripheral housing should take into consideration the heating effects of the sun, as well as the power-heat conversion factor. Each component has a designated temperature operating range, as shown in Table 11. In order to keep the system within the designed temperature limits, there must be a method of cooling to prevent damage to the components. Some design options for meeting this challenge include housing ventilation and heat sinks. Ventilation is a method requiring minimal system modification. In theory, the constant laminar airflow across the body of the trailer should provide an unmatched source for cooling potential. In the event that this method is not 100% effective, heat sinks will be included with the voltage regulator of each sensor to prevent the battery from overheating the encapsulated system.

Table 17 – Operational Temperature Ranges and Cooling Methods

Truck Smart Peripheral Sensor	Low Operating Temperature (oC)	High Operating Temperature (oC)	Preferred Cooling Method
HC-SR04 Range Sensor *	-40	70	Air-cooling and shading provided by protective housing design.
XBee RF Module	-40	85	Static dissipation
Voltage Regulator	-40	125	Heat sink
ATmega328P Chip @ 16Mhz	-40	85	Heat sink and air-cooling

There are system failures, other than overheating, that still need to be accounted for. The most critical of these events (in terms of peripheral sensors) is complete sensor failure, a circumstance requiring major system monitoring and “confirm/deny” techniques pre-designed into the Truck Smart software for the specific scenario. Once a sensor is considered faulty, the connection should be monitored by the hub to continue checking for functionality. The hub design should also be able to notify the user about the system fault and prevent said user from having a dependence on a false reading. The Truck Smart hub will notify users about system faults through a specific LED combination, or possibly a different color LED (Red) than the normal operational colors. Once the user is notified, the hub software may enter a “receive-loop” to verify the, once transmitting signal, is in fact a permanent loss. Only a short duration of the “receive-loop” should be necessary for confirmation.

5.1.3 Supplement Ways to Power the Sensor/Arduino

In this system, 6 sensors will be placed on the outer body of a truck that will be wirelessly connected to the display in cabin. All 6 sensors will have their individual housing and will be powered by rechargeable batteries. It’s desired that the system runs as long as possible between battery changes. Along with using energy efficient microcontroller and sensors, the run time can be increased even more by using solar panels to charge the batteries. Housing with solar panels for increased run time will be an optional package to this system and will not be included in the base model. This section details the specs and constrains of using solar power.

5.1.3.1 Power Calculations

Power generating from solar panel is constantly fluctuating depending on factors like the weather condition, angle that light hits the panel, constantly changing direction of the vehicle, etc. For these reasons, it's impractical to power the components directly using solar power. Instead, solar power should be used to charge a battery pack that supplies the sensor and other components. Power calculations for how long will it take to recharge a 7.4V battery to full capacity are below.

$$\text{Charge Time (Hours)} = \frac{\text{Amp−hours removed x 1.15}}{\text{charge rate}}$$

Equation 5 – Battery Charge Time

$$\text{Where Charge rate} = \frac{\text{Watts generated by the panel}}{\text{Battery Voltage}} = \frac{1\,Watt}{7.2\,Volts} = 0.1388 \text{ Amps}$$

Equation 6 – Battery Charge Rate

$$\text{Charge Time} = \frac{0.624\text{ Ah}*}{0.1388\text{ A}} = 4.5 \text{ Hours} * 312 \text{ mW} * 10 \text{ Hrs} = 3.12\text{Wh} = 0.624 \text{ Ah}$$

Equation 7 – Expected Charge Time

5.1.3.2 Conclusion

Using solar power isn't necessary for this system but it increases the interval between battery changes drastically. Based on calculations in the section above, adding a 1-watt solar panel (6"*6") can eliminate the need to change batteries for sensor peripheral as long as the truck is in solar light for at least 4.5 hours a day. However, adding solar panels increases the cost of the system and therefore will be offered as an optional package only. The base package will require battery change after 8 hours of run time.

5.1.4 Sensor Implementation

An HC-SR04 sensor is used with Arduino Uno for this system. The code returns the distance of the closest object in range; if this distance is less than the threshold distance, it means that there is a vehicle in the blind spot. To do this, the microcontroller sends a pulse to the sensor to initiate a reading, and then listens for a pulse to return. The length of the returning pulse is proportional to the distance of the object from the sensor.

The following schematic shows the electrical design of the HC-SR04 – Arduino system.

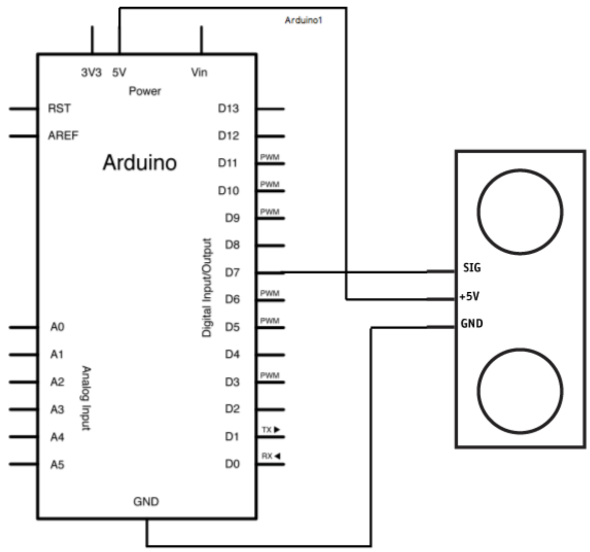


Figure 41 – HC-SR04 – Arduino System Schematic

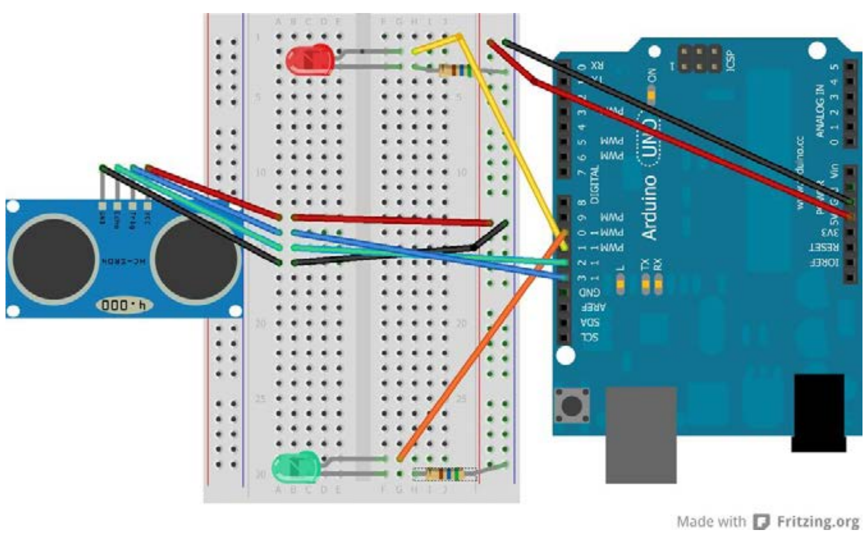


Figure 42 – HC-SR04 – Arduino System Physical Connections

5.1.5 Hardware Components and Implementation

The XBee module runs at 3.3V which means that their input/output lines will operate at this voltage. If the designer were to directly connect and XBee module to an Arduino microcontroller, it would mostly damage it since the Arduino operates at 5V. The XBee shield performs a level conversion which creates a mechanism that allows for the operation of both devices simultaneously. In order to connect an XBee module to an Arduino microcontroller, the DIN (data in) and DOUT (data out) pins of the wireless module must be connected to a serial port in the Arduino. The Arduino device in use contains two types of serial ports. First, the hardware serial port known as UART (Universal Asynchronous Receiver and Transmitter) is a piece of hardware inside the microcontroller that can operate a line by itself. It will only need to be accessed by the microcontroller if data was received or sent. Secondly, the software serial port uses normal digital input/output pins for each bit separately.

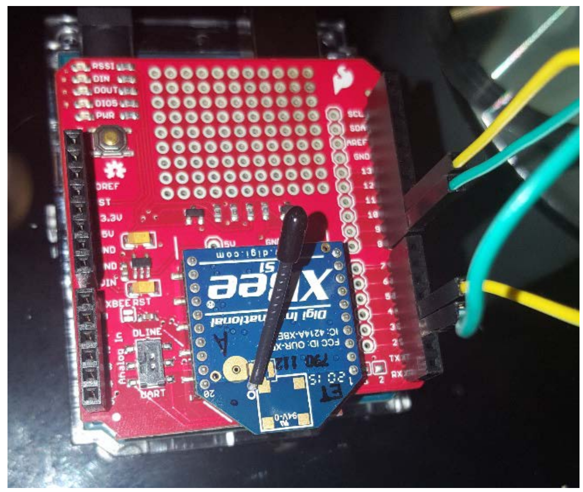


Figure 43 – Connecting all Components into a Single System

The microcontroller will have to monitor and control the states of the pins separately which can make any embedded software application that uses the serial port not work properly. The software serial port requires a lot more processing time and is more sensitive to timing problems when other codes and interrupts are involved. As a result, for the Truck Smart system, the XBee module and the microcontroller will communicate with each other using the UART serial interface. In order to allow for communication between the XBee module, the shield, and the microcontroller, the switch in the shield will be set to the “DLINE” position. This will allow for connecting the Arduino RX (receiving pin) to the XBee DOUT (data out) and the Arduino TX (transmitting pin) to the XBee DIN (data in). This connection will allow for communication in between the XBee and the shield. The use of jumper wires will be required to connect pins 2 to 8 and 3 to 9 which belong to the Arduino AltSoftSerial interface. These set of connections will allow for communication in between the shield and the microcontroller. A final layout after all connections have been made is shown in the following figure.

On the figure, the XBee S1 module can be observed attached to a SparkFun XBee shield, while its all connected to an Arduino Uno microcontroller which cannot be seen in the picture. The Uno is located right under the shield. The serial switch can be seen to be set at the DLINE position while the green cable jumps pin 2 to 8 and pins 3 to 9.

5.1.6 User Interface Design

The Hub's display concept (shown in Figures X and Y) is designed to relay only necessary information to the driver. Providing too much information through the interface can act as a distraction to the driver, instead of acting as an aid. The Truck Smart system will benefit the driver most if it is able to interweave into the driver's situational awareness without causing him or her to look away from the road. From the design perspective, this is accomplished by using subtle methods of notification. For instance, softening the light intensity from the LEDs through the use of an opaque cover plate prevents bright focal points (LEDs) from overwhelming the driver's peripheral sense. Sudden and random changes in an environment can cause the human brain to unconsciously deviate its attention to the focal point of variation. When a sensor LED is activated, instead of abruptly turning on or flashing, the LED should slowly brighten to a full luminescence and then gradually fade back to an off-state once the sensor confirms an object no longer exists in the blind spot. To achieve a gradual, brighten/fade effect, capacitors can be placed in line with the LEDs. Capacitors must charge and discharge before reaching their voltage peaks and lows, respectively. Adding capacitors to the system can also have the unintended benefit of lengthening the operational lifespan of the LEDs. Diodes are susceptible to burning out from continuous voltage spikes over time. Incorporating capacitors into an LED's immediate circuitry can attenuate these spikes and extend the component's life expectancy, therefore making the overall system more reliable.

5.1.7 Human-Machine Interface & Psychology

The study of Human-Machine Interfacing (HMI) covers a wide realm of psychological characteristics that allow for humans to easily adapt to a system's pre-defined functions. Some cases may be trivial, such as the positioning of a door handle in a design to allow the user to use maximum potential leverage from their arm when opening the door. Other scenarios may be more complex and can require studies to be done on error vulnerability and user efficiency. For the Truck Smart system, the study of HMI is ever-present in the Hub's display. To reduce the driver's cognitive load, the presentation of information should be easily connected to prior experience with a similar system. The "similar system" does not have to be comparable to the Truck Smart system's functionality. Only the information display and unconscious assumptions must be connected.

Shapes and colors are very useful when merging well-known conventions. For the Truck Smart system, the blind-spot obstacle notification color will be deemed yellow. Yellow signifies a warning or potential hazard in the user's mind and is a color that is used for countless notifications of similar status in everyday life. The yellow LED will illuminate in a location on the Hub that pertains to the peripheral sensor. The display will be positioned to imitate the length of a truck's trailer, with LEDs representing every one-third length of the trailer for each side. The color red will symbolize a sensor error, a system fault, or a dead sensor battery. Red is a well-known indicator for faults and is accepted across multiple industries as an error signal. Green will signify a healthy, operational system that has been successfully synced. Table 16 lays out all possible signal colors, notification flashing patterns, their associated meanings, and methods to resolve any system issues or errors.

Table 18 – Signal Identifiers

Signal Identifier	**Meaning**	**Error Code (Y/N)**	**Resolution**
Red	Sensor error or system fault. Error should not be left unattended to.	Y	Reset and re-sync system
Flashing Red	Dead sensor battery for location specified on display.	Y	Charge sensor battery
Yellow	Warning for obstruction in blind spot. Driver should take caution.	N	-
Flashing Yellow	Low battery warning. Flashes intermittently every 2 minutes.	Y	Should not rely on sensor until charged
Green	Healthy system with successful sync. Operational and trustworthy.	N	-
Flashing Green	Currently attempting sync with peripheral sensors, similar to loading.	N	Wait until system is completely synced.

A sample of what the driver will see at any given time is shown below in Figure 41. For this instance, sensors 1, 3, and 5 are displaying that there is an obstruction being sensed in the general area. Sensor 4 is shown intermittently flashing red, meaning the battery in sensor 4 is dead, needs recharged, and re-synced. Sensors 2 and 6 are not illuminated on the hub display panel, signifying that there are no obstructions sensed in the blind spot locations.

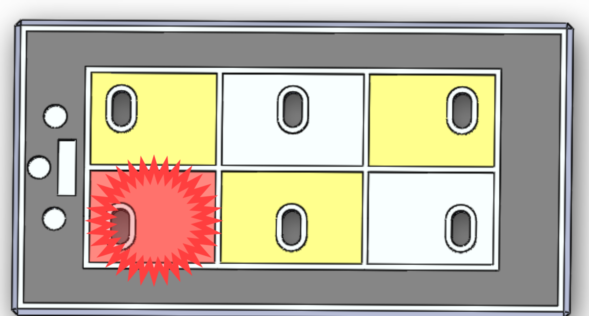


Figure 44 – System Output Interface

5.2 Software Design

The microcontroller is essentially a small computer that controls the various devices that are connected to it. The Truck Smart system will be using eight microcontrollers. These microcontrollers will provide various functions to the system which will vary by which unit they are part of, the sensor versus the hub unit. The Truck Smart system will contain seven sensor units and one hub unit.

5.2.1 Design Methodology

While developing the necessary code for the Truck Smart system, the team will be taking advantage of the agile software development paradigm. The agile methodology puts a focus on adaptive planning and development. As the product is created, requirements or constraints may change as new information is gathered. This is the Truck Smart team's first time working with most of the technologies used to create the product. It is entirely possible that during the process of development, the team discovers that a particular component is not actually compatible with the system and needs to be replaced or that a particular software library is not robust enough to support the application. The agile movement encourages its users to respond and readjust their development tactics if any hurdles, such as the previously mentioned, appear. This is achieved through a system of early delivery and continuous improvement.

The traditional method of development involves a linear series of planning, designing, implementing, testing, and release. On the other hand, agile development follows a circular schedule where the testing phase leads back toward the planning phase and you repeat until you are ready to release. This type of schedule will allow the team to constantly accommodate for any new changes or knowledge.

More specifically, the team will be using the scrum framework. Under the scrum system, the development of the Truck Smart system will be split into multiple sprints, each lasting one to two weeks depending on the complexity of the primary task. The team will hold a meeting, led by the scrum master, at the end of each sprint to confirm that progress is being made and to discuss any necessary modifications. The role of the scrum master will fall to the leader of the primary task; Aris Socorro will lead sprints focusing on wireless communication while Abhijith Santhoshkumar heads sprints targeting microcontroller programming.

5.2.2 Development Tools

When writing software, it is important to take advantage of every tool available. This will result in a shorter, simpler, and easier development process. The tools used to develop the Truck Smart system can be seen in the following sections, this includes the IDE and the VCS.

5.2.2.1 Integrated Development Environment

An IDE is an application that contains a code editor, code builder, and debugger. It provides developers with an all-in-one tool to use while creating products. Since the Truck Smart system will be prototyped using the Arduino Uno board and creating a final product using the Atmel ATmega328P microcontroller, the IDE of choice is the Arduino Software IDE, version 1.6.12.

Arduino Software comes with built in support for Arduino products. All that is required are simple plug and play drivers and you can immediately start programming the Arduino board. Most importantly, it also comes with tools to burn the Arduino bootloader onto other microcontrollers. This function is crucial as it will make the code from the Truck Smart initial prototype made using the Arduino Uno compatible with the hardware even after the Uno is switched out for a final PCB and custom microcontroller combination.

Arduino projects are called sketches and are written in the Arduino language, based on C/C++. This is convenient as every team member has prior experience with these languages. The Arduino IDE also comes with pre-existing code examples are fully functional with the Uno board. This is highly meaningful as the team has no prior experience with any Arduino products and will provide a good starting location. Through these examples, it was found that in addition to the main functions every program tends to have, Arduino sketches also include a setup function that runs once at every reset or start-up.

This IDE is open source and extensible. This means that the community is free to create add-ons for the IDE which could potentially drastically ease the development process. The Arduino environment also supports a vast array of libraries including support for ZigBee communication, something that is essential to the Truck Smart system. Furthermore, it is a cross platform application with support for Windows, Mac, and Linux. So unlike with another IDE such as Visual Studio, the team will not have to worry about operating system compatibility issues.

To load code into the Arduino microcontrollers, the Arduino Integrated Developing Environment (IDE) will be required. The choice of using the Arduino devices for the Truck Smart system is due to the many supported libraries for interaction with ZigBee modules. The developing team will be using the latest version “Arduino 1.6.11”. This will allow for maximum support and usability of the built-in features in the Arduino Uno microcontroller while interacting with the XBee Pro S2C wireless transceiver.

5.2.2.2 Version Control System

According to Git’s Getting Started guide, “version control is a system that records changes to a file or set of files over time so that it is possible to recall specific versions later." This will allow the team to have a record of development while also keeping backups in case something goes wrong and it becomes necessary to roll back to another previous version, even if that version was last accessed weeks or months ago.

In addition to its bookkeeping and redundancy features, the use of the version control system will also make the team's collaboration on code a lot easier. Distributed Version Control Systems house all files that are relevant to the project in something called a repository. This repository is stored remotely in the cloud. Users can then clone the repository to their local computer. This provides an additional layer of backup for the files since the repository will exist not only on the remote server, but also locally on each user's computer. Once cloned, users can work on any files at their leisure, including working on the same file simultaneously. Users will then commit and push their changes to the server. If two users push changes to the same file, it will automatically merge them and warn you of any potential problems. Once the changes have been successfully pushed, other users can pull the latest version and continue to work on the most recently updated files. This system will enable every to see exactly who contributed any given line of code along with the time and date, making documentation hassle-free.

When choosing a version control system to use, the team's initial thought was GitHub, the most eminent in its field. However, it was quickly dismissed upon learning that any projects published to GitHub would be public. There is an option to add private repositories but it comes at a monetary cost which would become an unnecessary expense. Bitbucket, while slightly less notable than GitHub, is another option for a version control system. While researching the team's options, it was found that Bitbucket actually provides all the features of GitHub at no cost for teams of up to five, including private repositories. For this reason, Bitbucket was chosen to be the team's version control system.

5.2.2.3 XTCU

There are 2 main software applications that will be required to implement the wireless communication in the Truck Smart system. The Arduino IDE was the first. The second one is the XCTU application developed by Digi International. According to the manufacturer, XCTU is an application "designed to enable developers to interact with Digi RF modules through a simple-to-use graphical interface. It includes new tools that make it easy to set-up, configure and test XBee® RF modules". In other words, XCTU is a must have application for any developer that wants to set up a ZigBee network using the XBee modules or transceivers. The application enables the developer to configure the devices to talk to each other in a network. Some advantages and functionality include the ability to assign network IDs, wireless channels, enable/disable channel verification, update firmware version, modify devices to work as coordinators, routers, or end devices, and many other functionality applications. The developing team will be using the latest version 6.3.2 released back in August 2016.

The latest version of XCTU contains unique features such as the ability of displaying a graphical network view in which the developer can visualize the network along with the signal strength for each connection. Another important feature is the ability to manage and configure devices over the air. Once the network has been set up, and the devices have been added, if one of the devices need to be modified in any way, there is no need to plug it directly into the computer. The only requirement is for it to be powered and connected to the network and the programmer can simply go in the XCTU application and make any required modifications to it from there. Another very efficient feature is the ability to perform range tests from within the software application. This is a very useful feature for the Truck Smart system since it will allow the developing team to test how effective the network will perform before actually attaching the system to a truck, which can take a significant amount of time. The developers will have the ability to set the devices at some distance from each other and using the software, can see the maximum range supported by the network.

5.2.3 Sensor Unit

The sensor units will be placed around the truck and will all transmit its data to the hub unit using the node bouncing technique explained in the networking technology section. The functions of the sensor units that the microcontroller must be programmed to perform are as follows:

- Join the ZigBee Node network.
- Supervise communication between the ZigBee nodes.
- Operate the ultrasonic sensor and receive the corresponding output.
- Send the sensor's output data to the hub unit using the node bouncing method.
- Pass along any data received from another node towards the hub unit.
- Detect any errors or hardware malfunctions and relay to the hub unit.

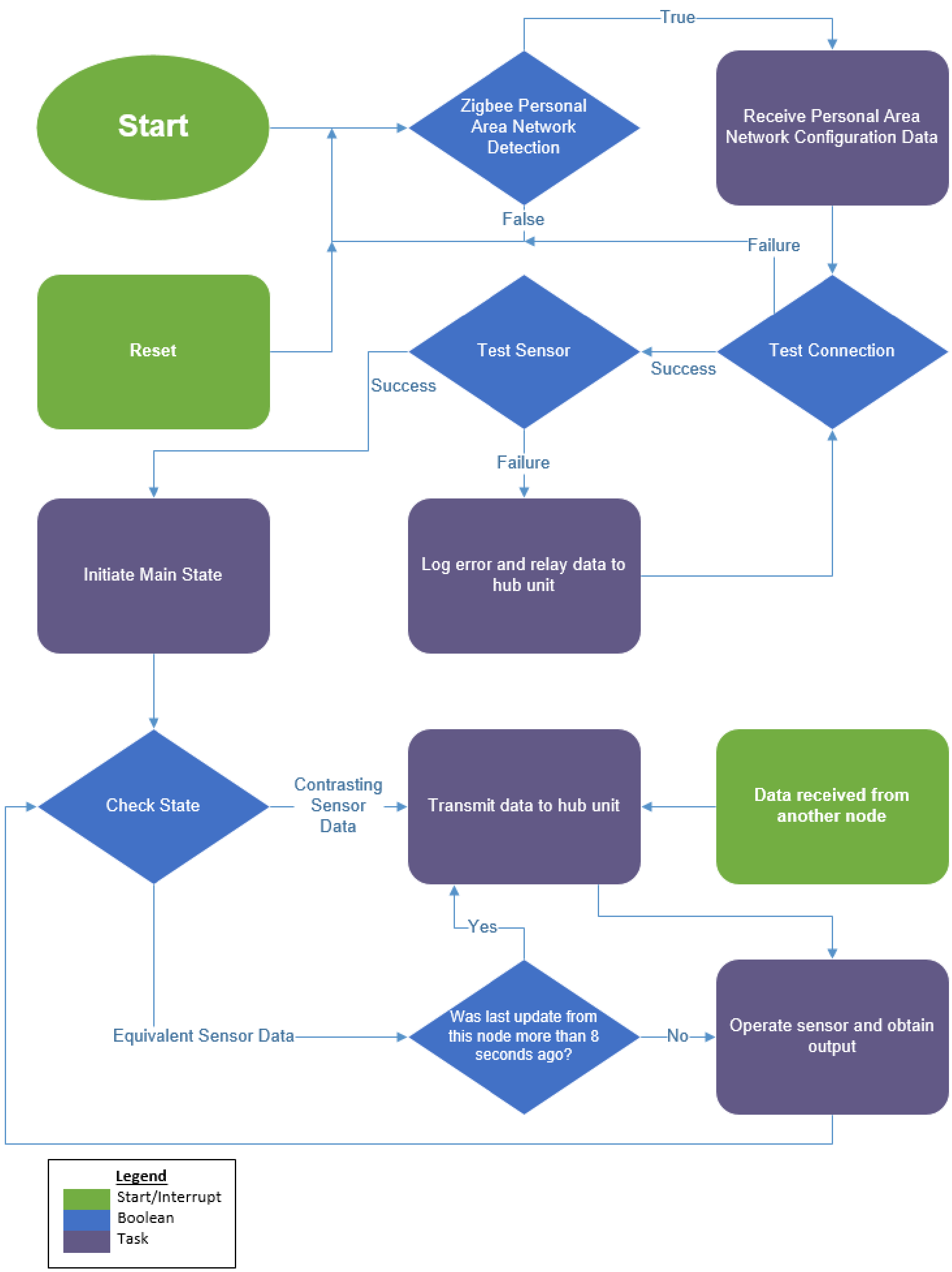


Figure 45 – Software Flowchart: Sensor Unit

The logic for the software in the sensor unit can be seen in the software flowchart in Figure 45, this will be the topic of the rest of this section. In that chart you can see that when the sensor unit is turned on its first priority is to detect a ZigBee

Personal Area Network (PAN). Joining a PAN is the utmost priority as that is how the XBee units communicate with each other. An XBee unit cannot talk to other units that are not in the same PAN. This is crucial as this is what prevents the major issue of interference if two trucks using the Truck Smart system were to drive near each other.

Once a PAN is discovered, the unit will then attempt to join it. The hub unit will send it network configuration data such as that sensor unit's unique identifier token. This token will separate this sensor from the other sensor so that the hub can differentiate between them. From here it will test the connection by sending and receiving multiple packets of sample data and checking that it was received and transmitted back by the hub in an accurate and timely manner. If there is an error in the connection, the unit will leave the network, rediscover the network, and repeat the connection test. It will repeat this cycle until it is either successful or turned off. If this check is, it will then move on to the sensor testing.

The sensor assessment consists of a simple connection test between the microcontroller, sensor, and XBee module. The microcontroller will operate the sensor, receive the sensor's output, and transmit it to the hub unit using the XBee module. This evaluation only checks if the sensor is communicating properly with the other devices. It is up to the user to make sure the sensor is detecting properly and not showing any errors such as false positives. This is because there is no way for the system to check if the sensor data is actually accurate. If the system is unable to properly communicate with the sensor, it will log an error and alert the hub unit which will notify the user. From here, it will enter a loop of testing the connection and then testing the sensor until both assessments are positive. Once the sensor test is passed, the system will enter its main phase.

As seen in Figure 45, the sensor unit's main stage consists of three simple steps and one interrupt. The sensor will continuously be operated and analyzed in real time. If the sensor's output has changed, it will update the information and pass it along to the hub unit. Otherwise, if the output has not changed, it will check when that last update to the hub unit was. If it was more than 8 seconds ago, the sensor will resend the data to the hub. This function is to let the hub know that the sensor is still working properly and has not just failed to respond. Once this is done, the sensor will be checked again and the whole process is repeated.

At any point during this operation, the unit could potentially receive data from another unit. This is part of the node data bouncing method and this action will process as an interrupt. This unit's priority will be to pass along the data to the next node in the network, until the data reaches the hub unit. Once this data is passed along, this sensor unit will go back to its regularly scheduled programming.

5.2.4 Hub Unit

The hub unit will be displayed in the cabin of the vehicle in a spot easily accessible by the driver. It acts as the liaison between the sensors and the user. The functions of the hub unit that the microcontroller must be programmed to perform are as follows:

- Create the ZigBee Personal Area Network
- Manage the ZigBee PAN by:
    - Detecting sensor units
    - Adding sensor units to network after configuring its properties
    - Keeping track of the status of each sensor unit
    - Try to fix malfunctioning sensor units through software resets
- Analyze output data of sensors received via ZigBee communication
- Display results of sensor data to user through the use of LEDs
- Warn user through LEDs of the following:
    - System errors
    - System malfunctions
    - Sensor unit low battery

As with the sensor unit, the logic flow of the hub unit is also split into two main stages and can be seen in Figure 46. Each step of the flowchart in Figure 46 will be explained in order. When the device is turned on, it will begin in the setup stage. In this phase, the first step is to setup the ZigBee PAN. Once the PAN is configured, the hub unit will scan for any sensors that are asking to join the network. It will verify the sensor units, give them each unique identification tags so that each sensor can be differentiated between, and add them to the network.

As soon as the hub adds a sensor unit to its system it will immediately run diagnostic test to make sure the connection is working properly. This is accomplished by sending a simple string back and forth, making sure the data is sent and received properly by both units. If a connection error is found, then the hub will remove the sensor unit from the network and warn the user. It will then try to re-add the sensor unit to the network until it succeeds the connection test or the user interrupts it. Once it is established that there are no connection errors, the hub will repeat the same test but this time with the sensor output data. If the sensor test fails, it will alert the user and start a diagnostic loop from the connection test to the sensor test until both are successfully cleared or the user interrupts it.

From this point on, the hub unit will be in the main phase. This phase basically contains an infinite loop consisting on continually checking the sensor output data and displaying it to the user. For readability and simplification purposes, the data analysis stage and troubleshooting stage in Figure 46 are displayed separately in Figure 47 (Data Analysis) and Figure 48 (Hub Troubleshooting).

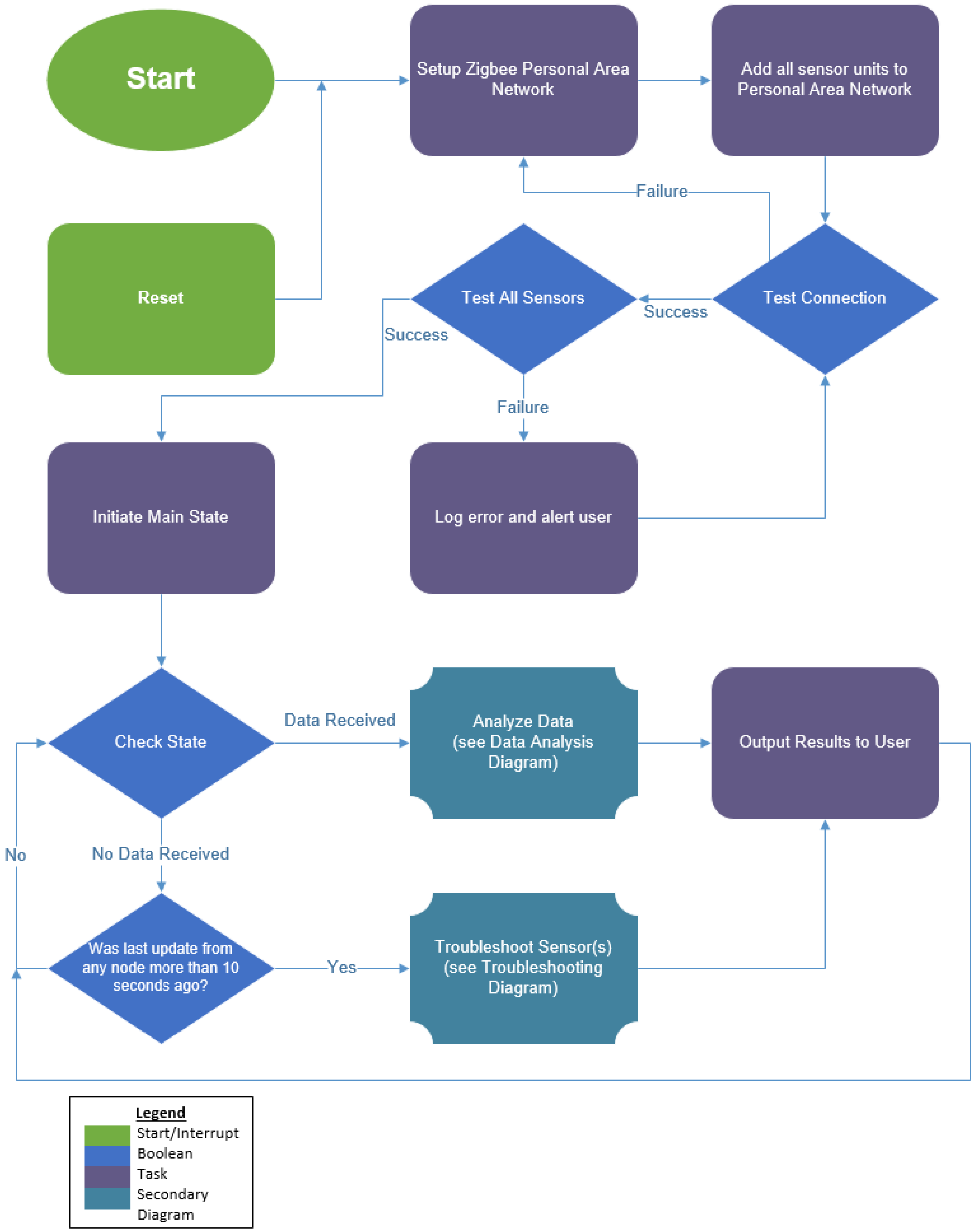


Figure 46 – Software Flowchart: Hub Unit

The data analysis flowchart in Figure 47 below shows a logic cycle that will be completed by the hub unit every single time that sensor data is received from the sensors. It will first use the information obtained by the sensors to determine what the distance is from the sensor to the closest object. It will then run a simple Boolean check to see whether or not the distance is more or less than the threshold value. The threshold value will be chosen based on road width standards and potential trailer widths.

If the calculated value is less than the threshold, it means there is an object (car, truck, motorcycle, etc.) in the sensor's area and should be marked accordingly. So the hub unit will then turn the LED corresponding with the sensor that sent the data on to show the driver that there is an object in the area. For the opposite case where the calculated value is more than the threshold value, the hub unit will make sure the corresponding LED is off to show the driver the area is clear of any obstacles.

The LEDs will be controlled by a digital value. This means a value of 1 turns on the LED while a value of 0 turns off the LED. The other option was to have the brightness of the LED range from 0 (off) to 255 (brightest). The would allow the system to give the user a relative idea of how far the nearest object is. The brighter the LED is, the closer the object is. This could potentially be useful in areas with a large amount of lanes. In Table X below, the example of a 4-lane highway is used to show how the brightness of the LED would work to give the driver a clear idea of his or her surroundings. In this example, the driver using the Truck Smart system is assumed to be in the far right lane.

Table 19 – LED Brightness Levels: 4 Lane Example

LED Status (% of brightness)	Blind Spot Status
0%	All lanes are clear
33%	Vehicle in third lane over
67%	Vehicle in second lane over
100%	Vehicle in first lane over

The goal of this method is to give the driver as much information as possible. In the end, however, it was decided that this system would just plainly be more confusing to the user than a straight forward on/off digital method. The reason for this is that some users may have trouble differentiating between relatively small differences such as 33% and 67%. An error in judgement between these two brightness levels is the difference between two lanes of clearance and three lanes of clearance and could easily result in an accident. This is an outcome that the Truck Smart system will be working to avoid at all costs. For these reasons, the relative brightness system was avoided and it was decided to use a digital on/off system for the blind spot status LEDs.

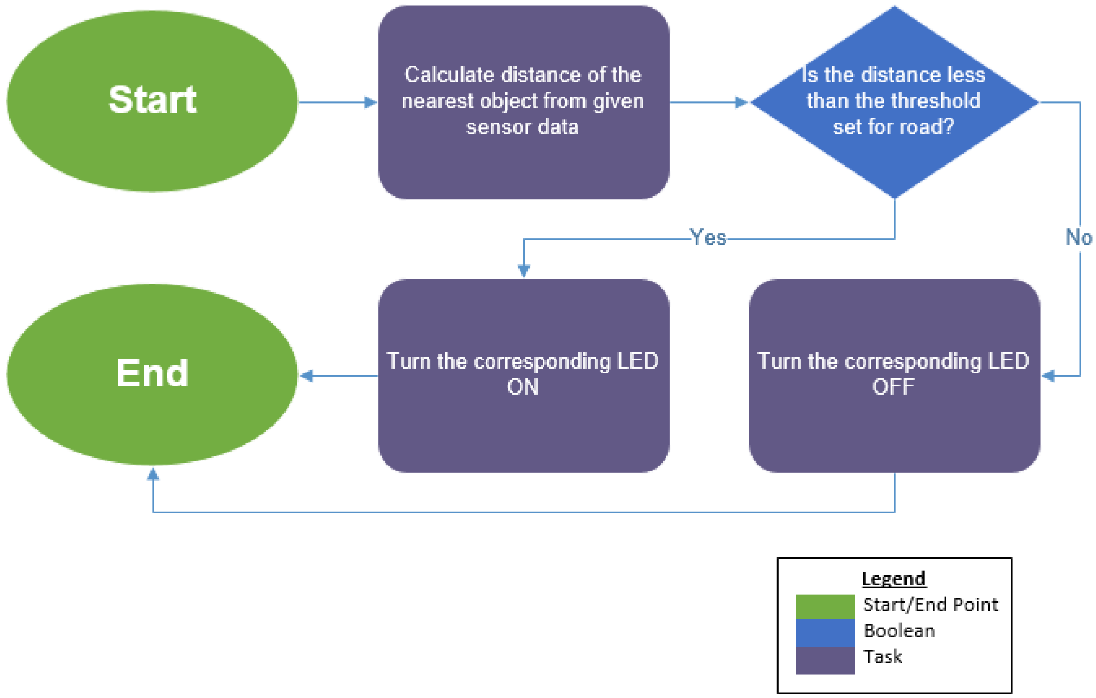


Figure 47 – Software Flowchart: Data Analysis

The Truck Smart system's automatic software troubleshooting steps can be seen below in Figure 48. These steps are in place as a precaution in case a sensor unit develops a glitch while the system is in action. When entering the troubleshooting stage, it is important to let the user know that there is an issue. To accomplish this, the system will enable a blinking yellow warning LED that corresponds with the bugged sensor unit. This will give the user knowledge that there is a problem with that specific sensor and to not fully trust its output.

With the user notified, the hub will then proceed to try and communicate with the sensor unit by pinging it. If a ping is successfully sent back by the sensor unit, that means the network is still working properly and communication is still possible between the two devices. The hub will try to update the sensor data by manually asking for it from the sensor unit. If the sensor unit provides the necessary sensor data without any issues, it will continue monitoring for ten seconds to make sure that the sensor unit is continually providing updated information on a regular basis.

If at any point in the above steps the sensor does not respond in a timely manner, then that means there is an issue in the ZigBee connection between the two units. This is an extremely serious issue and the user should be notified immediately. This is done by changing the blinking yellowing warning status LED to a solid yellow to let the user know the problem has escalated.

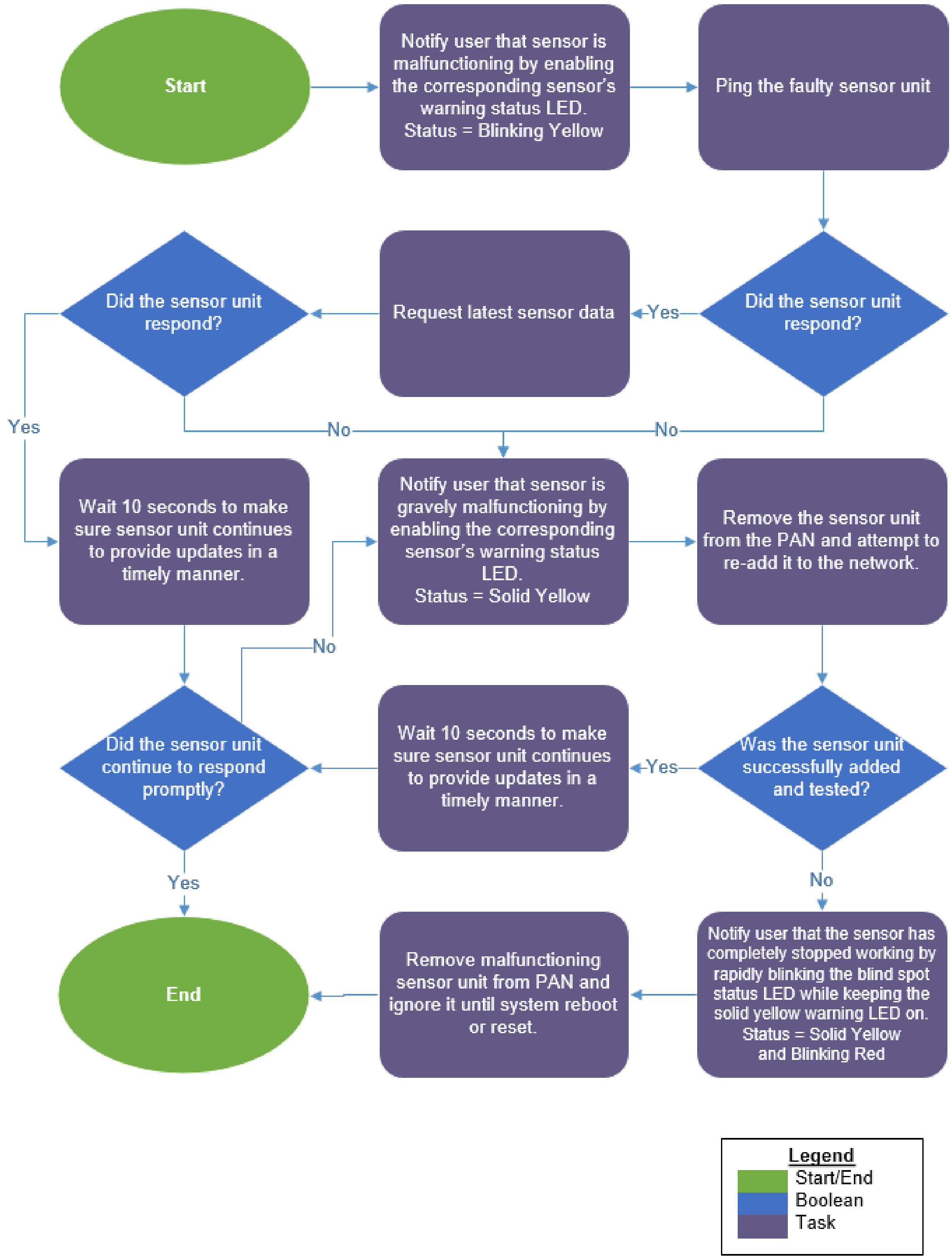


Figure 48 – Software Flowchart: Troubleshooting

At this point, communication between the two units have failed. Therefore, the hub unit will remove the isolated sensor unit from the Personal Area Network. This means that the other sensor units will also sight of the broken sensor unit. This will prevent them from trying to update info by sending it through the broken sensor unit. Once it is successfully removed from the PAN, it will then attempt to re-add it to the network. Since the sensor unit is still powered on, the removal of the device from a PAN should trigger its programming to reset and attempt to join a PAN.

When re-adding the faulty sensor unit to the PAN, the hub unit will follow the same steps as it did in boot that was explained earlier in Figure 45. It will follow all the steps prior to the “Initiate Main Stage” step depicted in Figure 45, except for the initial task of setting up the ZigBee network as the network was already created at boot. The network will not be completely recreated because that would involve removing all sensors from the PAN and then re-adding them. This would be an unwise decision as it would mean the Truck Smart system would be entirely useless for the duration of the reset and that should not happen while the host vehicle is in motion as it would leave the driver completely blind.

If the sensor unit was able to be successfully added back into the PAN, its network communication and sensor output would be tested. If both tests pass, it will then wait and see to make sure it continues to provide timely updates. If that final test passes, the sensor unit can be deemed working again. The warning light for the corresponding sensor will be shut off and the system will exit the trouble shooting stage. On the contrary, if the sensor unit was not able to be added back into the PAN or if it failed any of the connection tests, the sensor unit will be considered beyond help solely through the use of software and will be abandoned until the next system reboot or reset. If it managed to make its way back into the PAN, it will be removed. Regardless, there will be no further attempts until a reboot for it to be added again. The solid yellow warning light will remain on and the sensor’s blind spot status light will blink at a rapid pace to signify to the user that the sensor is not working.

Another potential bug is that the sensor itself could stop working. This will be mostly handled in the code of the sensor unit. If at any point communication with the sensor fails or if it starts to give nonsensical data outputs, the parent sensor unit will notify the hub unit of its failure. From here, the hub unit will act accordingly and turn on the solid warning light and rapidly blinking blind spot status light to convey its status to the user. status light to convey its status to the user.

5.2.5 Network Implementation

The network implementation is divided into three sections. Gathering data to send, sending it, and receiving it. There are a set of coding standards and algorithmic implementations that must be followed. Different sets of libraries are used to communicate across the network as well as to gather the sensor's information.

5.2.5.1 Sending and Receiving Data

One of the greatest advantages of using the ZigBee protocol is that it automatically finds the best way to route data from one node to another with maximum chance of success. ZigBee uses standard networking terms as defined by IEEE which include concepts such as "Data Request (which means to transmit data), Data Confirm (which means the acknowledgment of a data request), and Data Indication (which means to receive data)" (Gislason 112). When the data from any of the sensors in the system is read by the microcontroller, it is subsequently encoded by the wireless transceiver and the first step of the transmission begins. The first step is known as the "data request" when referring to it using IEEE terminology. The second step is the "data confirm" and is a direct result from the data request. It acknowledges the data confirm at the beginning of the transmission. The data confirm can either be a success or failure of the initial request. The data indication is the last and longest step in the procedure and can come at any time.

The data request can come in different formats such as unicast (with end to end acknowledgement), unicast (without end to end acknowledgement), broadcast, and group-cast/multicast. Unicast data is transmitted specifically from one node to the other. Unless the nodes are neighbors and within close proximity, route discovery takes place the first time these two nodes speak to each other. Once end – to – end acknowledgment has been completed the unicast is retried three times. If any of the retries do not go through, the system implements a different route to send the data such as sending it across a different node. Broadcasts is when data is transmitted from one node across all the nodes in the network within a defined radius. Broadcast transmission is used when all the devices in the network rely on each other for their functionality. For example, in a home wireless network when a user is streaming from his cell phone to a television as well as a laptop at the same time, the wireless data coming from the cell phone is broadcasted to all the devices in the network such as the TV, computer, Blu-ray player, etc. The main disadvantage of using broadcast transmission is that is only supported across a small distance range making it unsuitable for the Truck Smart system.

The final data request format supported by the ZigBee protocol is group-cast. This format is similar to the broadcast module, but instead it transmits simultaneously to specific nodes in the network. During development, the programmer creates a set of different groups which will contain a set number of nodes. Data from one node can only be sent to other nodes within the same group. This application is optimal to reduce bandwidth across the system and decrease network usage as well. The Truck Smart system will employ the unicast with end to end acknowledgement since the data will be sent from one node to the adjacent one until it reaches the base.

A difficult feature to predict when using the ZigBee protocol is packet latency or how long it takes for data to be sent in between two nodes. A good assumption would be for latency to be about 10 milliseconds per hop (from one node to the other). However, when one of the devices is down and the data must bounce from another node or needs to be retried, the latency can be significantly longer. Figure 9 explains the concept of latency in a network. The data is being sent from the first node on the left to the last node on the far-right side. For each hop, it takes 10 milliseconds for the data to be moved from one to the other until it reaches the base 40 milliseconds after. After the data request was sent at 0 seconds, it will take 80 milliseconds for the acknowledgement to get back to the original sender node. The data moves across 4 nodes at 10 milliseconds per node. Secondly, the acknowledgment goes back from the final receiving node to the original sender. In cases where retries or when route discoveries need to be implemented due to a system failure or any other type of error in the system, latency can take a lot longer. How much longer depends on the type of system failure or error but in the worst case scenario it can take up to 5 seconds.

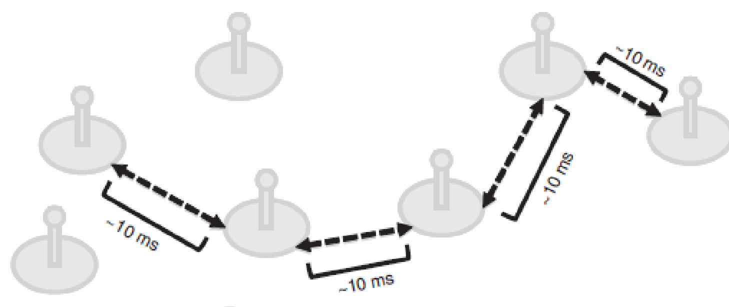


Figure 49 – Network Latency Diagram

For the Truck Smart system, which will use unicast with end to end acknowledgment, the chances of having a system failure are slim to none due to the small range the devices will be sending data across, as well as the number of devices in the network. The system will have only 7 to 8 devices in total and no more than 20 feet in between two nodes. The ZigBee protocol and transceiver employed can support up to 2000 nodes and has a range of 4000 feet respectively. A good practice to prevent errors when sending or receiving information across the network, is to keep the data packets being sent as small as possible. If the wireless channel being used for communication is noisy, and the packets being sent are long, it can result in an increased number of retries which in turn will increase the latency. For the system being implemented, latency needs to be as small as possible since any delay in the output from the sensors can result in an accident. Keeping the packets small will decrease bandwidth as well as latency. Fortunately, the Truck Smart system will use small data packets due to the encoding of the sensor's output. Therefore, even if one of the nodes fails, the rerouting of the data packets will be implemented fairly quickly.

5.2.5.2 Channels

The 2.4 GHz channel is a worldwide unlicensed portion of the RF spectrum for use by many radio products. Many wireless technologies operate in this spectrum such as Wi-Fi, Bluetooth, cordless phones, ZigBee, and even microwave ovens. ZigBee operates in channels 11 through 26, since channels 0 through 10 are used by sub-1GHz 805.15.4 radios. The channels are separated by 5 MHz in the 2.4 GHz band. The 802.15.4 radio is half duplex meaning it allows for listening (receiving data) or talking (sending data), but not both at the same time. The ZigBee protocol does not change channels, meaning that once the network is in operation and the channel is set, it will operate on that one channel unless manually modified.

Table 20 – Scan Duration Times

Value	Scan Duration (ms)	Value	Scan Duration (ms)
0	31	8	3,948
1	46	9	7,880
2	77	10	15,744
3	138	11	31,473
4	261	12	62,930
5	507	13	125,844
6	998	14	251,674
7	1,981		

The ZigBee protocol sets the channel using what is called an "Application Profile" (Gislason 137). For the Truck Smart System, the first step in setting up the wireless communications system is to create the network using the base/receiver device. Each of the devices that will be in the network perform a channel scan. This is a critical step in the process of forming the network and it is also the longest procedure. A channel scan consists of two parts; an energy detect scan, and an active scan. An energy detect scan is used to find which is the quietest channel out of the 14 available. An example of the time it can take to perform the scan is shown in Table 1 below. For the Truck Smart system, which will employ a total of 6 to 8 devices, it can take up to 4 seconds per device in a network of 8 nodes for a total of approximately 35 seconds. The scan sends out a beacon request and determines what other ZigBee devices are in operation and what channels they are using. The system then selects channel in the order of which has the fewest networks in operation, or is the quietest. An active scan is performed when the network is already active. It is used when joining a network and is performed by all the other devices but the hub.

5.2.5.3 ZigBee Personal Area Networks (PAN)

All devices in a ZigBee network can only send data to nodes within the same network. A ZigBee network, regardless of the number of devices that it contains, is called a “Personal Area Network” (PAN) (Gislason 135). Each PAN contains a unique identifier, which helps prevent interference as well as confusion with other ZigBee networks when sending data. A ZigBee PAN is composed of coordinators (ZCs), ZigBee Routers (ZRs), and ZigBee end devices (ZEDs). Only coordinators can form a network while ZigBee routers and end devices can join a network; but not create one. A ZigBee Extended PAN ID (EPID) is a 64-bit value set for the entire network by the network coordinator. Before 2007, ZigBee employed a 16-bit value simply known as the PAN ID. The EPID is set at the time the network is formed and cannot be changed at any time while the network is in operation.

The EPID will change only when the device leaves the network and it is usually not transmitted when sending packets due to its large overhead of 8 bytes. However, when a network update is being performed, the EPID will appear in the beacon results seen by the active scan. A network update is when the manager nodes informs the other nodes in the network about a channel change or a conflict within the network. The EPID is always saved to the non-volatile memory in order for it to be kept within the device during the lifetime of it; unless it leaves the network. The EPID is the primary criteria used to match the joiner to the joinee. There is a total of over 16000 PAIN IDs available to choose from and the system automatically chooses a new one every time a network is created. Having a unique PAN ID is what prevent networks from interfering with each other. For example, in the Truck Smart system there is the possibility of two trucks using the same system while being within close proximity from each other. This can cause interference and data from a sensor from one system ending up being sent to the other system that is operating right next to it.

To prevent this from happening, the ZigBee protocol checks where the data is coming from. Secondly, it compares the PAN ID for each of the devices that are sending the data. Finally, if the ID is defined in the network, it receives and processes the data; otherwise, the data is disregarded. PAN IDs used to be unique for every channel, meaning that for example, ID 0x1234 might exist in channel 14 but it might also appear in channel 15, 16, etc. After 2007, when ZigBee added the feature of “Frequency Agility” which allows the network to change channels if the current channel is no longer appropriate, PAN IDs became completely unique. As a result, separate ZigBee PANs may exist on the same wireless channel.

5.2.5.4 Transmitting in a Simple Network

The initial step when configuring the simplest ZigBee network which can consist of just one sender and receiver device is to start the XCTU software and connect the device to the computer through a USB cable. Secondly, the developer needs to go in the application and select the “Discover Devices” option which will automatically detect any XBee modules connected to the computer regardless of which USB port or what connection type is being used. The first device that must be configured right away is the coordinator device. After the module has been recognized by the computer, a window will pop up in which the developer will select the Product Family, Function Set, and Firmware Version. The product family is automatically selected by the computer. For the S2C device, which is the one being used for the Truck Smart system, the programmer will need to go in the configuration options and set the Coordinator Enable (CE) entry to 1.

This will allow the device to act as the only coordinator for the network that will be implemented in the later steps. The following step is to update the device’s radio’s firmware to work as a coordinator. The next sequence of event involves setting the destination address for the transmissions. These values are determined by the Destination High (DH) and Destination Low (DL) configuration fields. A usual practice is to set the DH to 0 and the DL to 0xFFFF which allows for messages to be broadcasted across all nodes in the network. Selecting the “Write Radio Settings” options is the last step in configuring which will apply all the changes to the device. Now the device is enabled to send data across the network to another device.

The first step in a data transmission is selecting how the data will be sent. A good initial example is to use the console built into the XCTU software. Before sending any data, the “Open connection” option needs to be selected to initialize a connection to the other modules. To configure the receiving device, the programmer must begin by configuring the radio’s firmware to the latest “ZigBee Router AT” version. The code that will be loaded into the Arduino Uno which will be receiving the data, starts by opening a serial connection at 9,600 bps (bits per second) to the XBee module and 115,200 bps to the computer. The code follows by listening to the channel for receiving data and then checking if the XBee serial is available. If the serial is available and data is being sent, it calls the XbeeSerial.read() function from the XBee library which will pick up the data coming from the sending devices in the network. Finally, the data coming from the sending device is stored byte by byte into a string and later outputted to the serial terminal. The Arduino shield is attached to the microcontroller and the XBee S2C device is plugged into the shield. After opening the console in the XCTU software and typing the data to be sent, the serial monitor in the Arduino IDE is opened at the same time. As the data is being transmitted by being typed in the XCTU terminal from the sending device to the receiving device (Arduino Uno), the Arduino serial terminal will display the data as it is being received.

5.2.5.4.1 Software Sketches/Algorithm Implementations

The sketches that will be uploaded to the microcontroller will make use of several libraries. These libraries will help the designing team to call predefined functions while reducing the complexity of the embedded software applications. The AltSoftSerial library will be used to call functions that allow for the serial aspect of the communications in between the XBee module and the microcontroller. The XBee – Arduino libraries allow for building, receiving, and parsing API (application interface) data frames. In order to send and receive data the coordinator (receiving device) will need to have the receiving sketch uploaded to it. On the other side, the sender (router) device will need to have the sender sketch uploaded. Sending a message and receiving a message in between two devices will follow a set of steps. First, the sender (Arduino microcontroller) sends a ZigBee transmit request API frame to the XBee module which will contain the destination address and the message.

For the second step, the receiving device will receive a ZigBee Receive Packet API frame which will contain the sender's address and the message. It will also send back to the sender an acknowledgement that the data was successfully received. Lastly, the sender acknowledges within itself that the receiver could obtain (or not) the previously sent data. This acknowledge is knows as a "ZigBee Transmit Status". The process is illustrated in the following diagram. Each of the vertical lines indicates a component in the system while the arrows represent the messages moving in between them. The transmission starts at the top in the Arduino, the request is sent to the XBee module, and the packet is transmitted. The XBee on the receiving side gets the packet, which goes into the receiving Arduino. Finally, the receiving device sends an acknowledgment (Ack) back to the sending device's XBee module which ends up transmitting the status response to the sending microcontroller.

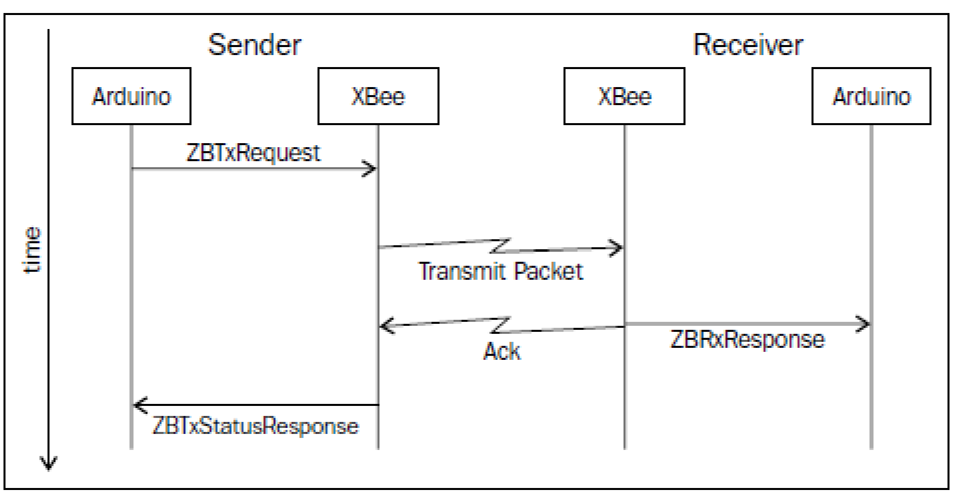


Figure 50 – Network Flow Diagram (used with Permission from Packt)

From the diagram in the figure above, the initial data being sent in between the sending microcontroller and the XBee module is the “ZBTxRequest”. This type of request is known as a ZigBee Transmit Request and it contains several important fields. The first being the 64 – bit destination address which will contain where to send the data packet. The second field is optional and consists of the 16 – bit destination address and is used if the sending device knows the address of the receiving node to save the overhead of looking up an address on the network. The last field is known as “the payload” and consists the actual content of the message being sent. For the Truck Smart system this will be composed of the sensor’s output.

5.2.5.4.1.1 Algorithm Used to Send Wireless Data

The sketch that will be loaded into the all 6 to 8 sending devices in the system will be required to contain a function that sends the packets. This function will need to be called from the main function within the sketch. The function will begin by declaring an instance of the ZBTxRequest which will prepare the ZigBee transmit request. Secondly, the transmission address will be set to zero in hexadecimal enabling to send the packets directly to the coordinator/base device. A string type variable is declared which will contain the payload or the actual message being sent. This message will be the actual bits that state whether the sensor has been activated due to a vehicle or object being in the blind spot section. How the message that goes in this string is populated will be discussed in the later section that talks about collecting the sensor’s data. After the string has been created the txRequest.setPayload function will be called. This function sets the message that will be sent to the previously declared string. Finally, a variable used to tell the status of the sent message is declared. Inside this variable the XBee library function named “send and wait” will begin the transmission request and set a threshold in which it should stop waiting for feedback from the coordinator.

The threshold represents how long it should wait for a response and it is set to a value in milliseconds. With the default configuration, a ZigBee packet is tried up to three times lasting approximately 4800 milliseconds. The message status will most probably be received within this time, therefore setting this function’s parameter to 5000 milliseconds (5 seconds) will be an efficient implementation. The send packet function will need to be called from the loop function in the sketch. Arduino uses the loop function to represent the constant functionality that the microcontroller must implement while being powered. Inside the loop function, the XBee loop function will be called. This function allows for the module to listen if there is data to be sent. The XBee loop function needs to be called regularly to poll for new data. Since the programmer will not know exactly how long the messages will take to be sent, a helper variable will be created to check if it is time to send a packet. The variable tracks the time of the last sent message. The value is continuously calculated and if it is longer than ten seconds, then a packet is transmitted. The following figure represent the implementation of the two functions described in this section. For now, the transmitted payload is replaced with "data" but it will later on be replaced with the sensor's output.

```
void sendPacket() {
    // Prepare the Zigbee Transmit Request API packet
    ZBTxRequest txRequest;
    txRequest.setAddress64(0x0000000000000000);
    uint8_t payload[] = {'d', 'a', 't', 'a'};
    txRequest.setPayload(payload, sizeof(payload));

    // And send it
    uint8_t status = xbee.sendAndWait(txRequest, 5000);
    if (status == 0) {
      DebugSerial.println(F("Succesfully sent packet"));
    } else {
      DebugSerial.print(F("Failed to send packet. Status: 0x"));
      DebugSerial.println(status, HEX);
    }
}

unsigned long last_tx_time = 0;

void loop() {
  // Check the serial port to see if there is a new packet available
  xbee.loop();

  // Send a packet every 10 seconds
  if (millis() - last_tx_time > 10000) {
    last_tx_time = millis();
    sendPacket();
```

Figure 51 – Transmitting Data Algorithm (used with permission from Packt)

To send the data through the transceiver, the code in the main function must make use of polling. This functionality will allow for waiting in changes at the sensor. The changes will only happen if a vehicle in the blind spot will trigger the sensor. Polling regularly checks a pin, and reads the buffer to see if a changed has occurred since the last read. The Truck Smart system will only send data wireless if the sensor has been triggered otherwise it would be a waste of resource to create congestion in the network if the sensor is not active. Polling must be done quickly to prevent events from being handled too late or even missing an event. An alternative to polling is using interrupts. For interrupts, an external event will cause a function that will be called which will run an algorithm that takes care of that specific event. For example, in the Truck Smart system, if a sensor is activated, the interrupt will call the send data function and the bits from the sensor will be transmitted to the base microcontroller.

5.2.5.4.2 Receiving Data Algorithm

Per the ZigBee protocol, whenever a packet is received, the module sends a "ZigBee Receive Packet" API frame, therefore a callback registration will need to be implemented. The XBee Arduino library contains several integrated functions that can handle a lot of the workload when it comes to receiving wireless data. There is a set of object of type "ZBRxResponse" (ZigBee Receiving Response) which are used to represent the previously described API frames. There are two important functions that will allow the programmer to differentiate in between all the packets being received. The first function is the "getRemoteAddress16" function which will yield the programmer the address of the device that sent that specific message. This will allow for differentiating which of the sensors has been activated. The second function is the "getData" function which decodes the received message into a string. Calling this function with no arguments will return a pointer to the entire message received which can be passed to another piece of code should the need to perform modifications to the received message arise.

For the initial setup, the received message will simply be printed in the serial terminal for the receiving microcontroller. However, for a future implementation, the goal is to have a set of LEDs that will correspond to each sensor. After the message has been received stating which sensor is active, a signal to light up the corresponding LED will be sent. The receiving algorithm for the initial implementation will just print where the message is coming from, and what the message consists of itself. The algorithm uses a callback function named "processRxPacket" (process receiving packet). This call back is the very specific response that will enable the transceiver to know that it must follow the functionality inside of it. The callback function used to receive data has a unique name for all modules, therefore it will be critical to use the corresponding name for the transceiver in use. The two parameters that will be passed to the "Process Receiving Packet" function will consist of the ZigBee Response packet acknowledgement and the pointer to the received message. Inside the function's declaration, the algorithm calls the required functions to acquire the address of the device that sent the message, and then follows on to read the message itself. Specifically, the "read data" function is the one that performs the decoding of the message and displays it in the serial terminal.

The algorithm is show on the figure to the right. The loop function (not shown) simply calls the "XBee. Loop" predefined function that waits constantly to receive a message. If a message is received the functionality inside the setup function will be performed.

```
void setup() {
  // Setup debug serial output
  DebugSerial.begin(115200);
  DebugSerial.println(F("Starting..."));

  // Setup XBee serial communication
  XBeeSerial.begin(9600);
  xbee.begin(XBeeSerial);
  delay(1);

  // Setup callbacks
  xbee.onZBRxResponse(processRxPacket);
}
```

Figure 52 – Receiving Data Algorithm (used with permission from Packt)

5.2.5.5 The API Module

The module of communication used in the previous section is known as the “transparent mode”. In this mode, the data received on the serial port is transmitted as it is by the radio. However, this mode is very limited since the data can only be sent to a single preconfigured address. In addition, when there is data being sent from multiple nodes, their messages might end up being interleaved. The Truck Smart system will have an approximate of 6 to 8 nodes in total, all sending data at the same time. As a result, it will not be a good approach to use the transparent mode. Application Programming Interface (API) is a good alternative to use in the Truck Smart system. In this mode, “all serial applications use a binary protocol consisting of API frames” (Kooijman 17). The layout of a sent frame is shown in Figure 10.

0	1	2	3	4 ... n	n + 1
Frame Start 0x7E	Frame length MSB	LSB	Frame type	Frame Data	Checksum

Figure 53 – Layout of a frame using the API module (used with permission from Packt)

Each of the frames being sent using the API module have a unique structure that tells the receiving end the operation that should take place and the event that occurred. Another advantage of using the API module is that it allows the receiving device to know the source of the frames being sent. In the encoding of the frames there are several parameters such as frame length, frame type, transmission options, and the data that is being sent. The layout of these parameters allows the developer to build more complex networks such as the one in the Truck Smart system that contains multiple devices sending wireless data at long distances, all at the same time.

In order to update the devices in the network to use API, the developers will need to go in the configuration options of the devices in the XCTU software and set the AP field to 2. For some types of transmitters, the firmware needs to be modified but for the S2C there is no need to apply this change since the same firmware can run on all transmission modes.

5.2.6 Network Design

The software used to design the network and assign the required parameters is called "XCTU". The acronym stands for XBee Configuration and Test Utility. Through this application the network configurations are created and written to the wireless transceivers. Each device's configuration depends on whether it is a coordinator or router.

5.2.6.1 Starting and Joining a Network

To begin sending data between the XBee modules, they will all need to be joined to the same network. The previously described examples used a single coordinator and router. However, for the Truck Smart system this will not be the case. The application will have a single coordinator but will employ multiple routers. In addition, the API module will be employed since the transparent module is very limited in functionality and therefore not practical for the desired use. The software application XCTU allows the network designer to set the nodes as coordinators, routers or end devices. A coordinator is the device that starts a network and allows others to join it. Each network must have exactly one coordinator. For the Truck Smart system, the coordinator is the microcontroller that will be located in the truck's cabin. Not only will this device do the raw functionality of the coordinator, but it will also be responsible for receiving the encrypted data from all the other devices, decoding it, as well as displaying it. The coordinator is definitely the most important device in the Truck Smart system which is why it must be carefully handled and configured.

A router is a device that will join a network and forward data pockets on behalf of the other members in the network which allows for creating a mesh network. However, routers cannot enter sleep mode therefore will need to be fully powered all the time even when the sensors are inactive. An end device is a module with very limited functionality. It cannot do any routing itself and must rely on routers or coordinators to relay their messages to the destination node. The good aspect of end devices is that they can be put in sleep mode to save power while inactive. However, for the Truck Smart system, the use of end devices is not suggested since we want each sensor to send the data independently without relying in other devices. In the ZigBee protocol, a network is identified by the previously discussed Personal Area Network Identifier (PAN ID). After the coordinator is started with its default settings, it will start a new unencrypted network by selecting a random PAN ID and a random channel.

When a router or end device starts, it will begin scanning all channels for existing networks, select one, and join it. This process is automated only if the coordinator device allows for other devices to join without restriction. This will not be the case for the Truck Smart system since the designers do not want any device to be able to join. Therefore, before plugging the transceiver into the microcontroller, the designer will need to manually set the coordinator to restrict the devices allowed to join a network to only the ones that know the PAN ID. This will prevent unauthorized devices from accessing network data without the coordinator's approval. The XCTU application allows the network engineer to manually set the values for the coordinator and router transceiver devices after loading the firmware version into it. The following figure shows the software screen that allows to modify these parameters as well as the specific values that the programmer can assign to the networking properties.

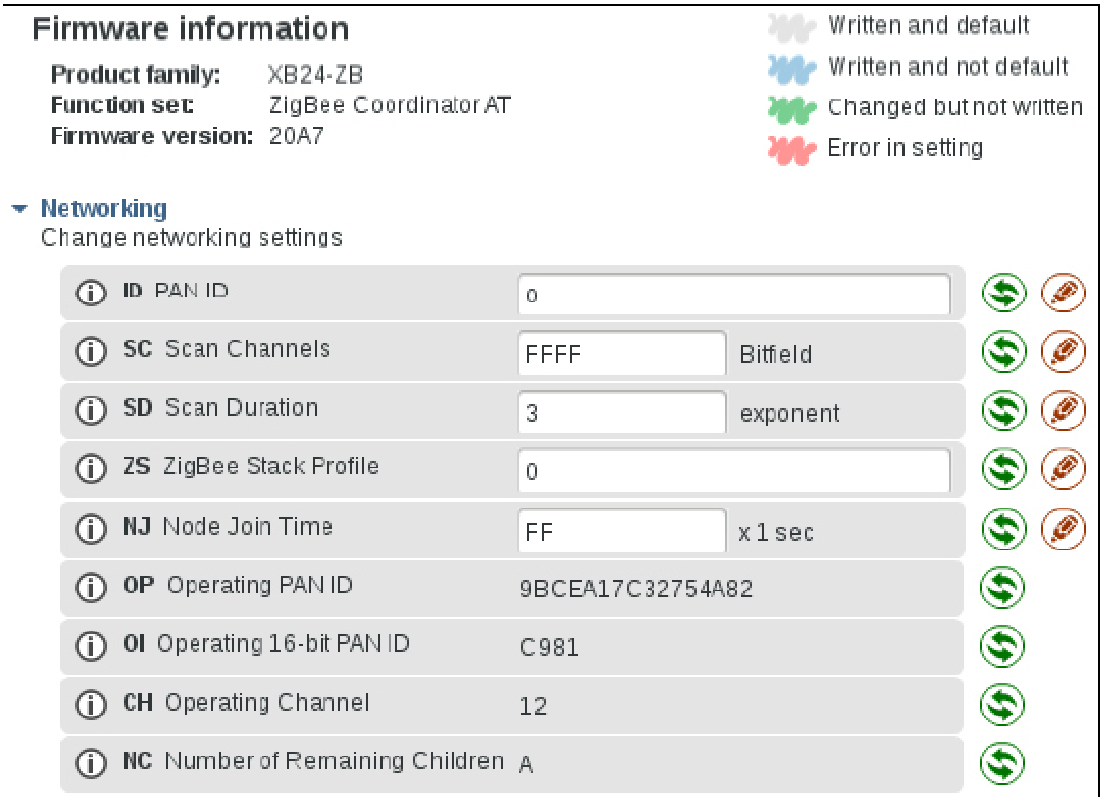


Figure 54 – XCTU Configuration Window

At the top of the figure above, the software application displays the Product Family, device's function, and firmware version. For the Truck Smart system, these values will be filled with XB Pro – S2C, ZigBee Coordinator, and 20A7 respectively. These values can be modified by the programmer at the initial startup when the firmware is loaded into the module, except for the Product Family which is preset by the software. Almost all the values are configured as hexadecimal numbers. The Operating PAN ID is the ID of the network that was successfully joined and the one created by the network's coordinator. The Operating Channel field is the wireless channel in the 2.4 GHz frequency used by the network. It can be any value from 11 to 26 and it can be set only by the coordinator. The channel is the only value displayed as a regular decimal constant. The figure displays the network operating in the 12^{th} channel, but this value will vary depending on conditions such as network bandwidth, latency, how many networks are in operation in the surrounding area, and how many devices will operate on the same channel. This value is usually automatically selected by the coordinator when the network is created but the programmer has the option of manually setting it. However, it is not recommended to manually set this field, since the device will automatically find the most efficient channel to operate if it is not manually set. The second PAN ID that appears in the figure is 64 – bit value used mostly for scan requests, beacons, and network joining. For data transmission, the 16 – bit value is the one used to save transmission time.

5.2.6.2 Cross Network Interference

The chances of another network using the same PAN ID as the one in the Truck Smart system are extremely small. However, if this were to happen, the conflict will be automatically detected by the system, and a new 16 – bit PAN ID will be generated and applied to all the devices within the network. The ID configuration field which displays a “0” in the figure above, allows for preconfiguring the 64 – bit PAN ID. For coordinators, it means that the started network’s ID will be the one set in this field. For routers and end devices, it means that all other networks will be ignored and this will be the ID of the network the devices will join at startup. Every device capable of interacting with another wireless device is assigned a MAC address or a unique identifier number. These identifiers are distributed by the IEEE networking committee to ensure that all of them are globally unique. For the XBee transceivers, they have what is called an “Extended Unique Identifier” (EUI – 64). It is a 64 – bit address that can be found in the XCTU software application by looking into the SH and SL values. The SH are the top 32 – bits of this address, while the SL represent the lower 32 – bits. The value will also appear in the device’s sticker.

Besides having an Extended Unique Identifier, all devices possess a 16 – bit short address. The address is assigned to each device when they join the network. The address will not change unless the device leaves the network. There is a small probability that two devices might be assigned the same short address which will cause serious issues in the network addressing. If this were to happen, XCTU will automatically detect the issue, assign new address for one of the modules, and overwrite the previous configuration. The designer’s first task after assigning networking parameters to all the devices that will operate on the wireless network is to perform a network scan. A network scan allows the programmer to know more about all the devices that have been authorized to communicate in the network. The scan is performed using XCTU’s “Discover Radio Nodes” operation in the top right window.

The results of the scan will consist of aspects such as graphs containing links for each node in the network, which networking mode the device operates on (AT, API), how are they connected, and what distance are they from each other. The network scan is a useful feature to prevent cross network interference since it allows the network designer to know how to identify the devices that belong to his or her network. If a network scan is performed and it shows devices that should not be connected to the network, the programmer has the option of kicking them out of the network by changing the PAN ID. This would be a very useful feature if another XBee device connects to the network by mistake while the network does not have a security encryption implemented to prevent from unauthorized modules. An extra feature included with a network scan is the ability to modify network parameters for devices that were encountered in the network, even if they are not connected to the computer.

5.2.6.3 Network Security

The use of wireless communication allows the Truck Smart system to be a more convenient and easy to use application than if it were to use wires for communication in between the microcontrollers. However, the use of this technology opens the possibility for others to sniff the data being sent or unauthorized devices to send packets into the network. These issues can cause serious abnormalities in the system and lead to a complete system failure. For this reason, securing the network with the proper encryption is a significantly important task that must be implemented before the system is deployed for the first time. The ZigBee protocol makes network encryption as easy as possible for the programmer by implementing it in just a few clicks and modifications. The first step in performing a network encryption is to go in the device's configuration window on the XCTU software application and enable security. This will be done by setting the EE field to 1. Modifying this field will make most packets being sent across the network to become encrypted. It will also protect them using a "message integrity code". This will prevent unauthorized devices from inserting messages into the network.

The encryption will not be performed across every single packet that is being sent in the network. Sent packets at the lower level such as beacon requests, join requests, do not need to be encrypted since they will not contain security - critical information. However, the actual data that is being sent such as the sensor's output will be fully encrypted and protected across the network. After the encryption has been enabled, the next step includes setting the network key. The network key is equivalent to entering a password when logging in to check your email. All members of the network must know this key to be able to send and receive data across the network. The network key can be configured by modifying the Network Key (NK) field in the XCTU network configuration window. Setting the value to 0 will create an automatic key when the network key is created. This key will be 64 – bits long. The Truck Smart system will be employing this encryption mechanism in order to prevent unauthorized nodes from injecting data into the network. For example, if another truck with the system attached is within close proximity to the one in operation, and tries to wrongfully send data to the network it does not belong to; then the system will prevent it from it. This will happen due to the illegal node not knowing the encryption key.

When a new module joins the network, it automatically receives the encryption key from what is called the "trust center". The trust center is who is responsible from assigning the keys to the nodes authorized to send data within the network.

The trust center runs inside the coordinator which will be located inside the truck's cabin. However, it can also be run inside other modules should the designer have the need to. If the Network Key is set to 0, then the coordinator will send the key unencrypted to any device that tries to join the network which creates a significant security risk since it allows for someone to spy during the key distribution and receive it as an unencrypted value. To prevent this from happening in the Truck Smart system, will disable external nodes from joining and only allow for nodes to join for a brief period of time. The nodes allowed to join will be required to know the encryption key when they try to enter the network. The network designer will be required to turn off automatic joining from any node and enable new devices to be added only when there is a new node to add. This will lead to the network being secured while "network joining" is disabled. To make the coordinator send the network encrypted, the XBee modules allow for the implementation of what is known as the "trust center link key". This key is manually configured in each module and set before it tries to join or create a network. If a device does not know the trust center link key, then it will not be able to receive the encrypted network key and therefore not able to read any data moving across the network. The trust center key allows for an additional layer of encryption on top of the regular network key providing maximum security to the wireless communication in between the devices that form the network. An encryption key can be up to 128 - bits long, making it next to impossible to crack using brute force or even super computers.

5.2.6.3.1 Setting up a Secure Network for the System

After previously describing how the ZigBee protocol works in terms of security implementation and how the specific XBee transceiver being used for the system allows for setting the security parameters, the next step is to go in the physical transceiver and modifying the security - related fields. The network designer wants to use the maximum security allowed by the protocol, therefore the use of a trust center link key as well as a network key will be implemented. The updated fields will need to be implemented in both coordinator devices as well as router devices. Router devices will be the transceiver modules attached to the microcontrollers that house the sensors. To enable security in the network, several fields need to be updated in the module's configuration. The "EE" field will need to set to 1 to enable encryption within the network. The KY (Encryption Key) field will need to be modified to select a random 128 – bit "trust center key", or the user can manually enter it; but this is not recommended. The trust center key should not be confused with the regular network center key since they both represent different values. For now, the encryption options (EO) field will be left blank. The NK (Network Encryption Key) will be left blank that way the system will randomly generate the 64 – bit network encryption key. An initial layout of the modified security parameters can be seen in the following figure.

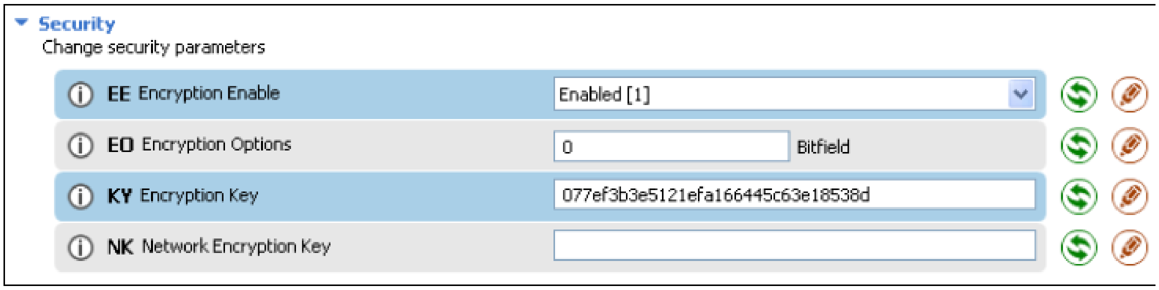


Figure 55 – Modified Network Encryption Parameters

5.2.7 Sensor Implementation

The HC–SR04 ultrasonic sensor works by sending a soundwave, and calculating the time it takes to receive it back. Based on this value, the algorithm determines if there is a vehicle in the blind sport or not. If so, the data is transmitted to the coordinator device.

5.2.7.1 Collecting the Sensor's Data

The device that will be used to detect whether there is a vehicle in the blind spot will be the HC – SR04 ultrasonic sensor. This module offers excellent non – contact range detection with high accuracy and stable readings. The device can detect objects at distances from 1 to 13 feet long. Its operation is not affected by sunlight, wind, snow, or rain (unless it is extreme to the point where you should not be driving) making it an excellent device to be used in the Truck Smart application. Since truck drivers work all over the country where the weather will vary depending on location, picking the SR04 makes it a great choice. The device comes complete with an ultrasonic receiver and transmitter module. It contains a total of four pins and operates with a voltage of 5 Volts (DC) and a working current of 15 milliamps (mA). The four pins are labeled as VCC (supply voltage), GND (ground), TRIGGER (input), and ECHO (output). The "New Ping" library will be used to help handling the sensor's functionality.

The sensor will be connected to the microcontroller for power supply as well as for input and output handling. Its VCC pin will be connected to the 5V output pin from the microcontroller while its GND (ground) pin will be connected to the ground pin from the MCU as well. The ECHO (output) pin can be connected to the one of the digital input/output pins but for the initial design pin 12 will be used. The TRIGGER (input) pin will be connected to pin 11. The choice of the pins is based on the fact that pins 8 and 9 will be used for the serial output of the XBee transmission therefore these two cannot be used to implement the sensor's functionality. The following figure represents the previously described sensor layout with the microcontroller.

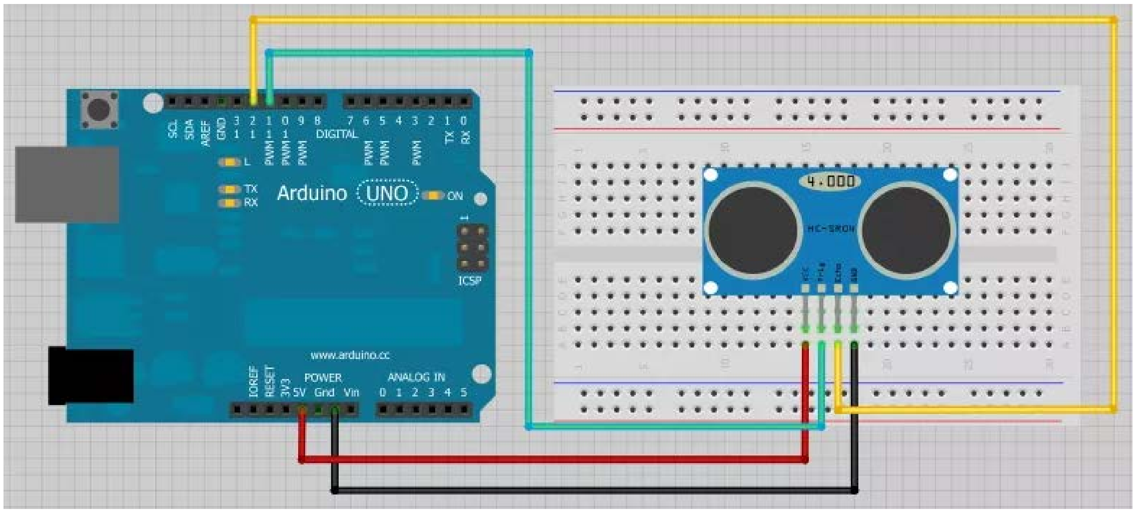


Figure 56 – Sensor’s Layout and Integration with the System

The breadboard that appears in the picture will be replaced with a printed circuit board (PCB). In addition, a new power supply will be integrated and more components will be added to increase the functionality of the system in the PCB.

5.2.7.2 Sensor’s Reading Algorithm

The algorithm used to read the sensor begins by declaring the pins that will be used as input or outputs. The trigger and echo pins that are used to read or send data to the microcontroller will be labeled as an input and output respectively. The sensor will be triggered by a high pulse of 10 or more microseconds. To simplify this statement, the transmitting part of the sensor will send a pulse. If the receiving part gets this pulse back within 10 microseconds, then that means that there is an object in the blind spot. To ensure a clean high pulse is received, and since the initial status of the sensor might be unknown, an initial low pulse is sent to the trigger pin and a delay of 5 microseconds is set. After it, a high pulse is sent with a delay of 10 microseconds. Finally, a low pulse is sent with no delay clearing the pin from any erroneous data that may cause an invalid read. The combination of low and high pulses will help set the sensor to an initial state in which it will wait for a vehicle to trigger it.

Even though at the beginning of the algorithm, the Echo pin was set as an input, the procedure is done one more time after the trigger pin has been cleared. The “pulse in” function will be called to detect whether there is a vehicle in the blind spot or not. This function will consist of two parameters, the first being the pin that will know the sensor has been activated, while the second is whether a high or low signal was detected. For the Truck Smart system, the two parameters in the function will consist of the ECHO pin which will read the sensor’s output, and the HIGH signal which only happens if the sensor has been activated. The High value is generated by the bouncing soundwave coming from the object the signal collided with. After receiving the high signal, it will measure how much time (duration) it takes to get it back to determine how far the object is from the sensor.

When the “pulse in” function is called, its result will be stored in a variable that represents the duration in microseconds from the sending of the ping to the reception of its echo. If there is not a vehicle in the blind spot, then there will not be any echo being received back, therefore the duration will be zero. Last but not least, the duration value is printed in the terminal in inches or centimeters by performing the corresponding mathematical conversions. A delay of 250 milliseconds is set to create a small break in between readings. The output of the sensor will be read as a value in centimeters. For the system, the programmer will want to set 6 feet as the threshold limit when a car is in the blind spot. After the duration has been calculated, the distance will be calculated in centimeters. An if statement will be created which compares if the calculated distance is less than 183 centimeters (6 feet). If it is, then a string will be set which will contain the sensor’s location within the truck, and a statement displaying that the blind spot is “hot”. The string will be copied to the payload and sent across the wireless network. The use of the “New Ping” library will allow the programmer to skip most of the steps described above since it has predefined functions in which the programmer will simply need to pass the parameters to it. Another predefined function will be called to calculate the distance. Finally, the algorithm will check if the distance is longer than the threshold. If so the data will be sent to the coordinator microcontroller.

Even though most of the functionality described in the section is handled by the predefined libraries, it is critical to understand how it all works before merging it with the wireless sending algorithm. The payload to be sent in the “Send Data” function has been changed to “Sensor 1: Hot”. This simply means that the sensor has been activated due to a vehicle being in the blind spot. The sensor number or message can be easily modified such as “Sensor X: Active” (X being the sensor’s location within the truck and Active meaning there is a vehicle in that corresponding location). The truck’s driver will need to be informed of where the sensors are located within the truck that way he or she will know which one has been triggered. The Arduino sketch merging the sensor’s reading and sending the output of the data through the network is shown in the following figure.

Simple functionality such as declaring the sensor’s pins as outputs or inputs, importing libraries, or setting up the serial ports will not be shown. Instead, the shown algorithm focuses on setting up the sensor by calling the “sonar” function from the “New Ping” library and passing the Trigger, Echo, and distance parameters. An initial delay of 50 milliseconds is set and the sound wave’s echo distance is calculated. If this value is less than the one needed for 6 feet, then there is a vehicle at this distance or less, so the value in the payload string is sent to the coordinator.

```
// NewPing setup of pins and maximum distance.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, TRIGGER_DISTANCE)
unsigned long last_tx_time = 0;

void loop() {
  delay(50);
  unsigned int distance = sonar.ping_cm();

   if (distance < TRIGGER_DISTANCE)
   {
     xbee.loop();

     // Send a packet every 2 seconds
     if (millis() - last_tx_time > 2000)
     {
       last_tx_time = millis();
       uint8_t payload[] = {'S', 'e', 'n', 's',
       'o', 'r', '1', ':', 'h', 'o', 't'};
       sendPacket();
     }
   }
 }
```

Figure 57 – Sensor Reading Algorithm (used with permission from Packt)

Flow of the Code:

- Serial communication is initialed
- Variables are established
- Pin modes are assigned (*trig* as Output and *echo* as Input)
- Inside a loop, *trig* pin is assigned low for 2 microseconds and high for 5 microseconds
- Duration for which the *echo* pin was high is measured
- Based on this duration, the distance in cm is calculated by equation:

$$\frac{\#\ of\ microseconds}{29/2}$$

Equation 8 – Sensor Feedback Threshold

- If the distance is less than the threshold distance, signal is sent to turn on the corresponding LED in the cabin.

6.0 System Housing

In order to house and protect the Truck Smart system, a housing must be designed to withstand the expected conditions the system will be facing. Some environmental factors include 70+ mph wind, rain, heat, freezing conditions, and condensation. To tackle the conditions concerning water, the encasement must have tight tolerances and possibly rubber seals around all system-critical areas. The provisions for heat should be designed through means of materials research and providing ventilation ports to areas at risk for over-heating. Freezing conditions can cause cracking to the case if the design is too thin and brittle due to expansion/contraction of the case material. Condensation will be handled through channeling excess water from system-critical areas.

6.1 Aerodynamics of the Housing

Throughout the Truck Smart system's lifespan, it will be exposed to many elements of the environment. The most forceful of these elements will be wind. The assumption is made that, during daily use of this system, it will be exposed to winds of 70+ mph over long durations of time during travel on major interstates.

To ensure the operation and durability of the Truck Smart system, some basic aerodynamic principles must be applied to the housing design. The first and foremost principle of concern is drag. According to NASA's Glenn Research Center, the drag coefficient "contains all the complex dependencies and is usually determined experimentally". While NASA's drag coefficient calculations may be much more complex, the concept remains the same; decrease drag to improve the efficiency of the overall system. Drag is caused by a low-pressure system, called a "turbulent wake" in the tail-end of the airstream. This force has a tendency to "drag" the object in the opposite direction of travel. Larger turbulent wakes are associated with higher drag coefficients. As housing shapes are concerned, the shape that will produce the smallest turbulent wake is a housing with a raindrop design.

A concept model of the Truck Smart peripheral sensor housing is shown in Figures 58, 59, 60. This design meets the aerodynamic needs and also provides a practical amount of space to enclose the Truck Smart system's components. An area has been extruded for the sensor, battery pack, and PCB. Vents should be added to allow for heat dissipation. Areas to place the magnets have also been included on the bottom of the housing.

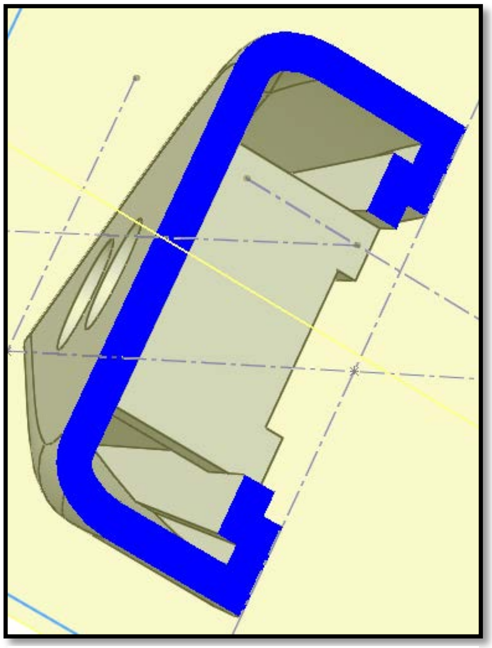


Figure 59 – Peripheral Sensor Cutaway

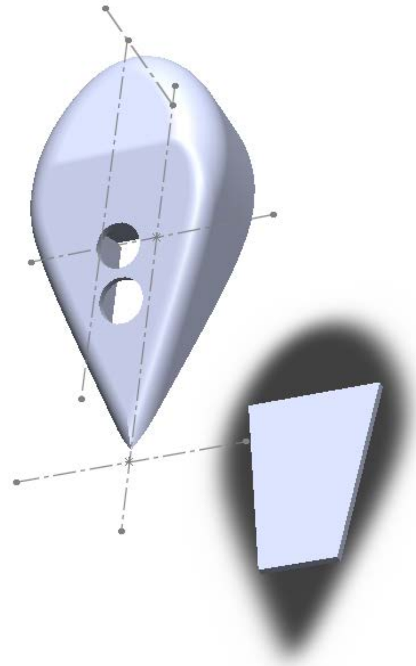


Figure 58 – Peripheral Sensor Shell

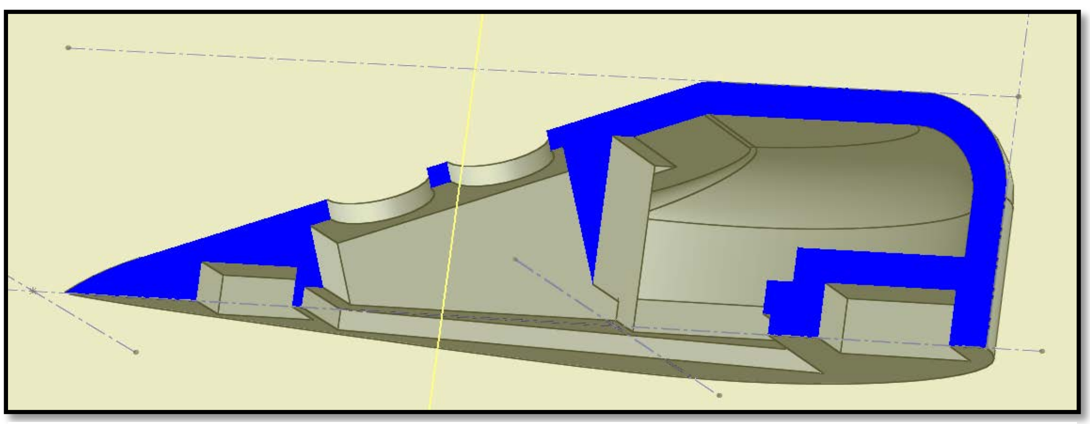


Figure 60 – Peripheral Sensor Horizontal Cutaway

6.2 Weatherproofing

The Truck Smart design is intended to be durable and survive any type of weather. Rain is a danger to the system and can cause system failures if not provisioned for. The primary areas of focus for preventing housing leakage are near the sensor ports and on the bottom flat surface cover plate. Rubber or silicon seals can be used to block moisture from seeping into the enclosure. The sensor port holes will be designed to a millimeter tolerance and will include a set of O-rings to prevent water from entering the system. A backup option would be to design drainage holes on the underside of system-critical areas. Adding vents to the design may add unnecessary risk to the system during inclement weather.

6.3 Magnets & Materials Options

The magnets to be used on the Truck Smart system must be able to keep the system mounted to the trailer at wind speeds of at least 70 mph. The permanent magnet of choice would be a rare-earth magnet of the neodymium type. Neodymium magnets, also known as "NIB" magnets, are composed of an alloy containing Neodymium, Iron, and Boron, according to NdFeB-Info.com. NIB magnets are relatively affordable, can resist temperatures up to 150 C^o, and come in compact sizes. These characteristics make Neodymium magnets an excellent choice for integrating into the Truck Smart system.

To ensure the sensor peripherals will stay mounted to the trailer during high-speed operation, there are calculations that must be done to confirm the drag force and the force of gravity will both be countered by the pull force provided by the magnets. This concept can be compared to the calculations necessary for overcoming the normal force when considering static friction in a system. For calculation purposes, the assumptions being made include: A) the two magnets will each have an area of 1 $inch^2$, B) the magnets will have a thickness of 0.125 inches, C) the total weight of the peripheral sensor is approximately 1.5 lbs. (0.68 kg), D) the drag coefficient of the streamlined half-body shape is 0.20, and E) the area of force impact is 7.00 cm^2 (over-estimate for safe measure) having a streamlined airflow. With these assumptions in mind, the calculated drag force is estimated to be 3.85 N in the horizontal plane. The gravitational force from the assumed 0.68 kg peripheral sensor mass is 6.65 N. When integrating the magnets into the peripheral sensor design, the distance between the magnetic surface and the trailer surface must be taken into account. To prevent paint scratches, the design will keep the magnet a distance of less than 2 millimeters from the surface. Direct contact can cause aesthetic damage to the vehicle and will deem the Truck Smart system as the liable cause for damages. In Figure 61, the pull force of the neodymium magnet is compared to the distance at which it is used. For this calculation, a distance of 1.5 mm was assumed. The expected strength of the magnet (based on experimental data from K&J Magnetics) is a pull force of 5.46 lbs (24.3 N), or roughly 48.6 N for a two magnet system.

Figure 61 provides a visual of the force vectors the Truck Smart peripheral sensor will be exposed to. From the calculations, the two magnets combined pull force should over-compensate for the combined forces of drag and gravity. If these force vectors were not compensated for, the peripheral sensor would be at risk for bowing off of the trailer, or falling off the trailer from the pull of gravity.

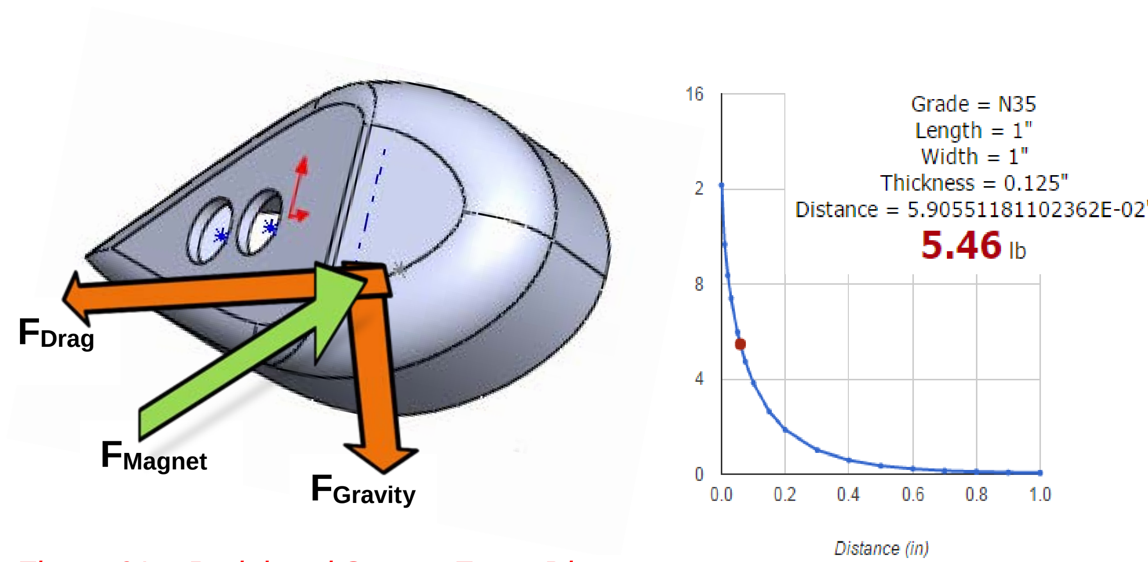


Figure 61 – Peripheral Sensor Force Diagram

There are concerns that arise when integrating strong magnets into any type of microelectronic device. Magnetic fields have the potential to influence the current flowing through a circuit in unpredictable ways. When magnets move around a circuit, or when inductors switch on and off, large temporary current loads are created due to quick changes in voltage caused by the varying electromagnetic field(s). If a magnet is a part of a moving system, there is the possibility that the re-directed flow of current can damage the components within the circuit. Many circuits containing inductors provision for these dynamic current states by having diodes placed in strategic locations. Luckily, the magnets used in the Truck Smart system will be in a fixed state. Therefore, circuit safety components (like diodes) will not need to be integrated into the Truck Smart system since these detrimental effects of the permanent magnets are not anticipated.

6.4 Material Considerations and Costs

Once the Truck Smart system is fully integrated onto a PCB, the housing must be adjusted and prototyped to accommodate the system's size and shape. Currently, the most practical method of prototyping is 3D printing. For small scale production and demo purposes, each of the six peripherals will be 3D printed with PLA, ABS, or resin. These material types have similar specifications and are capable of creating tight tolerance products. There may be some support material necessary during the printing phase due to the shell-like structure of the housing. The need for support material is solely dependent on which plane of the model is chosen as the print base. The chosen supplier for prototyping is the University of Central Florida Idea Lab. The price of the housing is based off the amount of raw material necessary for a print. In the case of the concept housing, the volume is approximately 103 cm^3. At the current premium price of 6 cents per cm^3, the materials cost is expected to be \$6.18 per peripheral sensor housing. This cost can be reduced through shedding material volume in a refined housing design. During the prototyping phase, PCB mounting brackets will be incorporated into the final rendition to keep the internal components stable during operation. CE glue or silicon based glue will be used to mount the magnets into their designated pockets.

6.5 HUB Housing

The Truck Smart Hub housing concept (shown in Figure 62) is designed to accommodate a set of twelve sensor LEDs (two per sensor), a system power switch, a button for syncing/resetting the system, two LED ports for system status notifications, and a port for the antenna to be placed. Once the electronics are fully integrated, size adjustments may need to be made to accommodate the PCB. A port for the power cable must also be created on the back cover panel which will be a separate design.

An opaque acrylic or plastic sheet will be fitted over the front face of the housing to cover areas where components would be shown. The chosen material will still allow light to penetrate through. This design will give the hub a styled-modern look that will still maintain functionality. The top LEDs will represent the forward-most sensors on the vehicle. The reasoning for placement and notification characteristics will be further discussed in the User Interface Design section.

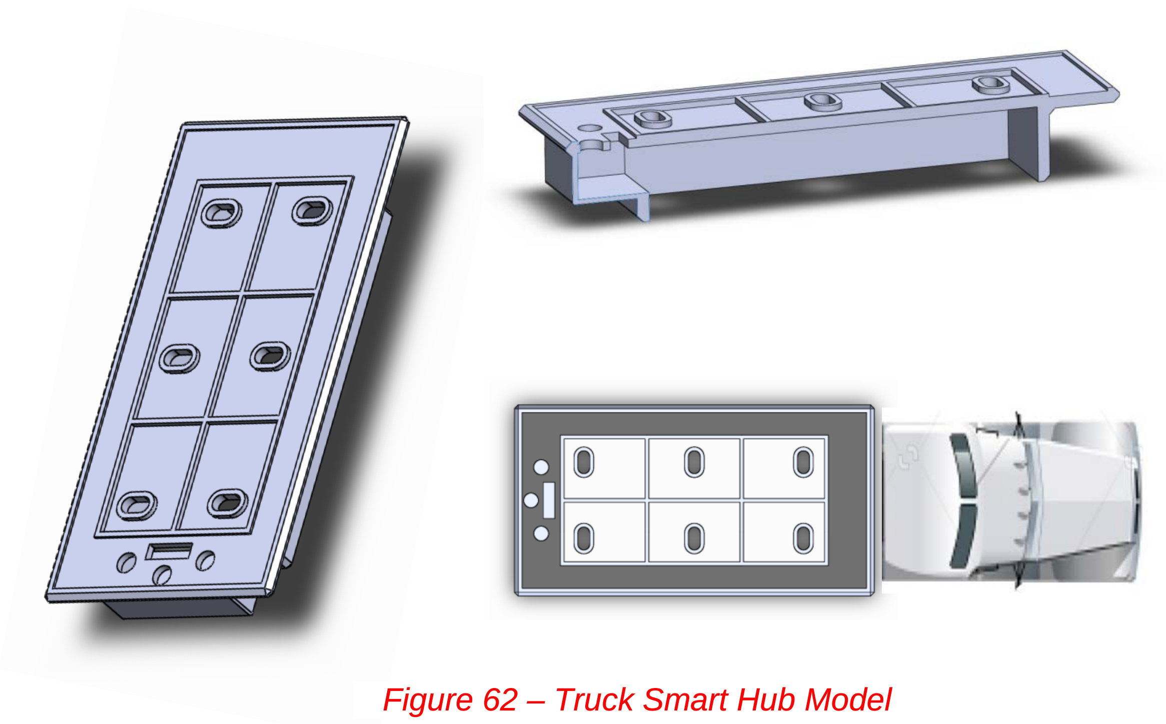


Figure 62 – Truck Smart Hub Model

6.6 User Interface Design

A commonly overlooked aspect of many electronics and software applications is the user interface (UI) design. The human brain is wired to understand simplistic designs and functionality. When the use of a product or tool causes mental stress for a user, the user will choose to use another product, or (if using a software tool) generate a poor quality end result.

There is a wide range of design criteria to abide by to ensure the user will enjoy interacting with the end product. According to usability.gov, the concept of user interface design “focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions”. For the Truck Smart system, there will be many pieces of the UI that need attention. The piece of the system that will have the most interaction will be the Truck Smart Hub. At any given time, the hub will be displaying a live feed of crucial information to the driver. If this information is incorrect or misunderstood, the end consequences could be disastrous. There will be two perspectives encountered during the UI design; 1) the setup, installation, and interaction with the system and 2) the display and interpretation of information.

The user interface design scheme has been accommodated to the general populous. Design criteria have been adapted to meet the necessary requirements for a statistical majority of drivers and is not intended for use for drivers at risk of seizure or who have been diagnosed with epilepsy. Due to the nature of the user interface’s output, there is a known risk for random and sporadic illumination. Drivers with this condition should consult with their physician prior to installing the Truck Smart system.

The Truck Smart system does not condone distracted driving and the end user should assume all liability during the system’s use. The interface has been designed with the concept of a warning notification to the driver before the system begins operation. In the event the system displays an error code or fault code as a result of a system malfunction, the Truck Smart system should not be relied on as a dependable means of a safety aid. Drivers assume full risk for every maneuver and should not solely rely on the Truck Smart Hub readout for blind spot detection.

During the installation of the Truck Smart system, all provided instructions should be followed accordingly to prevent system misuse and ensure the safety of the driver. As stated in the section pertaining to housing design, a critical aspect of the installation includes system placement less than, or equal to 35 degrees in order to keep the driver’s focus on the road.

7.0 System Testing and Demonstration

Once prototypes are made, the team intends to run multiple test runs to test performance of the system and tweak the design until accurate results are acquired. Testing will be done in two phases; phase 1 is to test the individual circuits on a breadboard before they are sent out for printing and phase 2 is to test the system as a whole using an actual truck in normal driving conditions.

The team contacted a trucker who has agreed to let them use a truck for phase 2 testing. For this testing, a total of six sensors are to be placed on the truck: one in the front, one in the back, and two on each side. Each sensor will have a part number; number 1 will be the one in the front and going clockwise, number 6 will be the one in the front left. Each one of these sensors will correspond to one LED on the hub unit placed in the cabin. Every time the sensor senses a vehicle in its proximity, the corresponding LED on the hub unit will light up. During the testing phase, these sensors will be placed at different locations around the truck to figure out which spot is the most effective for detection. The code may also need to be tweaked times to increase or decrease the sensing range. Once acceptable results are seen while the truck is stationary, the trucker - Mr. Reynold Marrero – will drive the team around for dynamic results. At this time a video of the system in working will also be taken for project demonstration. This video will show the LEDs turn on and off depending on whether there's another vehicle on that side of the truck or not. The ultrasonic sensors used in the system work by transmitting a signal out to the environment which bounces off of the nearest object and returns to the receiver part of the sensor. Based on the time it took for the signal to come back, it determines the distance of that object. An initial concern the team had was that the signal won't have enough time to return to the peripheral if the truck is moving too fast. However, based on the current calculations, the sensors will work at speeds up to 130 mph – which is more than the legally allowed speed of a truck. For the scope of this project, the sensor will be tested at speeds up to 80 mph, which is the highest speed allowed on all major Florida highways.

For the indoor demonstration of the system, the team has decided to put together a video of the system working on the road as well as create a model to replicate a truck trailer. This model will consist of a plywood sheet on 2-3 sets of wheels. Two sensors will be placed on each side of this sheet and will be wirelessly connected to the hub just like the actual system. The team plans to demonstrate the system by moving the plywood sheet around the room. Every time it "drives" by an object like a chair or table, the corresponding LED on the hub will turn on indicating the system has detected a "vehicle" on that side of the model truck. This model truck along with the video taken in an actual truck will present a good demonstration of the working system.

8.0 Administrative

This section contains the management portion of the project. It will first show and discuss the budget plans of the Truck Smart design. Next, it will examine the planned schedule of the project in the milestones section. This chapter will also compare the planned completion dates with the actual completion date.

8.1 Estimated Budget & Financing

The following table represents estimated values for the overall cost of the system. As the project's system requirements change throughout the course of its implementation, the overall cost will be adjusted accordingly.

Table 21 – Financial Plan

Item Description	Price / Unit	Amount	Estimated Price
Transmitter Microcontrollers (Uno)	$25.00	2	$50
Transmitter Microcontrollers	$5.00	8	$40
Receiver / Base Microcontroller	$50.00	1	$50
LCD/LED display for the Base	$30	1	$30
Sensors	$25	8	$200
Power Supply	$10	8	$80
System Housing	$35	8	$280
Printed Circuit Board	$30	8	$240
Wireless Transmitter	$40	8	$320
Wireless Receiver / Base	$40	1	$40

Total Cost (subject to change) = $1,330.00

**The project will be self – financed by the group members. **

8.2 Milestones

The milestones dates were set based on the corresponding task and with accordance with each of the team member's schedules and responsibilities. Most of the milestones for the first half of the implementation are based on research and familiarization with the system's components while functionality will be the focus of the second half during Sr. Design II.

Table 22 – Project Milestones

Milestone	Task	Start Date	End Date	Status	Lead
Senior Design 1					
	Administrative Outlook				
A	Concept Ideas & Project Selection	8/25/2016	9/6/2016	Complete	Group 32
B	Responsibility Assignments	9/6/2016	9/9/2016	Complete	Group 32
C	Initial Project Document	9/6/2016	9/9/2016	Complete	Group 32
D	Develop Table of Contents	9/6/2016	11/4/2016	Complete	Group 32
E	Draft Document	9/6/2016	11/11/2016	Complete	Group 32
F	Final Document	9/6/2016	12/6/2016	Complete	Group 32
	Technical Outlook				
1	Component Research & Gathering	9/6/2016	12/6/2016	Complete	Group 32
2	Circuit Design & Integration	10/1/2016	SD2	Researching	EE
3	User Interface Design	11/1/2016	SD2	Researching	CE
Senior Design 2					
	Administrative Outlook				
G	CDR Presentation	-	-	TBD	Group 32
H	Peer Review	-	-	TBD	Group 32
I	Conference Paper	-	-	TBD	Group 32
J	Mid-term Demo	-	-	TBD	Group 32
K	Final Presentation	-	-	TBD	Group 32
L	Web Exit Interview	-	-	TBD	Individual
	Technical Outlook				
4	User Interface Design (continued)	-	-	Researching	CE
5	PCB Layout Design	-	-	Researching	EE
6	Housing Design	-	-	Researching	CE/EE
7	Network Interfacing	-	-	Researching	CE
8	Display & Presentation Preparation	-	-	Researching	Group 32

9.0 Conclusion

Many engineering challenges were overcome during the research and development phase of the Truck Smart system. The system requirements that were shaped during the project's conceptual design stage have been taken into account for each and every component and line of code that has, and will be, integrated into the Truck Smart system.

Truck Smart's core infrastructure consists of many simple components that, when integrated properly, deliver a marketable function to an industry in need of an update in safety technology. Considerations have been taken in the design phase to make the end user's safety the number one priority for the Truck Smart system. Second hand factors, such as system testing, fault reporting, and even the cognitive impact on the user, have all been accounted for to ensure the end product is unrivaled to any similar technology of its kind.

The design workload has been evenly distributed among the computer engineering and electrical engineering disciplines. The Truck Smart system has been technically well-rounded to cover all bases. The integration of components and software will allow the Truck Smart design team to mesh interests, technical skills, and concept ideas. It is imperative that the team focuses not only on each technical detail, but how each detail impacts every other segment of the Truck Smart system.

One of the biggest accomplishments of Truck Smart's design is the successful integration of the up-and-coming ZigBee technology. Without this cutting-edge wireless protocol, the Truck Smart system would be much more complex and expensive to implement. This method of wireless communication is an ever-growing engineering masterpiece which gives conceptual ideas like Truck Smart an opportunity to propagate into feasible solutions for everyday problems.

10.0 Appendices

The final section consists of the appendix. This is made up of the references used throughout the paper. It will also show proof of emails sent for copyright permissions.

10.1 References

[1] 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks. (n.d.). Retrieved November 27, 2016, from https://standards.ieee.org/findstds/standard/802.15.4-2015.html

[2] Arduino - ArduinoToBreadboard. (n.d.). Retrieved October 19, 2016, from https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard

[3] Arduino - Ping. (n.d.). Retrieved October 25, 2016, from https://www.arduino.cc/en/Tutorial/Ping

[4] ATmega2560. (n.d.). Retrieved October 19, 2016, from http://www.atmel.com/devices/ATMEGA2560.aspx

[5] ATmega328 with Arduino Optiboot (Uno). (n.d.). Retrieved October 19, 2016, from https://www.sparkfun.com/products/10524

[6] ATmega328/P Datasheet. (2016, June). Retrieved October 18, 2016, from http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_datasheet.pdf

[7] ATmega328P. (n.d.). Retrieved October 18, 2016, from http://www.atmel.com/devices/ATMEGA328P.aspx

[8] Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V. (2014, February). Retrieved October 19, 2016, from http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[9] BCM2835. (n.d.). Retrieved October 25, 2016, from https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/README.md

[10] Bluetooth Technology: A Summary of its Advantages and Disadvantages. (n.d.). Retrieved November 23, 2016, from http://www.buzzle.com/articles/advantages-and-disadvantages-of-bluetooth-technology.html

[11] Bogs De Castro, TLE Secondary School Teacher I Follow. (2013). Basic concepts of wireless communication system. Retrieved October 17, 2016, from http://www.slideshare.net/skyrocker0004/basic-concepts-of-wireless-communication-system

[12] Different types of Antennas with Properties and thier Working. (2014). Retrieved October 17, 2016, from https://www.elprocus.com/different-types-of-antennas-with-properties-and-thier-working/

[13] Electronics Components: Introducing Microcontrollers - dummies. (n.d.). Retrieved October 25, 2016, from http://www.dummies.com/programming/electronics/components/electronics-components-introducing-microcontrollers/

[14] Gislason, D. (2008). ZigBee wireless networking. Oxford: Elsevier, Newnes.

[15] H. (2001). How Bats Work. Retrieved October 04, 2016, from http://animals.howstuffworks.com/mammals/bat2.htm

[16] H. (2013). Arduino's ATMega328 Power Consumption - Gadget Makers' Blog. Retrieved October 16, 2016, from http://gadgetmakersblog.com/power-consumption-arduinos-atmega328-microcontroller/

[17] Infrared vs. Ultrasonic - What You Should Know | Member Robot Tutorials. (n.d.). Retrieved October 11, 2016, from http://www.societyofrobots.com/member_tutorials/node/71

[18] Interstate Truck Driver's Guide to Hours of Service. (2015, March). Retrieved October 19, 2016, from https://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/Drivers_Guide_to_HOS_2015_508.pdf

[19] Introduction to RF & Wireless Communications Systems. (n.d.). Retrieved November 2, 2016, from http://www.ni.com/tutorial/3541/en/

[20] Jackson, P. (n.d.). Understanding PCB Manufacturing: Silk-Screening. Retrieved November 20, 2016, from http://www.omnicircuitboards.com/blog/bid/312861/Understanding-PCB-Manufacturing-Silk-Screening

[21] Kester, W. (1998). Practical design techniques for power and thermal management. Retrieved October 25, 2016.

[22] Kooijman, M. (2015). Building Wireless Sensor Networks Using Arduino. Packt Publishing, Limited.

[23] Managing in-vehicle distractions: Evidence from the psychological refractory period paradigm. (n.d.). Retrieved November 23, 2016, from http://dl.acm.org/citation.cfm?id=1969775

[24] Mar 22, 2013 Lou Frenzel | Electronic Design. (n.d.). What's The Difference Between IEEE 802.15.4 And ZigBee Wireless? Retrieved November 27, 2016, from http://electronicdesign.com/what-s-difference-between/what-s-difference-between-ieee-802154-and-zigbee-wireless

[25] MC13202 Datasheet with Addendum - NXP Semiconductors. (n.d.). Retrieved October 25, 2016, from http://www.nxp.com/files/rf_if/doc/data_sheet/MC13202.pdf

[26] Microcontroller with 4/8/16/32K Bytes In-System ... (2009, February). Retrieved October 19, 2016, from https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf

[27] Mitigation Strategies For Design Exceptions - Safety | Federal Highway Administration. (n.d.). Retrieved November 29, 2016, from http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.cfm

[28] Mohammad, T. (n.d.). Using Ultrasonic and Infrared Sensors for Distance Measurement. Retrieved October 10, 2016, from http://waset.org/publications/6833/using-ultrasonic-and-infrared-sensors-for-distance-measurement

[29] MSP430FG4618 (ACTIVE). (n.d.). Retrieved October 20, 2016, from http://www.ti.com/product/msp430fg4618

[30] MSP430G2553 (ACTIVE). (n.d.). Retrieved October 20, 2016, from http://www.ti.com/product/msp430g2553

[31] MSP432P401R (ACTIVE). (n.d.). Retrieved October 23, 2016, from http://www.ti.com/product/MSP432P401R

[32] NASA Drag Coefficient. (n.d.). Retrieved November 20, 2016, from https://www.grc.nasa.gov/www/k-12/airplane/dragco.html

[33] P. (n.d.). PAN802154HAR00 2.4 GHz Low Power Module for the IEEE802.15.4 Standard. Retrieved October 25, 2016, from http://datasheet.octopart.com/PAN802154HAR00-Panasonic-datasheet-50963.pdf

[34] Part 2: Discharging. (n.d.). Retrieved September 08, 2016, from http://www.itacanet.org/a-guide-to-lead-acid-battries/part-2-voltage-and-capacity/

[35] Power Supply Safety Standards, Agencies and Marks - cui.com. (n.d.). Retrieved November 12, 2016, from http://www.cui.com/catalog/resource/power-supply-safety-standards-agencies-and-marks.pdf&p=DevEx,5084.1

[36] Radar and Ultrasonic Sensors. (n.d.). Retrieved October 04, 2016, from https://www.apgsensors.com/about-us/blog/radar-and-ultrasonic-sensors

[37] RF Basics. (n.d.). Retrieved September 29, 2016, from https://www.digi.com/resources/standards-and-technologies/rfmodems/rf-basics

[38] School of Life Sciences | Ask A Biologist. (n.d.). Retrieved October 04, 2016, from http://askabiologist.asu.edu/echolocation

[39] Sensors: Ultrasound. (n.d.). Retrieved September 29, 2016, from http://www.sensorwiki.org/doku.php/sensors/ultrasound

[40] Simpson, C. (n.d.). Linear and Switching Voltage Regulator Fundamentals. Retrieved September 8, 2016, from http://www.ti.com/lit/an/snva558/snva558.pdf

[41] Standard for Tests for Flammability of Plastic Materials for Parts in Devices and Appliances. (n.d.). Retrieved December 03, 2016, from http://ulstandards.ul.com/standard/?id=94_6

[42] Technical Data - NXP Semiconductors. (n.d.). Retrieved October 25, 2016, from http://www.nxp.com/files/rf_if/doc/data_sheet/MC1321x.pdf

[43] Toth, E. R. (n.d.). Ten best practices of PCB design. Retrieved November 20, 2016, from http://www.edn.com/electronics-blogs/all-aboard-/4429390/Ten-best-practices-of-PCB-design

[44] Ultrasonic Distance Sensor. (n.d.). Retrieved September 29, 2016, from http://arduino-info.wikispaces.com/Ultrasonic%20Distance%20Sensor

[45] Ultrasonic Ranging Module HC - SR04. (n.d.). Retrieved October 26, 2016, from http://www.micropik.com/PDF/HCSR04.pdf

[46] Ultrasonic Sensing. (n.d.). Retrieved September 29, 2016, from http://www.ab.com/en/epub/catalogs/12772/6543185/12041221/12041229/print.html

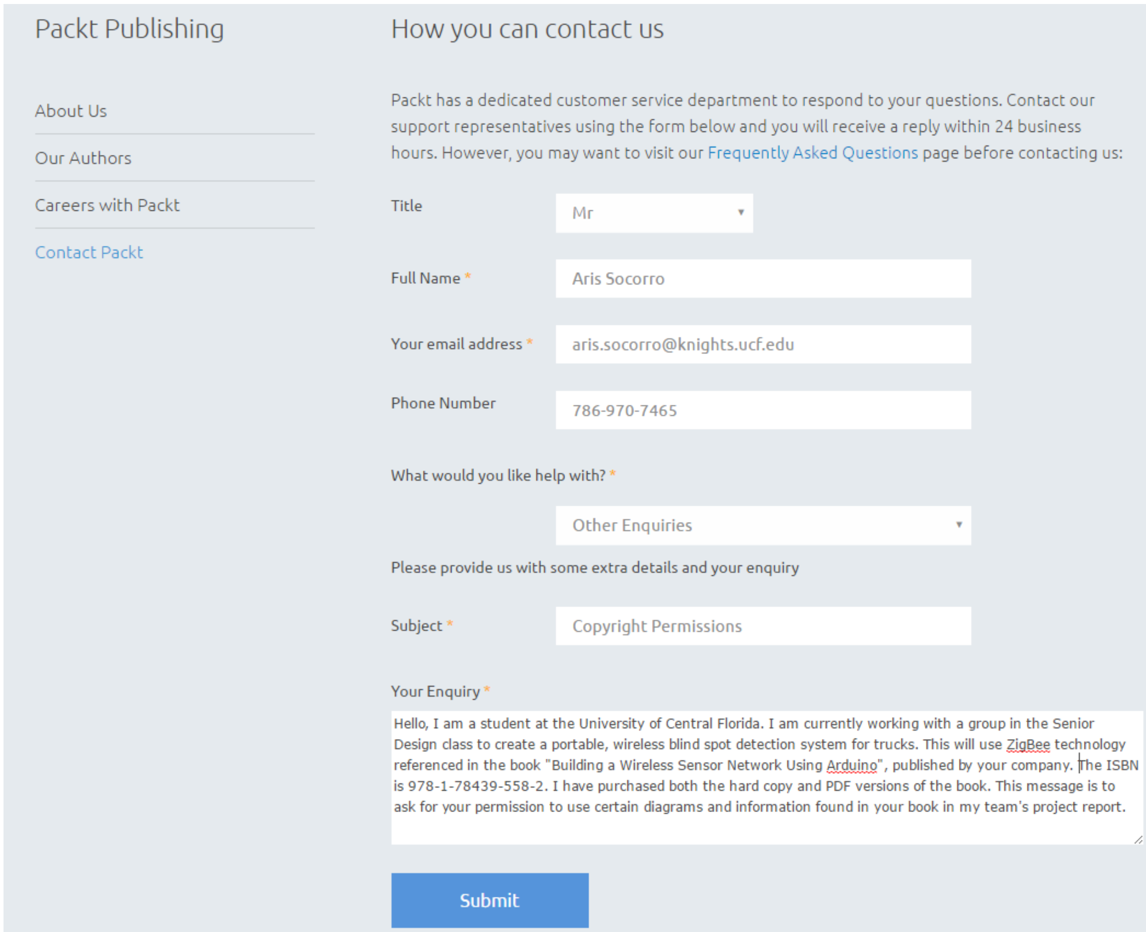
[47] Using EAGLE: Board Layout. (n.d.). Retrieved December 03, 2016, from https://learn.sparkfun.com/tutorials/using-eagle-board-layout

[48] Using EAGLE: Schematic. (n.d.). Retrieved December 03, 2016, from https://learn.sparkfun.com/tutorials/using-eagle-schematic

[49] What is an Arduino? (n.d.). Retrieved October 19, 2016, from https://learn.sparkfun.com/tutorials/what-is-an-arduino

[50] XB24CAWIT-001. (n.d.). Retrieved October 17, 2016, from http://www.digikey.com/product-detail/en/XB24CAWIT-001/602-1891-ND/6010100?WT.z_cid=ref_neda_dkc_buynow

10.2 Copyright Permissions



Packt Publishing

About Us

Our Authors

Careers with Packt

Contact Packt

How you can contact us

Packt has a dedicated customer service department to respond to your questions. Contact our support representatives using the form below and you will receive a reply within 24 business hours. However, you may want to visit our Frequently Asked Questions page before contacting us:

Title	Mr
Full Name *	Aris Socorro
Your email address *	aris.socorro@knights.ucf.edu
Phone Number	786-970-7465

What would you like help with? *

Other Enquiries

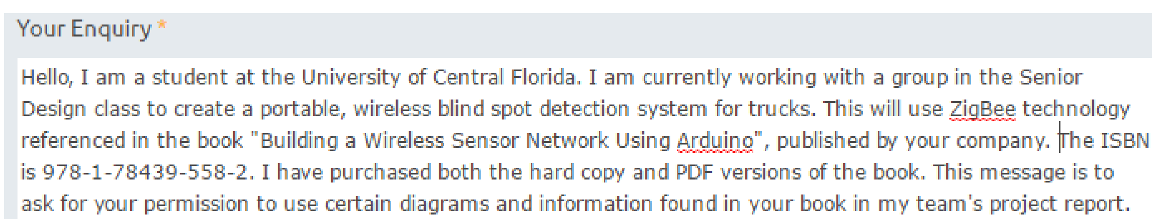
Please provide us with some extra details and your enquiry

Subject *	Copyright Permissions

Your Enquiry *

Hello, I am a student at the University of Central Florida. I am currently working with a group in the Senior Design class to create a portable, wireless blind spot detection system for trucks. This will use ZigBee technology referenced in the book "Building a Wireless Sensor Network Using Arduino", published by your company. The ISBN is 978-1-78439-558-2. I have purchased both the hard copy and PDF versions of the book. This message is to ask for your permission to use certain diagrams and information found in your book in my team's project report.

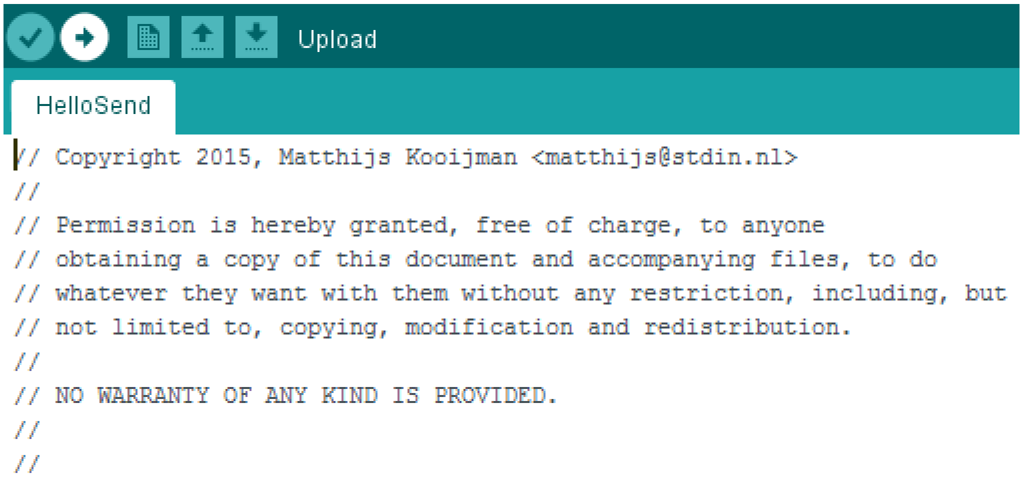
Submit



Your Enquiry *

Hello, I am a student at the University of Central Florida. I am currently working with a group in the Senior Design class to create a portable, wireless blind spot detection system for trucks. This will use ZigBee technology referenced in the book "Building a Wireless Sensor Network Using Arduino", published by your company. The ISBN is 978-1-78439-558-2. I have purchased both the hard copy and PDF versions of the book. This message is to ask for your permission to use certain diagrams and information found in your book in my team's project report.

Figure 63 – Packt Publishing Permission Request




```
// Copyright 2015, Matthijs Kooijman <matthijs@stdin.nl>
//
// Permission is hereby granted, free of charge, to anyone
// obtaining a copy of this document and accompanying files, to do
// whatever they want with them without any restriction, including, but
// not limited to, copying, modification and redistribution.
//
// NO WARRANTY OF ANY KIND IS PROVIDED.
//
//
```

Figure 64 – Arduino Code Permissions

The book used to implement the wireless transmission of the system comes with a set of reference software to help understand its concepts. The following header file is included in all programs written by the author. It authorizes any individual to use the material for any purpose without restrictions.

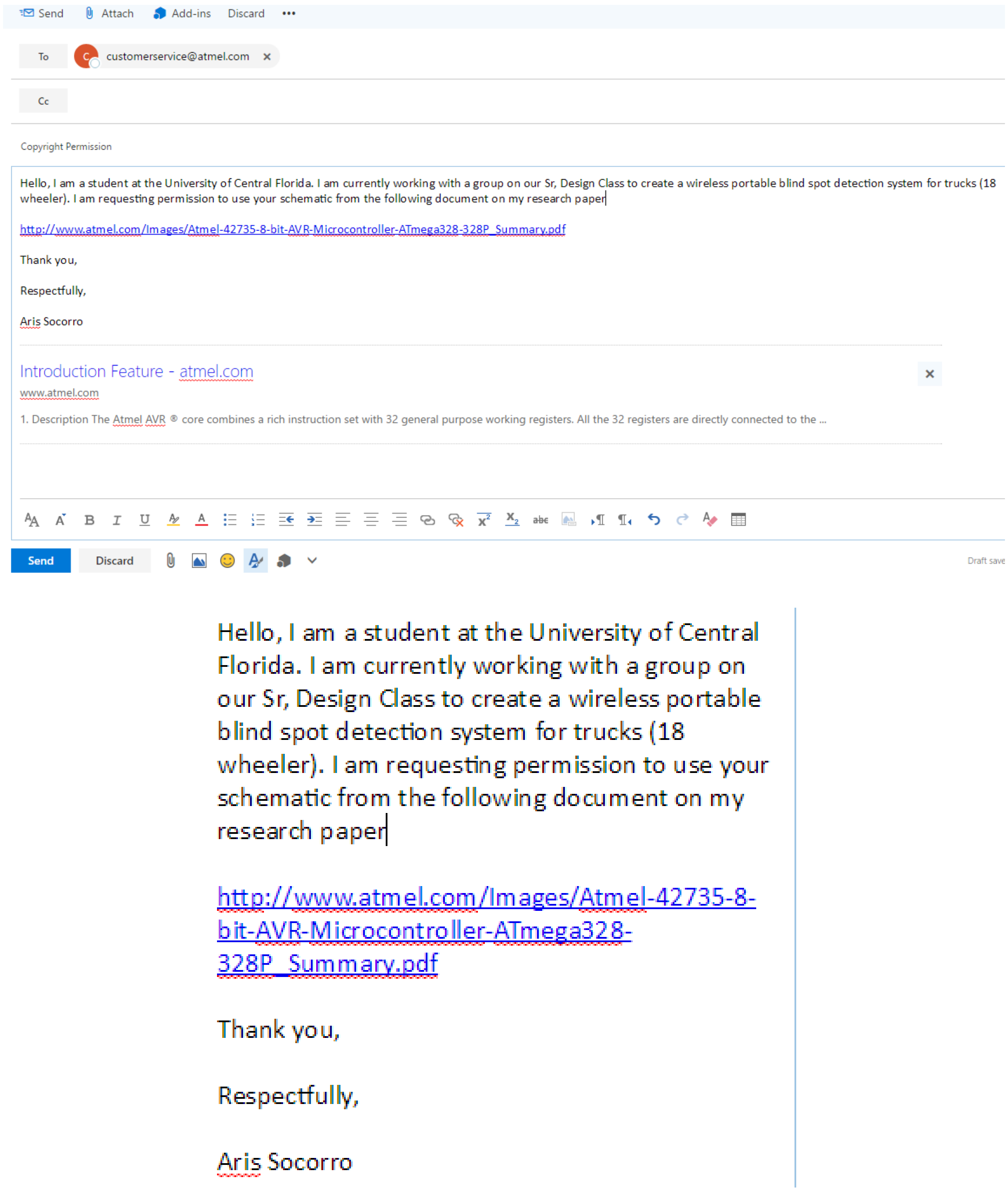
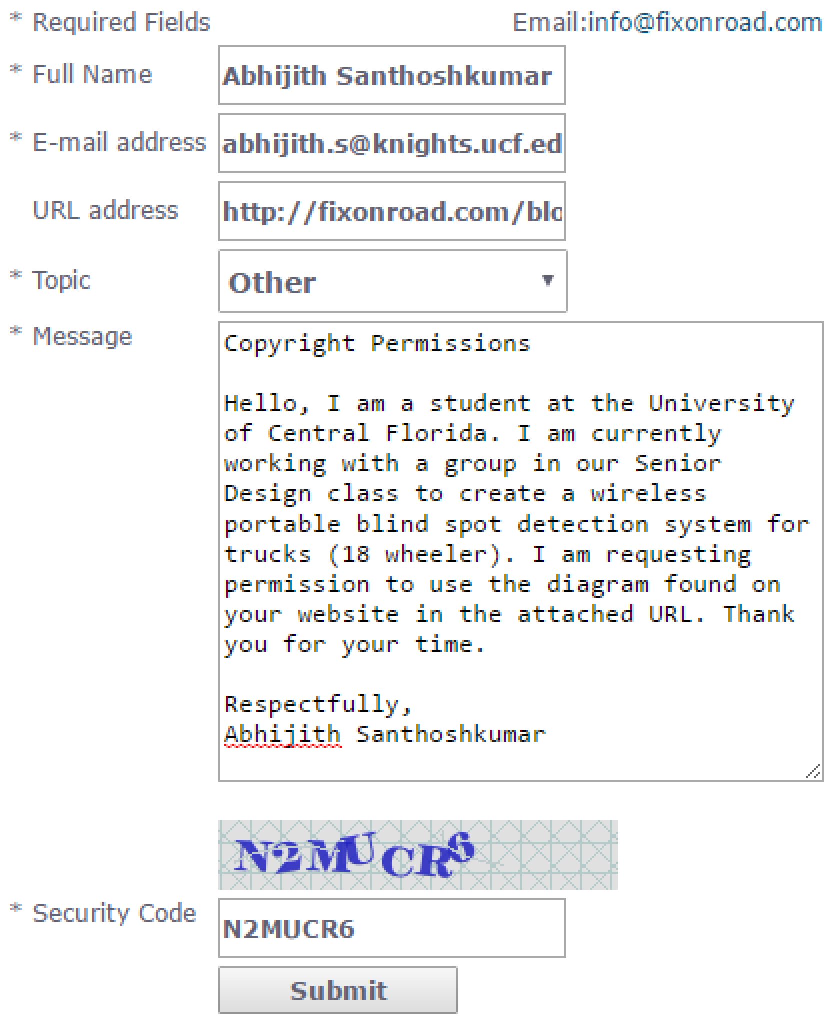


Figure 65 – Schematic Copyright Permission Request



* Required Fields Email:info@fixonroad.com

* Full Name: Abhijith Santhoshkumar

* E-mail address: abhijith.s@knights.ucf.ed

URL address: http://fixonroad.com/blo

* Topic: Other

* Message:

Copyright Permissions

Hello, I am a student at the University of Central Florida. I am currently working with a group in our Senior Design class to create a wireless portable blind spot detection system for trucks (18 wheeler). I am requesting permission to use the diagram found on your website in the attached URL. Thank you for your time.

Respectfully,
Abhijith Santhoshkumar

N2MUCR6

* Security Code: N2MUCR6

Submit

Figure 66 – Truck Diagram Permission Request

10.3 Datasheet

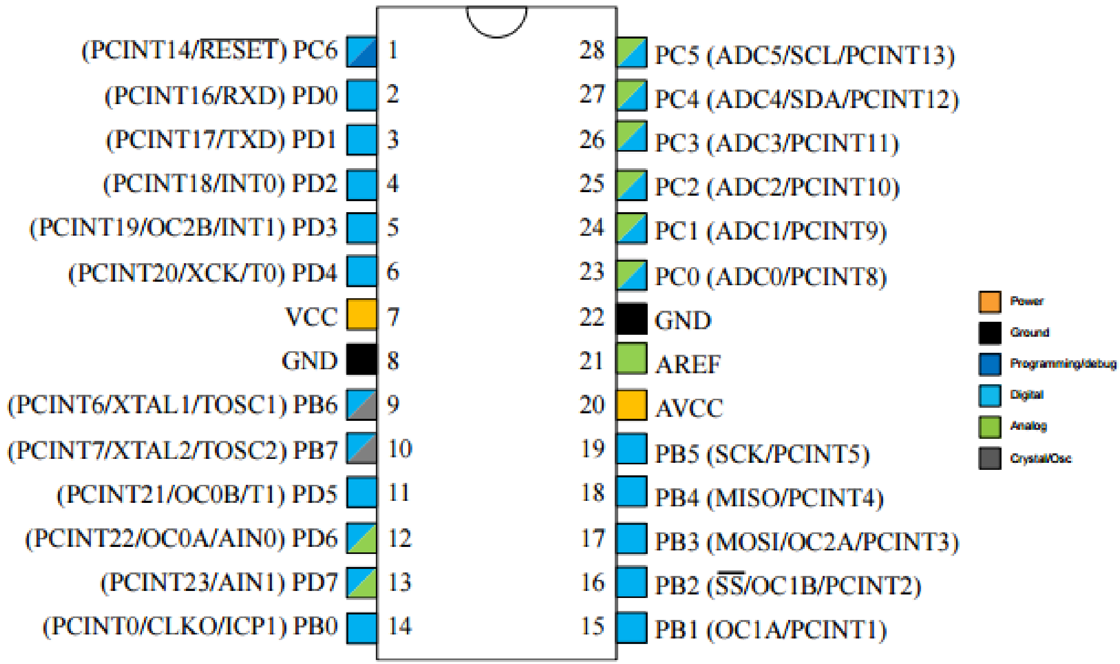


Figure 67 – Atmega328P Pin Out Diagram (used with permission from Atmel)