Situation-Aware Stop Signal

Senior Design 2
Spring 2019

Group 3

Jonathan Ling - Electrical Engineering
Annabelle Phinney - Electrical Engineering
Trent Sellers - Computer Engineering
Joseph Walters - Computer Engineering

University of Central Florida
College of Engineering and Computer Science

Table of Contents

List of Figures

List of Tables

List of Formulas

1 Executive Summary

In a world that is increasingly populated by drivers and dominated by distractions, there is an ever-greater need for safety to be maintained on the road. While attention has been turned to larger intersections, many accidents occur at the smaller crossroads of our community. Where conventional stop signs are insufficient, the **Situation-Aware Stop Signal** is intended to draw greater attention to the roadway and be a catalyst for safer crossings at small intersections. Influenced by technology found in existing traffic lights and autonomous cars, this intelligent system is designed to be a modern and more effective replacement for the stop sign. Using advanced sensors and real-time computing, the Situation-Aware Stop Signal is able to monitor the intersection for traffic and provide intuitive warnings to drivers.

The device is intended to be placed at an elevated height in the middle of the intersection. This perspective allows it to “see” in multiple directions and present information to drivers that are at different sides of the intersection. When drivers are detected as traveling toward the intersection, the stop signal lights up to visually alert them to slow down and to come to a complete stop. It enforces this stop by holding them stationary for a short time, at which point it releases them in non-conflicting order with another lighting queue. When non-halting cross-traffic is detected, the signal presents a uniform lighting scheme across all lights to keep traffic stopped on all sides of the intersection.

RADAR and LiDAR were carefully selected as two of the most-accurate and reliable sensor types for this application. As LiDAR was discovered to yield more-precise data at a closer range and as RADAR was found to yield less-precise data at a further range, it was decided to use the two jointly to create an accurate and power-efficient system. Low power consumption was an important factor in the selection of components and allows the Situation-Aware Stop Signal to be self-sustaining. Efficient solar technology was considered and chosen to allow generation of power on the device with a power system included to store energy for all-day and all-night operation.

Care was also taken in selecting an appropriate platform on which software could operate. A real-time operating system (RTOS) was chosen due to the need for quick response to environmental stimuli and detected events. Preemptive scheduling inherent of this kind of operating system allows lower-priority processes to be preempted in favor of higher-priority processes in the case of a critical event. In the terms of this project, this means the Situation-Aware Stop Signal is able to begin processing the moment it detects a vehicle and anticipate threats before accidents can occur.

This system is not without its challenges, and particular attention is needed to make it one that can be economically and legally feasible. Cost is likely a culprit as to why a similar product is not seen in our neighborhoods. National, state, and city governments also have laws that dictate the size, shape, placement, and use of roadway signs, as in the Manual for Uniform Traffic Control Devices (MUTCD) of the U.S. Department of Transportation. Our intention was to produce a best-attempt product that acknowledges these constraints and addresses them sufficiently to be a real candidate for ensuring a safer community.

2 Project Description

The Situation-Aware Stop Signal was conceived out of a concern for safety in the community. Originally proposed by group members in a brainstorming session centered on common interests and passions, it is a solution for a problem that is widely recognized but seldom addressed. More so, it presents a unique approach to solving this problem that could only be possible in the modern day. Not unlike the crossroads where it may be located, the stop signal is the intersection between sophisticated hardware and software. As will be explained, these advances allow it to play an intelligent and important role in the coming and going of local area traffic unlike any traffic control device that has existed prior.

The purpose of this section of the document is to give detailed explanation about the project and the scope that it covers. An account of the motivation behind the Situation-Aware Stop Signal will be given, as well as an explanation of its primary objectives. This will be followed by an overview of the requirements that define its functionality, quality, and performance. A review of this section will give the reader a better understanding of the purpose and significance of the stop signal, as well as introduce him to the terminology that is specific to the project.

2.1 Project Motivation

At the crossroads of one small neighborhood in Orlando lies a small memorial at the corner of a four-way stop. It stands in remembrance of someone's loved one – a neighbor – who lost their life in a fatal accident in that very same place. Every year, in our community and in others, there are a multitude of accidents that occur at small, residential intersections. These are often the result of drivers running stop signs due to visual distraction, the influence of alcohol, and other forms of carelessness. Stop signs have long been important tools for controlling traffic at small intersections, but they are becoming less effective in an age when both traffic and distractions are more present than ever on the road.

Since the first stop sign was installed in Detroit in the year 1915, the approach for managing traffic at small intersections has not changed by a great amount. The need for change, however, has never been greater. Living in an age dominated by small, bright screens and various forms of handheld or otherwise easily-accessible entertainment, there is more potential for accidents to occur today than in any other time in history. These new distractions introduce complex issues into the realm of roadway safety and pose a serious threat to drivers if not mitigated.

In attempts to draw more attention to small intersections, smaller stop signs have been replaced with larger ones. In at least one place in Orlando, two stop signs were placed on a single post in a desperate effort to get drivers to stop. Such efforts are clear indicators of a need for more effective traffic control. While we cannot directly change the way drivers behave, surely there is a better way to capture their attention and to protect them from those who choose to drive unsafely.

We believe that advances in technology, now made more affordable through manufacturing improvements, present an opportunity to revolutionize the way we advise, warn, and alert drivers on the small roads of our community. This opportunity and the promising ability it presents to keep others safe are the greatest sources of motivation for us as a team. We desire our work to be meaningful and to showcase the potential we have to make our community a safer place. This means making a significant difference in the lives of the drivers that cross our roads – our families, our friends, and our neighbors.

2.2 Objectives

Our mission was to produce a device with the potential to be implemented on a large scale, to have a daily impact on driver safety and, ultimately, to *save lives*. In order to achieve this ambitious mission, work for this project was focused in the following areas:

1. Accuracy
2. Self-sustainment
3. Efficiency
4. Low cost

Due to its safety-critical nature, it was necessary for the product to be accurate as well as self-sustaining. To make itself a candidate for real-world application, the Situation-Aware Stop Signal was designed to use solar power as its primary source of energy. This feat represents a departure from existing traffic light infrastructure and positions the stop signal as an independent yet fitting addition to intersections. By emphasizing sustainability and low-cost components, we intended to make a device that is appealing to companies and governments as an affordable solution for both private and public roadways.

Currently, the market targeted by this product is dominated by the very low-cost stop sign, a traffic control device that can be found for under $50 in its most basic form. Although it brings with it a low price, the stop sign is a primitive product that has changed very little in the last 100 years. With the advances of today's technology, we believed that we could design an affordable alternative with the features needed to ensure a safer community. Although we acknowledged that our device would cost more than a stop sign, we believed that we would be able to produce it at a cost low enough such that we could justify it plainly with its greater capability, quality, and maintainability.

To set the Situation-Aware Stop Signal apart from existing traffic control devices at small intersections, we considered the problems existent at these intersections and determined three needs. These needs we consider the primary objectives for the technical aspect of this project. They are:

1. Preventing the threat of drivers not stopping safely before the intersection
2. Protecting drivers from others who may *not* be stopping safely at the intersection
3. Scheduling traffic when cars arrive at the intersection near the same time

2.2.1 Prevention of Threat

The first objective was representative of a need to prevent threats to safety at the intersection. A threat exists when a driver does not slow down as he or she approaches the intersection. This may be due to some form of a distraction, such as the use of smartphones while driving, or could be the result of driving under the influence of alcohol or other impairing drugs. For this situation, the stop signal was designed to alert the driver to slow down and to come to a complete and safe stop.

A key to meeting this objective was to capture the attention of the driver. By triggering the driver's senses, particularly sight, the stop signal is able to help him or her maintain his or her eyes on the road ahead. Implementing this required a need to understand human psychology and the behavior people tend to exhibit while driving. Throughout our study, we observed that people seem to respect traffic lights more than stop signs. This observation was fundamental in exploring our method of redirecting the driver's attention.

2.2.2 Protection from Threat

The second objective was representative of a need to acknowledge threats to safety at the intersection and to protect drivers from them. Should a driver constitute a threat and not slow down as he or she is entering the intersection, it is of critical importance to alert all other drivers at the intersection and keep them immobilized until the threat has passed. In such a situation, the traffic control system must attempt to keep *everyone* safe. This situation is given the highest priority amongst all other situations that may occur at the intersection.

To meet this objective, the stop signal had to be able to detect and classify a threat well before it reached the intersection. Here, a study of car velocity, driver reaction time, and safe stopping distance was required to properly identify danger. These factors were considered in the interpretation of sensor data and the primary function of the algorithms that were written in the project software. This ability to process data was directly dependent on the speed at which it could be captured. This second consideration, a need for real-time situational awareness, was the principal consideration of the embedded software design.

2.2.3 Scheduling of Traffic

The third objective was representative of a need to perform minor scheduling of traffic when conflicts are presented at the intersection. Oftentimes, drivers stop at the intersection at the same time and do not proceed for concern of incorrectly assuming or yielding the right of way. When this occurs, not only is traffic flow constrained, but a new threat to safety may be presented. By implementing basic scheduling with the stop signal, we took an opportunity to correct this issue.

We acknowledged current traffic laws and designed our system to direct traffic in a manner that is consistent with the behavior they dictate. We aimed to recognize all combinations of driver presence at the intersection and understand the order by which traffic should proceed in every unique case. A study of driver behavior and arrival times at small intersections was necessary to sufficiently consider timing when scheduling traffic. Although this was likely to vary greatly between drivers and locations, an average or more probable pattern of traffic flow and management was defined.

2.2.4 Summary

The objectives presented represent three distinct needs for safety to be better ensured at small intersections. Through careful consideration of the problem at hand, we recognized that we should both prevent threats and protect drivers. A focus on one but not the other would constitute an incomplete solution to the problem we desired to solve. Just as well, we recognized a need to schedule traffic and mitigate issues that may be introduced by drivers' misunderstanding. By designing a solution considerate of each of these needs, we believed there to be a higher probability of success in maintaining safer roadways.

These principle objectives guided our approach to this significant task and were frequently considered in our decision-making as a team. Rightfully so, they embody the purpose of the project and all work that was done. Summarizing such, they have been foundational in the drafting of requirements that define the Situation-Aware Stop Signal and its operation. These requirements, found at the continuation of this section, are the basis for the design proposed in this document.

2.3 Requirements Specification

The below requirements constitute all major considerations made for the functional and non-functional aspects of the Situation-Aware Stop Signal.

Design Requirements

- The signal shall be one centralized unit.
- The device shall be visible during the day.
- The device shall be visible at night time.

Operational Requirements

- The signal shall detect an oncoming vehicle within 20 meters.
- The signal shall be operable 24/7.
- The device shall signal drivers to stop safely before the intersection.
- The device shall signal all drivers to stop when a threat is detected approaching the intersection.
- The device shall prevent conflicts when vehicles arrive at the same time at the intersection.
- The device shall be responsive in real-time operation.
- The device shall normally run autonomously.

Power Requirements

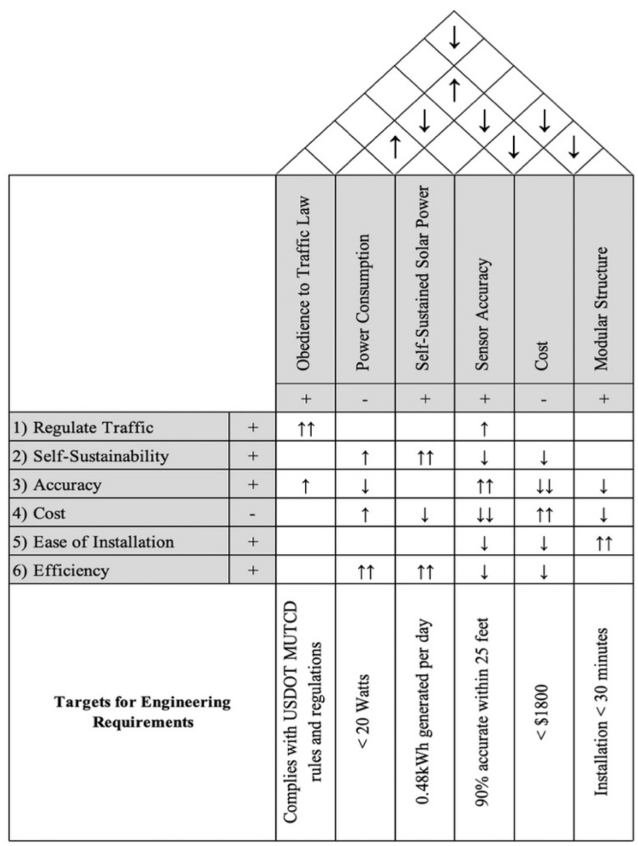
- Solar charge controller shall output 12V.
- Solar panel shall output greater 6V.

- Battery shall hold enough power for a minimum of one day of operation.
- Solar panel shall output greater than 40W.
- Solar panel shall charge the battery while device is in operation.

Safety Requirements
- The signal shall abide by road sign laws specified in the Manual for Uniform Traffic Control Devices (MUTCD).
- The signal shall detect vehicles that are traveling up to 13 m/s.

2.4 Marketing and Engineering Requirements

For the Situation-Aware Stop Signal, marketing and engineering requirements were compared and contrasted to provide additional insight for product design. This analysis is represented by the House of Quality depicted in Figure 1 on the next page.



House of Quality Legend	
Strong Positive Correlation	↑↑
Weak Positive Correlation	↑
Weak Negative Correlation	↓
Strong Negative Correlation	↓↓
Positive Impact	+
Negative Impact	-

Figure 1: House of Quality

3 Standards and Constraints

In order to create both a functional and marketable device that could potentially be used to control traffic in the future, certain standards or requirements had to be met. These standards provide specifications regarding how traffic products must be created or implemented. While most of the relevant standards included electrical, mechanical, and software engineering design, this project also required us to abide by environmental, safety, and traffic standards.

3.1 Legal Standards

Traffic standards usually vary depending upon the country, the traffic patterns, and the population of the area. In the state of Florida there are rules that must be followed when designing and implementing a traffic signal for use. To understand these regulations, it is important to note that one of the major goals of the Situation-Aware Stop Signal is to replace stop signs on private properties or areas that have low to moderate traffic flow. Because this device incorporated items that are not placed on or in a typical traffic light, there are many standards that aren't relevant and others that cannot be abided by completely.

After the first traffic signal was developed in Salt Lake City, Utah around 1912, the first traffic signal to include a yellow light wasn't established until 1920. Since then, the yellow light has been an important aspect of the traffic signal. Without it, both car accidents and automotive related deaths would rapidly increase. As a result of the yellow light's high importance, there have been standards created specifically stating that a functioning traffic signal must include a yellow light for an amount of time between the red and green lights. While this yellow light time often depends on the speed limit and the area's traffic density, the average yellow light time is 3.7 seconds for a road with the speed limit of 30 miles per hour. Below in Formula 1 is the standard equation used to calculate the yellow light time. This formula depends on the current speed of the car (Vg), the type of road the traffic signal is placed on (grade or Gr), along with the driver's perception or reaction time (Tpr). These types of calculations are not only necessary for determining the yellow light time, but also the time of the total traffic cycle or how long it takes the signal (or group of signals functioning in the same intersection) to go through all three light colors (i.e. red then green then yellow then red again).

$$\mathrm{Y(v)} = T + \frac{V_g}{2D_r + 2gG_r}$$

Formula 1: Yellow Light Traffic Time

3.1.1 Traffic Color Standards

In general, all traffic control signals must be limited to the colors red, green and yellow. These colors all have the same meaning across the United States. According to the 2018 Florida Statutes, a solid green light signifies that traffic may proceed with caution whether turning right or proceeding straight. Vehicles facing a green light must also yield to pedestrians and those turning left must yield to currently moving oncoming traffic. A solid yellow light is a warning to oncoming traffic that a red light will be coming soon so stop if possible. A red light indicates that any oncoming vehicles should come to a complete stop at the white line until a green light indication is shown.

There are different meanings for a steady light and a flashing light. For example, if a traffic light is either damaged or not in operation yet, the lights on the signal will flash at a steady speed. When all lights on the signal are flashing, this may indicate the stop light is damaged. Each color when flashing separately, means something different. A flashing red light indicates that the oncoming vehicles should treat the intersection as a stop sign. All oncoming vehicles will stop at the indicated white line or before the intersection until the route is clear of other traffic or pedestrians. A flashing yellow light means oncoming traffic should proceed with caution and make sure to watch out for traffic and pedestrians. A flashing green light, however, is not used in the United States and has no meaning. This signal will not be seen on functioning traffic signals nor will be used in the project.

3.1.2 Traffic Signal Placement Standards

In addition to the signal colors and symbols, the placement of the lights on the traffic signal are also regulated and as a result, all three colors must be in a certain order. The Manual on Traffic Control devices specifies in Part 5, that the red signal must always be first, followed by the yellow, and the green signal shall be last. This rule also applies for traffic signals with multiple green or yellow lights on the same signal. For example, if there are two green lights (i.e. a green arrow and a solid green circle), the yellow light must be before both green lights and both green lights must be adjacent to each other. Also, in this situation the solid green circle must be placed on top or to the left of the green arrow. This case applies for all colors.

While the meaning of each color on a traffic signal remains constant within the United States, the amount of lights and what type of lights on a traffic signal may vary depending on the location of the traffic light. For the purposes of this project, the type of area that will be considered is a low-volume traffic area. According to the Federal Highway Administration, a low-volume area lies outside of built up cities and towns. This area cannot be a freeway, expressway, a highway, or a residential street in a neighborhood. This road classification is important because it determines certain standards and whether it is necessary to install a traffic signal or a street sign. Along with these standards, another factor that helps to determine the road's volume is the AADT. The AADT is the average annual daily traffic (measured in number of cars that pass a specific point in the road divided by 365 days). This value helps determine how busy the road is and what type of traffic control systems should be put in place. For example, the maximum AADT value for a low traffic road is 400 AADT. This is the type of road that was considered for this project.

Because low-volume roads are often controlled just by road signs alone, there tend to be more accidents and careless drivers that do not follow traffic rules. Traffic signals have been proven to help limit these incidents by attracting drivers' attention with the use of bright lights and different colors. This allows every driver to understand what he should do when approaching a traffic signal. Unlike signs, traffic signals give the drivers a specific instruction. As mentioned previously, each color has a specific meaning and is known to every driver. Additionally, traffic lights provide organized movement of traffic and allow more cars in the intersection. Traffic signals can also be modified if the area becomes higher or lower volume and can be repaired if damaged. More importantly, traffic signals have decreased the amount of crashes and automotive related deaths as well as speed up the flow of traffic. While these devices have greatly improved traffic flow, there is always more that can be done to improve this type of device.

Even though it is obvious that traffic signals significantly improve the flow of traffic, they are not always necessary and can reach a point where they become a hindrance rather than an aid, especially in low volume areas. For example, in extremely low volume areas, an unnecessary amount of traffic lights may create a large buildup of traffic on one road because of a long red light. Excessive traffic signals can result in disobedience of the lights by drivers and force drivers to take alternative routes. These disadvantages can also result in more accidents. Traffic control signals that are placed improperly or poorly designed can also have a negative effect on traffic. For example, traffic lights placed behind bushes or at a non-visible angle can put drivers, as well as pedestrians in danger.

These drawbacks of implementing traffic signals have caused less desire for further installation of traffic signals. As a result, the Situational Aware Stop Signal will successfully avoid these flaws that have been put in traffic systems previously. One of the major flaws mentioned above is the improper installation of traffic signals. In order to avoid this mistake certain standards regarding height and distance from the intersection are essential in implementing a traffic light.

3.1.3 Traffic Signal Distance and Speed Requirements

The Florida Department of Transportation requires a minimum and maximum distance that a traffic signal must be from the stop line. For example, in the case of a hanging light, the signal face must be at least 40 feet from the stop line or 50 feet from the beginning of the intersection (in the case where there is not a stop line). The maximum distance depends on the type of lights used. Indicated in Figure 4D-4 in the Manual for Uniform Traffic Control Devices, the standard for the 12-inch lights presents a maximum of 180 feet from stop line to signal face. The standard for 8-inch signal indications is 120 feet from signal face to stop line. An additional 10 feet must be added to these measurements in cases where there is not a stop line, which is more likely in low-volume roads.

In addition to these distance requirements, it is important to know that the speed limit of the traffic light area also has an effect on the light placement. For example, according to Figure 4D-2 in the Manual for Uniform Traffic Control Devices, if the speed limit of the road is 25 miles per hour, the minimum distance for the driver to be able to see the stop light must be 215 feet. If the traffic signal is not placed within these standard dimensions, the driver might not be able to see the light in time to stop. As the speed limit of the road increases, the minimum sight distance also increases, meaning that the driver needs to be able to see the signal from further away in order to stop before the intersection. The relationship between the speed and distance is shown in Figure 2 below.

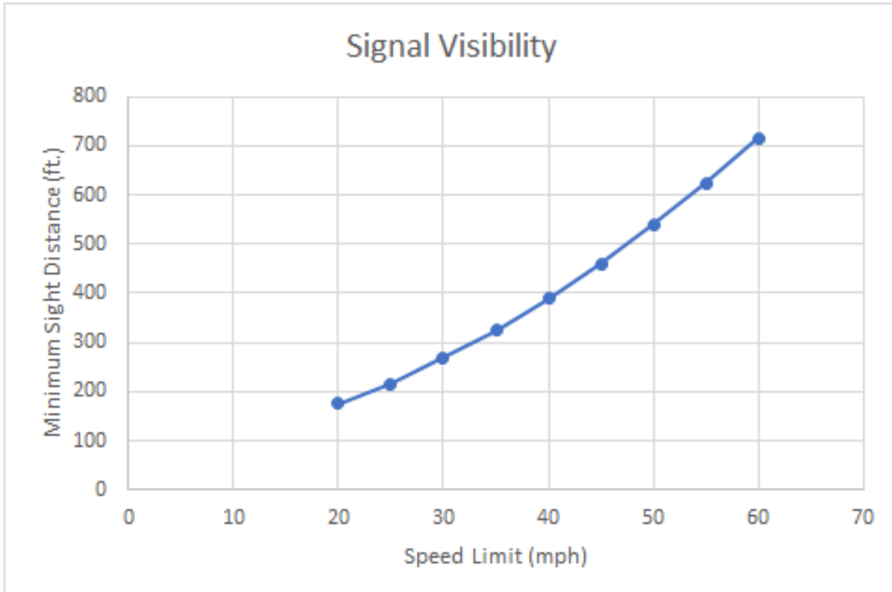


Figure 2: Signal Visibility Based on Stopping Distance

Because signal visibility is so important on every traffic signal, there are additional standards created and tested by the ITE. While most traffic lights have not been updated, some newer signals have incorporated LEDs into the light signal due to their efficiency, these standards have been developed and will applicable for this project.

3.1.4 Traffic Lighting Standards

The intensity of each light depends on the light's size. As stated above, the only allowable light sizes are 12 inch and 8-inch lights. This is the same for LEDs. The brightness or light intensity of LEDs are measured in candela or cd. When using a 12-inch LED, the minimum light intensities must be as follows: red will be 365 cd, yellow will be 910 cd, and green will be 475 cd. When using an 8-inch LED, the light intensities will be as follows: red will be 165cd, yellow will be 410 and green will be 215cd. It is important to note that due to the yellow LED's large luminosity requirement (almost double that of red), it must be 1.7 times brighter than the signals with non-LED lights.

In addition to the brightness of the lights, in the case of a blinking signal, each light on the signal must flash at the same rate and at the same time. Alternating flashes can confuse and distract drivers and can have a negative effect on the flow of traffic. To prevent this becoming an issue, it is required that the blinking of lights must flash continuously at a rate no less than 50 seconds and no more than 60 times per minute.

Traffic signals and power lines lie on a grid. Each light cycle is run on a timer and runs according to those signals around it. This is so no two opposing traffic signals display green, causing a collision. Over time, with the addition of red light cameras and ground sensors to adjust the light cycle according to the volume of traffic at that time, traffic control has slowly improved, but have not experienced any major breakthroughs and have been proven to both not always be efficient and to not work properly. Even though traffic can be sensed, little is immediately altered when a road gets too congested or is completely empty. Because there is little information and little real-world applications of more in depth traffic control, there are limited standards on actively controlling signals.

3.1.5 Traffic Control Standards

Traffic control standards make traffic signals smarter and in turn allow drivers to reach their destination as quickly and as happily as possible. In order to successfully do this, the traffic cycle will give the right of way to those that gotten to the signal first or those sides of the intersection with the most traffic. The Manual on Uniform Traffic Control Devices states that traffic signals should determine who has the right of way by using either preemption control or priority control.

Preemption control allows the traffic cycle to transfer from its normal operations and cycle time period to a special control mode. This mode given certain vehicles, such as trains, boats, and emergency vehicles the right of way at an intersection. This is helpful because these vehicles are typically large and will take up two or three car lengths in an intersection. Allowing these vehicles to go as soon as possible is a great way to clear up an intersection and allow traffic to run more quickly.

Priority Control is a way traffic signals determine who in the intersection will get the right of way. Instead of focusing on larger vehicles, priority control gives those ‘light-rail transit’ vehicles priority in an intersection. These kinds of vehicles include those who are not traveling a long distance, such as school or transit buses. This control method allows those who only need to travel a few miles get off of the intersection or road faster and allow those who are traveling farther distances spend less time waiting in traffic.

While these control methods will work well on large volume roads, they may be unnecessary for the types of roads this project is focused on. This is mainly because there are limited amount of large vehicles such as boats and buses on less traveled roads. The Manual on Uniform Traffic Control brings up a possible solution to traffic control on low volume roads, that will not create an unnecessary use of resources. The manual states that creating traffic devices that could provide a more detailed guidance or warning of unsafe movements or unusual conditions on the intersection. Creating an intelligent device like the one suggested in the manual is one of the major goals of this project.

Even though the standards might limit the creativity and design of traffic signals, there is still room to solve the issues of and improve upon what is out on the road. Providing drivers with more information and letting them be aware of those who aren’t practicing safe driving could not only improve traffic flow, but it could also prevent car crashes and even save lives.

3.1.6 Hardware Standards

There are certain standards that must be upheld by the Situational Aware Stop Signal and those designing it in order for this project to be applicable in an intersection and for it to work successfully. In addition to the traffic standards previously mentioned, there are requirements for all of the hardware this project requires along with how it can be used.

While we are only dealing with low power electronics, it is still vital that the power supply and delivery standards be followed correctly. These efficiency standards for power supplies provide tests for the worst-case scenario of the circuit and require certain minimum and maximum power outputs in any scenario. For example, the IEC standard AS/NZS 4665 Parts 1 and 2 describe in detail what the power requirements are for each system and other power standards to abide by such as the power required at the input and at no load. These standards were followed in the device, which helped to improve power efficiency.

Placement of the hardware both in and outside of the device must also follow the appropriate standards. With products such as ours that have electronics and hardware in an enclosure. There are certain space requirements for electrical components to prevent risk of too much heat or an electrical fire. For basic plastic encasing that will be used for the Situational Aware Stop Signal, there must be a minimum 2 mm clearance between our hardware and the plastic encasing. This clearance can be a type of insulating material or an air gap. In the case of this project, an air gap will serve as insulation.

Testing is important for any projects success. Any type of testing must be done in accordance with electronics testing standards. It is important to keep safety in mind when performing any testing on electronics. The IEC 61010-1 Safety of Measurement standard requires the measurement of current and voltage to be gathered by certain equipment (oscilloscopes, multimeters, etc.) that is considered safe for people to use without any specific protection. Similarly, the IEC 61010-2 discusses safe testing methods for products that emit radiation or lasers. This standard also discusses how to handle these potentially hazardous elements. Because this project includes both LiDar and RADAR modules that can output potentially dangerous radiation. Knowing these hazards and following these standards will allow us to accurately perform testing on these modules without damaging the circuit or any one around us.

Since the PCB board designed for this device includes multiple MOSFET transistors, it is also important to follow the testing standards for these devices. The standard IEC 62373 requires that MOSFET transistors be tested for bias-temperature stability, electrostatic discharge, and other stability tests to make sure the transistor is working properly and can function accurately in your circuit. Following and performing these semiconductor tests will help determine the temperature of the transistor and if extra heat dissipation components are needed in the circuit.

Before assembling the device or soldering items onto the PCB, it is important to examine and test each major component for hardware defects or other issues that do not conform to the items datasheet. The J-STD-001F standard defines methods ways of testing each component for defaults. This standard explains certain tests for the function of a product and what to look for when testing. This standard helps the user determine if the product itself is faulty or it does not function properly in a certain environment. Because both of our RADAR and LiDar sensors will be partially exposed to the outside elements. Testing these components before and after placed outside will help us to determine whether these components will function adequately in our design. This standard also explains how malfunctioning products should be repaired or disposed of.

The IEEE standard 32.220.01 lays out how certain sensors should perform and how they should be integrated into any design. This is an important standard for our project due to the sensor fusion needed to collect data from traffic. Following this standard will help us understand how both the LiDAR and RADAR should be behaving and how they can be integrated into our design.

In addition to these standards stated above, the device will be Dark Sky friendly. Due to the increase in light pollution over the years, the IDA began the “Dark Sky” Movement to encourage consumers to only purchase lights that emit a limited amount of light pollution. As a result, developers have begun to use or make products with these approved lights that produce less light pollution. In order to be Dark Sky friendly, these lights must minimize brightness and glare while producing only a small amount of blue light. Blue light has been known to cause the most light pollution and can cause eye discomfort for people and disrupt wildlife. Blue light has been shown to cause more pollution than any other type of light and can even reduce the glow from the sky at night. As a result, the ANSI standard C78.377 was created to limit the light temperature of any functioning outdoor light to a CCT (correlated color temperature) to 3,000 or lower.

Since the Situational Aware Stop Signal runs constantly, as any traffic light would, if proper lighting for the signal was not chosen, it could potentially create a large amount of light pollution where there weren’t any lights before. To avoid this, LEDs are used as the lights on the signal. LEDs can help lower emissions from carbon and their color is able to be controlled. Since the lights we are using are considered warm colors, the amount of harmful blue light produced by the device is much smaller than that of street lights. The brightness of LEDs can also be more easily controlled. While this initial prototype of the Situational Aware Stop Signal does not incorporate this, future design can display the lights at a lower brightness at night. This could save power as well as reducing the overall light pollution of this device.

3.2 Mechanical Constraints

The physical constraints of this device are set by the Manual for Uniform Traffic Control Devices (MUTCD) with consideration of the harsh Florida summers. Since this device is designed for operation in Florida, as it must be able to function in high temperatures. To ensure this device is reliable throughout the changing climates year-round, this device is operable between temperatures of 0°C - 60°C. To achieve this, the materials must be able to withstand high temperatures.

3.2.1.1 Design Constraints on LiDAR

On the LiDAR design itself there is no constraint since we do not design the LiDAR sensors, but when picking a LiDAR sensor it is important to note how they are powered as well as the connection interfaces involved. In regard to power, typically LiDAR is powered by a 5V input. So accordingly, to have working LiDAR we need to be able to allocate 5V at 135mA to each LiDAR sensor on our device without overloading our power delivery system or expending too much power and draining the battery of the system.

With connection interfaces, this depends on the exact sensor, some LiDAR use PWM, SPI, I2C, or UART connections. In our case the constraint is I2C, so our microcontroller design needs to be capable of connecting to 4 I2C LiDAR devices and capable of reading from all 4 as well so that we get accurate real time data to route traffic and warn other drivers safely.

3.2.1.2 Environmental Constraints on LiDAR

The environmental effects on the LiDAR device we are using deal with the operability and the reliability of the sensor. The operability of the device is affected because the temperatures involved can fluctuate between -26℃ to 60℃ . We have focused our design on the temperature range in Florida specifically, but this still gives rise to cooling issues. Most LiDAR do not function at high temperatures and because the stop signal needs to operate during the coldest and hottest times of the year, this can push the sensor to its limit. Normally the sensor has no problems with 38℃ temperatures, but the issue arises when the sensor is enclosed in our system, or if it is exposed to direct sunlight. The black body temperature of an object in Florida can easily be 60℃ which is the max limit on the operating range on our LiDAR sensor, so if it were to get any hotter at that point the sensor could be irreparably damaged and would need to be replaced. This is not an option for a system that is supposed to be inexpensive and a replacement for stop signs which need little to no maintenance. The issues with the reliability of the sensor have been previously discussed but it is important to note that because of the roadways and the temperatures involved that if we are not careful the sensor can be completely unreliable if the sensor cannot get reflected signals back to it. If there is too much noise, or if the angle of reflectivity is off then the sensor could be useless. So, to mitigate this we will rely heavily on sensor fusion, the physics of optics, and geometry to make sure that our sensors work in the environment that they need to. Both of these issues could possibly be eliminated by using different sensors but that would drive costs up above our allocated budget for the project.

3.2.1.3 Environmental Constraints on Microcontrollers

When dealing with microcontrollers, microprocessors, and other sensitive electrical components the most important factor is heat. Microcontrollers use sensitive components that are affected by high temperatures for example the RC clock signal in microprocessors is fast, but it is also changes drastically when exposed to varying levels of temperature.

That alone is not the only problem with microcontrollers, as expected microcontrollers cannot get wet or be exposed to excess amounts of water or electricity. Otherwise they may short connections or burn components, and this can lead to partial failures or full failures where the microcontroller can no longer work properly.

3.2.1.4 Safety and Operability Constraints for Microcontrollers

The main constraints that must be considered when selecting a microcontroller are processor speed, available memory and power draw, and circuit safety. When designing a microcontroller each of these must be fully considered before using specific components, especially when the application is related to human safety.

With regards to the security of the microcontroller the best way to hide information is to enclose as many wires as possible in a PCB so that the signals are not able to be intercepted by another device. Even with this done all external communication via UART, SPI, or I2C can be intercepted using extra hardware, so it is possible that a device that uses these externally can be reverse engineered with all signal information. For a commercial product this is not the best for the security of the device, but if that system does not store important data like user credentials or any other relevant user information this problem is most likely not an issue.

3.3 Power Constraints

This device is only powered by the energy provided by the sun; thus, the project was challenged to produce as much power as possible while consuming the least. This has been done by maximizing solar panel surface area as well as using power efficient hardware and conservative software. The Situation-Aware Stop Signal is in a black box suspended in the air, so the physical constraints have also defined the power design. This limited what batteries were usable in our application due to temperature and size constraints.

3.4 Software Constraints

An interesting phenomenon exists in the relationships between power, hardware, and software. There exists a cascading effect, beginning at power and ending at software, where a stage is only able to proceed with the resources it is provided by the previous stage. In this waterfall-like process, decisions, or constraints, imposed at each level restrict those below. Located at the end, software is the recipient of constraints imposed by both power and hardware. Decisions made at these levels directly impact the kind of software that can be designed and the capabilities it may render.

In the Situation-Aware Stop Signal, a very low budget in power defined first the hardware, then the software of the system. Power-efficient hardware generally suggests modest performance and this project made no exception. Having chosen the MSP432P401R microcontroller as the central processor of the stop signal, we acknowledged a trade-off in performance for greater power savings. Although the MSP432 remained a very capable MCU, this trade-off affected software by introducing constraints in the following areas:

- Memory
- Storage
- Processing Power

3.4.1 Memory

Memory refers primarily to the amount of RAM provided by the MCU. The MSP432P401R contains 64 KB of RAM. For a microcontroller, this amount of memory is average but usable for most simple embedded systems. For more capable computers running large, GUI-based applications, this quantity of RAM would be unsuitable for any practical purpose. The size of RAM limits the size of the executable programs that can be loaded onto it and determines the use of the system at large.

In the case of the Situation-Aware Stop Signal, a GUI was not needed, and the algorithms needed for proper management of the intersection were relatively simple. The memory footprint of the stop signal software was kept low by keeping its scope at an embedded level. With the addition of a real-time operating system kernel, the stop signal software was in the size of only tens of kilobytes and was within the bounds set by the microcontroller. As such, memory consumption was considered during both the design and implementation of software.

3.4.2 Storage

Storage refers to the amount of onboard flash storage built into the MCU. The MSP432P401R houses 256 KB of onboard flash. As with RAM, this quantity is only suitable for an embedded application and means that there is little room to expand software or add on large libraries. Such a limited amount of RAM also implies that no complex data sets should be used, neither large file types – more of a concern should a file management system be employed. An application adapting to this limitation should reduce the number of files, particularly executables, needed for its execution.

In the software for the Situation-Aware Stop Signal, we did not anticipate loading onto the flash storage more than the program to be executed. We also did not anticipate a need to store additional files on a file system provided by the real-time operating system. However, we believed there to be sufficient space remaining such that this would be possible. To better contain the software and to reduce the number of executables required, we designed a multithreaded application. This approach was representative of a goal to maximize the density and minimize the footprint of the software.

3.4.3 Processing Power

Processing power refers to the speed at which the MCU can normally operate. The MSP432P401R runs at a clock speed of 48 MHz. This speed is relatively high for microcontrollers but very low compared to personal computers. While the MCU remains very capable at 48 MHz, this magnitude limits it to more modest and less processor-intensive tasks. While it surely succeeds with basic arithmetic and string manipulation, such processor-intensive tasks as image processing and manipulation prove difficult for the MCU to handle.

Fortunately, we did not foresee issues with the demands of the stop signal software. With this issue in mind from the beginning, we sought an ARM processor, knowing its advantages and capabilities over other kinds of processor architectures in embedded devices. The sensors chosen for this project, RADAR and LiDAR modules, reduced the need for complex computation on the main MCU by doing their own pre-processing before delivery. What remained for the MCU was the interpretation of pre-packaged data and decision-making based on it.

4 Research

While there are multiple solutions the same problem, there is only one best solution. In observance to this, we researched and addressed possible solutions for the problem we have been presented. Due to negligence of drivers and ineffectiveness of current traffic signals, accidents happen every day. The current traffic device for the overlooked, lower traffic areas are over 100 years old and lack ability to efficiently control traffic and prevent accidents. To solve this issue, we researched ways to optimize traffic device hardware, software, power consumption, and power delivery. Addressing all effective solutions helped in the design and implementation of a situation-aware stop signal.

4.1 Hardware

Hardware is the physical components which the product is constructed from. Since there are several solutions to the same problem, it is important to explore all possible options. For our project, we created a device that detects threats, protect drivers, and schedule traffic. After identifying these goals, hardware has to be investigated to satisfy these conditions under our project constraints. When considering hardware, the software implementation had to be researched. These two work in concurrence and affect the functionality of the project. By undergoing an ample amount of product research, a suitable component can be easily chosen, and future rework can be avoided.

4.1.1 RADAR

As technology is quickly becoming more intelligent, modern gadgets have incorporated sensors in order to become both more accurate and more user friendly. Car companies, for example, have added side and rear bumper sensors to let the driver know when other cars are approaching or are in the driver's blind spot. Similarly, smart doorbells or security cameras are using sensors to alert a homeowner if there is any movement outside the door or around the house.

While these smart devices use sensors to detect objects within a close proximity, for this project, we need a sensor that detects objects that are farther than a few feet away. Unfortunately, the ultrasonic sonar sensors used for car blind spot detection lose accuracy over larger angles and can be affected by different types of surfaces. Even though some smart home devices use other forms of sensors, most use infrared sensors that detect heat and movement. The issue with infrared sensors becomes false or unnecessary readings that it could pick up. These sensors could pick up pedestrians, bikers, or even animals.

As a result, RADAR sensors are a viable option to achieve the goals of the project. RADAR or radio detection and ranging sensors transmit radio waves, then use the reflections from these radio waves to detect nearby objects. To detect how far away an object is, these sensors will use the time the radio waves take to be reflected back to the sensor to determine the distance between the sensor and the object detected. These radio waves are generated by a magnetron, which generates these radio waves by the rapid movement of electrons passing over a large magnet (which has generated a magnetic field along its length, as well as an electric field). As a result, microwave radiation is emitted. Below, Figure 3 shows a visual explanation of radio waves leaving the sensor and bouncing off nearby objects. The radio waves will then return to the sensor and measured for its time of flight.

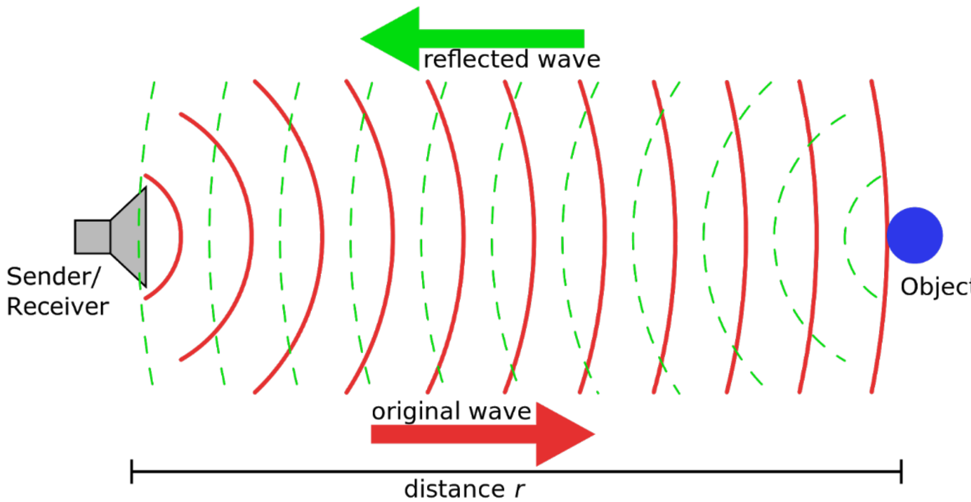


Figure 3: Visual Representation of RADAR Sensor (Public Domain)

The waves produced from the RADAR vary depending on the use and type of sensor. For example, while smaller sensors can produce waves a few centimeters in length, other larger RADAR sensors can produce waves nearly a meter in size. In the case of RADAR, shorter wavelengths are more effective in detecting smaller particles such as raindrops and insects. The drawback of RADAR signals that use smaller wavelengths is that attenuation can occur. Attenuation is the process of interrupting or reducing a signal, force, or any other oscillation. An example of attenuation is precipitation. When droplets of water interrupt the emission of RADAR waves by the sensor, inaccurate readings pertaining to the object's distance or location could be returned from the sensor. When using larger wavelengths, the issue of attenuation is reduced and there is little to no hindrance or absorption from intervening particles. Larger wavelengths will also produce less accurate readings as a result of the lower frequency.

The relationship shown below in Formula 2 relates wavelength to frequency and shows that with a large wavelength (λ) and the constant speed of light (c = 299,792 kilometers per second), the frequency (v) will be a very small value. Alternatively, a small wavelength will produce a high frequency. Since RADAR sensors can have a variety of different wavelengths, they can also produce a wide range of different frequencies. The frequency will not only affect the accuracy of the sensor, but it will also affect the cost.

$$c = v$$
$$E = hv$$

Formula 2: Light Equations

4.1.1.1 RADAR Waveband Spectrum

Similar to the electromagnetic spectrum, there is a waveband spectrum that presents each RADAR sensor and this spectrum organizes each RADAR sensor based on its frequency. Below in Figure 4 is a visible representation of the waveband spectrum.

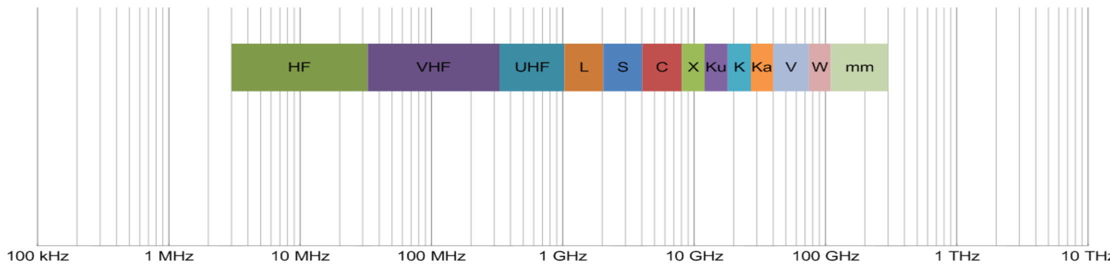


Figure 4: RADAR Waveband Spectrum (Public Domain)

The smallest frequency bands, as shown in the figure above in the dark green section, are the HF and VHF RADAR bands. These frequencies have been used since World War II and are one of the first devices that utilize radio wave technology. While the accuracy of using these frequencies is lower, as stated above, these frequencies can still be effective for larger scale uses such as military transmission bases and broadcasting stations. These bands, however, will not be helpful in this project due to the limited accuracy.

The next smallest level of frequency bands are the UHF frequencies, as shown in the figure above in the teal. This frequency set has been proven to be the best for tracking objects that are far away. For example, RADAR sensors of this frequency have been used in surveillance technology used by the Air Force to detect other aircraft. These frequencies are also not as vulnerable to weather-based attenuation. These frequencies are good for long ranges, but they are not as effective in detecting smaller objects and are very costly. As a result, these will not be helpful in this project.

The next frequency range is the L-Band RADAR, shown in the figure in orange. Like the UHF frequency range, these RADAR sensors are most effective in long-range object targeting. These sensors require a large antenna or radio wave transmission device. While these frequencies also have a large range of object detection, these sensors require larger hardware than would be necessary or fit in the scope of this project.

Following the L-Band is the S-Band RADAR, shown in Figure 4 in dark blue. These frequency ranges have similar effects to those bands stated above, but the attenuation factor becomes noticeably larger in this frequency band. The power required for this frequency band is also larger than the power needed for the lower frequency bands. At this point in the waveband spectrum, the range of the sensor begins to decrease; however, it is not a large amount of range loss. These sensors can still be used in military bases to detect aircraft. While this range presents a happy medium for both limited attenuation and higher accuracy, it still requires large hardware and has a high cost. The next frequency band, shown in the figure above under G has similar traits to that of the S-Band RADAR.

In the figure above in light green and light purple lie the X and Ku Band RADARs. These bands approach the middle of the spectrum and represent where the size of the hardware needed begins to decrease. While the attenuation factor is still higher than those of previous bands, it does not inhibit the accuracy of these sensors over large distances. These sensors are commonly used for navigation of both submarines and missiles. For this project, even though this sensor can find objects over a large range accurately with a smaller antenna, the power and cost required for this sensor was too large for the purpose of this project.

As the colors in the figure begin to become less saturated, the range begins to drastically increase for these sensors. These sensors can still be used outside but can only detect objects that are within a very short distance (a couple of meters) with less accuracy. While the hardware becomes more compact, the cost and the power are still larger than the limits of our project.

Finally, from the pale red section onward in the figure above are the W-Band RADAR sensors. These are used today most commonly on car bumpers for parking and lane assists. With this type of application, the high attenuation does not negatively affect the accuracy of the sensor. The accuracy of the low frequencies and small size of the hardware allow this sensor to be applied in a wide variety of smaller scale applications. The power required for these sensors is much smaller than the power required for any of the other sensors shown above.

Millimeter wave sensors are a type of RADAR module that emits waves within the W-Band. As the name suggests, the radio waves produced by this sensor are only millimeters in length which allows it to have a higher frequency. This sensor produces a frequency within the range of 30GHz to 100GHz. Due to the small wavelength, this sensor has a larger bandwidth which decreases the sensitivity as well as the sensor's signal-to-noise ratio. With such great accuracy, this type of sensor can detect objects moving at high speeds with ease. For example, according to Texas Instruments, their mmWave sensors can detect moving vehicles from up to 60 meters away traveling at a speed of up to 100 km per hour. These sensors have also become popular among current drone designs due to their ability to not only detect objects, but they can track their location as well. Unlike the sensors that use lower frequencies, this sensor can detect smaller objects along with a range of motion that is a fraction of a millimeter.

4.1.1.2 RADAR Safety

RADAR sensors generate a type of electromagnetic fields known as RF fields. These fields can easily flow through exposed tissue or skin and can produce heat under the tissue. The intensity and danger of this heat depends on the RADAR sensor's frequency. With a frequency below 10GHz, the sensor starts to have minimal potential health risks. Specifically, the intensity of heat measured under the skin over a period of time is known as Specific Absorption Rate, which is measured in W/kg. Once the Specific Absorption Rate or SAR reaches 4 W/kg, the RF radiation begins to cause damage to those exposed. The mmWave sensor, however, has a range of 30GHz to 100GHz, as stated above.

Exposure to this large of a frequency can output a large amount of power per unit area (around 1000W/m^2). This becomes more dangerous and can cause those around it to experience health issues such as skin burns.

While these RADAR sensors may sound dangerous, there are several factors that can reduce the negative health effects drastically. For instance, RADAR sensors, including the mm wave sensor are not constantly transmitting radio waves. These sensors send out these waves in pulses, which significantly lowers the power output. These sensors do not emit radio waves from all directions. Instead, the small sensor will transmit these waves in one direction for only a certain range. Finally, these health issues can occur when people are in a close proximity of a RADAR sensor for a long period of time. For the case of this project, these sensors will be placed well above the road and will not be able to affect those on the ground. People driving in their cars will only be in the vicinity of the RADAR for less than five minutes. As a result, both pedestrians and drivers will not be exposed to possible harmful radiation that could result from the mm wave sensor.

4.1.1.3 RADAR Drawbacks

Aside from the risk of radiation burn, there are some other drawbacks to the mm wave sensor. As stated above, sensors with higher frequencies and smaller wavelengths have a greater risk being attenuated by the environment. This attenuation could come from the weather or animals and can interrupt the readings from the mm wave sensor. This attenuation could also cause the sensor to pick up false readings, or in our case, detect a car that is not actually there. To prevent this issue, we will include a sort of thin casing around the sensor to prevent these miss-readings. There have also been cases of mm wave sensors being affected by cell towers and Wi-Fi hotspots. In order to limit these effects of surrounding transmission signals, the device will be placed in an area that is farther away from these cell towers or Wi-Fi hotspots. The thin casing as well as the placement of the sensor within the light should also help solve this issue.

Another drawback of using these small mm wave sensors is the antenna and sensor placement within the device. While a small antenna can be a positive for size constraints and pricing, it can also negatively affect the readings if placed in the wrong place. For example, placing the antenna on the side of the device may cause a shadowing effect. Shadow could act as an attenuation factor that will inhibit the transmission of radio waves. Additionally, placing the antenna in a certain orientation may allow for limited shadows, but will not detect the objects we are looking for (i.e. cars on the road).

When the RADAR antenna is placed at an angle and the beam width is less than five degrees, the coherence time increases slightly. The coherence time is the time in which the sensor's impulse response is not varying. If the RADAR antenna is placed straight or directly facing the object it is trying to detect, when the beam width angle is less than five degrees, the coherence time decreases exponentially. Once the beam width of the sensor reaches five degrees, the angled sensor and the non-angled sensor both decrease slightly. As a result, there is an optimal beam width for maximizing the time in which the radio waves do not vary or are not interfered with. Using this beam width with help maximize the accuracy of the RADAR sensor.

4.1.1.4 RADAR Benefits

The mm wave sensor was initially built with discrete elements that may have appeared to improve the device at the time but have also increased the total power consumed along with the cost. As time has passed, however, these sensors have become more compact and have been added to newer analog and digital components such as a clock, microcontroller, and many more semiconductor devices that have widened the sensor's uses and capabilities.

There have been studies that show that these RADAR sensors can accurately detect the movement and locations of cars. For example, in Shimoi and Kakamura's Application of Millimeter-Wave RADAR Traffic Observation System study, they observed that this sensor along with other observation elements are not affected by weather conditions and were suitable for the measurements of traffic on large highways with limited reflection errors. Studies like this are promising for our application and provide proof that these mm wave sensors can be used in the intelligent stop signal.

While there may be some challenges and drawbacks of this sensor, with small adjustments such as placement and angle of the sensor the mm wave sensor will still produce accurate readings that will help us determine whether cars are nearing the Situation-Aware Stop Signal.

4.1.2 LiDAR

Accompanied with the mmWave RADAR sensor, we plan to use multiple LiDAR sensors to augment the RADAR technology on the stop signal. RADAR works well over large areas, but generally is not good for getting pinpoint data on small areas. Due to this, we found that the best strategy would to use multiple sensors as well as different types of sensors. Once this decision was made the simplest solution would be to use multiple RADAR sensors, but this solution is expensive and impractical for a stop sign replacement. Ultimately we landed on LiDAR as a possible solution as it was cheaper than accurate RADAR, but also was less susceptible to the accuracy issues that ultrasonic sensors have which is a good compromise between cost and reliability.

4.1.2.1 How LiDAR Works

LiDAR systems are time-of-flight-based systems just like RADAR and ultrasonic sensors. In the case of LiDAR, the sensor will first send a beam of laser light out towards the object. After that, the beam of light will be reflected off an object and come back to the sensor. The sensor would then calculate the time it took for that signal to come back to it. Then that would be used to calculate the distance to the object using the known speed of light. This technique, illustrated in Figure 5, is seen in most distance sensing systems as it is simple, accurate, and reliable. RADAR sensors use the same technology with different wavelengths and frequencies, and ultrasonic sensors use it with sound and the known speed of sound. A common technique in LiDAR systems is to send a pattern in the laser signal and when the laser sensor receives a signal with that specific pattern back it then records that time. That time is then used with the speed of light to calculate the distance to the object. This is done because signal noise in the environment causes issues for the sensor, so they do this to allow the sensor to detect the signal when it comes back.

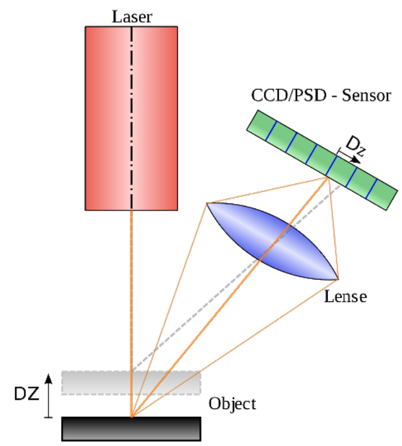


Figure 5:Simple Time-of-Flight-Based LiDAR System (Public Domain)

LiDAR operates in the electromagnetic spectrum, which is the same as RADAR, but at a different wavelength. Most LiDAR use wavelengths the near infrared range (~10µm) so that their laser is not visible to the human eye but sensitive enough so that it won't pass through solid objects. These ranges, shown in Figure 6, are used to increase the distance that the wave can travel. This is beneficial for our system because it is used in roadways and if the laser used visible light, it could cause serious road safety concerns as it would have the potential to be blinding or distracting to drivers during the day or night.

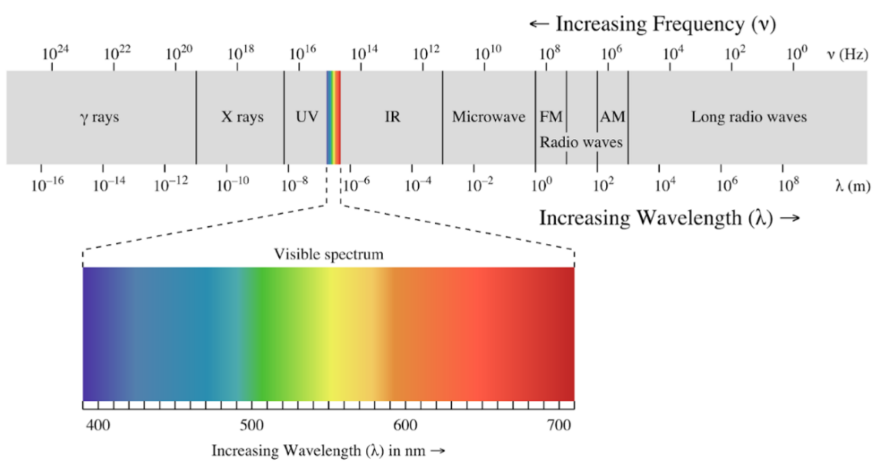


Figure 6: Electromagnetic Spectrum (Public Domain)

Most LiDAR sensors have a wavelength of about 10 micrometers and would be anywhere in the 30THz to 300THz frequency range. To pick a LiDAR sensor, it is important to know the beam diameter at the laser aperture and the beam divergence ratio. These two specifications on the LiDAR design allow users to calculate the beam diameter at longer distances and to accommodate for that in their analysis of the LiDAR sensor data output. Shown in Figure 7 below is the relationship that as the distance from the sensor grows, the diameter of the cone that the laser produces grows. This can cause false readings for data at long distances as the signals are partially degraded and can be overcome with noise. General purpose LiDAR sensors do not have good range because of this noise factor. For the sensors we are considering, after about 40 meters the sensor can no longer accurately get data on objects because the signal from those objects is too scattered and the noise from other electromagnetic sources drowns out the information embedded in the light signal. More expensive sensors typically get better ranges, but the drawback is these systems are also not able to be integrated into a small form factor as they normally require tripods to function due to their size and sensitivity.

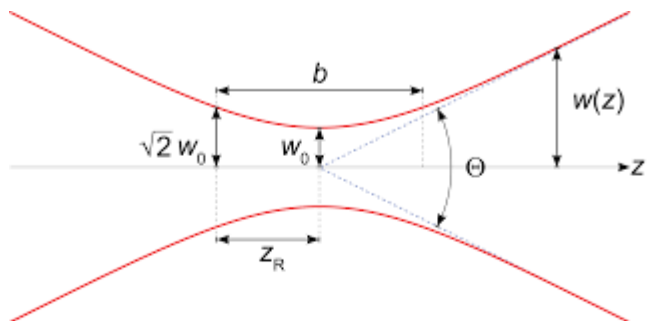


Figure 7: Beam Divergence in Laser Systems (Public Domain)

4.1.2.2 Types of LiDAR

The most common types of LiDAR sensors are topographic, bathymetric, and terrestrial LiDAR. For the stop signal, the type that is suitable is terrestrial LiDAR. This is because topographic and bathymetric LiDAR systems are used on planes and drones to survey large unexplored areas where research institutions need data to perform analysis on those areas. This leads us to look at terrestrial LiDAR which is most commonly seen now on small robotics systems and autonomous vehicles. Terrestrial LiDAR has two subtypes known as mobile and static types. The mobile subtype is not good for our system as our system does not move from its location and does not need GPS data or any other navigational data to function. The static type of LiDAR works perfectly for our system since the system itself is static and does not need to worry about navigational data.

For us, a sensor that would typically be found in a laser range finder is the best type since what we are doing with the LiDAR is collecting data on the distance and speed of the vehicle that it is currently tracking. When looking at laser rangefinders, you will find that most of those devices can work up to ranges past 500 meters, but they are expensive. Minimizing production costs is a major factor for this project because this system is designed to replace relatively cheap stop signs, so we do not need sensors that are that powerful. Accordingly, we are using sensors that have max sensor range of roughly 40 to 50 meters. This lowers the cost and it simplifies the design because the power draw of the cheaper sensors is significantly less. A downside to using these cheaper sensors is the level of accuracy they can maintain in non-ideal situations and their robustness in harsh outdoor environments. Another type of LiDAR sensor that could be an option for the project is a 360° sensor, the imaging of which is shown in Figure 8. The 360° view is very useful for gathering data but for us using sensor like this would double the cost and would be more work just to filter the unnecessary data involved. We chose a solution that will instead use 3 to 4 cheaper sensors with a better range. This way data collection is straightforward, and the project does not go over budget.

Figure 8: High Resolution 360 Degree View LiDAR Imaging (Public Domain)

4.1.2.3 Common Issues with LiDAR

As good as LiDAR is for detailed measurements and reduced noise compared to RADAR and other distance calculating methods, it can be still unreliable in harsher outside conditions. This typically depends on the sensor itself and how it is used in the device. The first major issue deals with reflective surfaces. Normally these surfaces are better for LiDAR as they reflect the laser light signals back towards the receiver on the device but depending on the angle that light hits the surface, those reflected beams of light can be disrupted completely. If that happens, then the signal is sent in a different direction than the receiver and this causes misses and lag in the data stream. Another common problem for LiDAR when it is outside in the sun, is that the receiver can get saturated. This is common for most sensors and it depends on the configuration that it is designed for. Due to the electromagnetic saturation, the device cannot sense the signal with all of the extra interference and this causes cut outs in the responses and lowers the reliability of the LiDAR sensor.

Generally, these issues are based on the overall quality of the LiDAR device; therefore, more expensive sensors are needed to get more accurate and detailed results at medium to long distances in harsher environments. This does not cause many issues as the majority of false readings and incorrect data will come from generic noise and non-ideal operating conditions. We hope to mitigate this by enclosing the LiDAR sensor in the housing for our device so that it is protected from elements that could cause problems. We also plan to angle the sensor so that it minimizes the reflectivity problem discussed earlier.

Slight noise in the data feed should not pose to be a major problem for this system because the objects we are dealing with are large and reflective, so the data will average out to the correct readings even with some level of electromagnetic noise. Surprisingly this noise would cause more problems for the larger more detailed systems that are trying to get accurate data, but this can be ignored for the system we are making. At most, the solution to noise like this would be to take an average of sensor readings over time to guarantee that no sensor reading is affecting the device directly. Another solution would to be to ignore data that is inconsistent with any other data that has been collected by other sensors using what is known as sensor-fusion.

4.1.2.4 RADAR, LiDAR, and Sensor Fusion

For our project, most of the issues that we will encounter with the RADAR and LiDAR sensors can be solved by sensor fusion, an example of which is provided in Figure 9. By using both RADAR and LiDAR for sensing vehicles, it drastically improves our chances of catching the right information at the right time by allowing our system to verify the data that the device receives. If there is ever inconsistency, it can be dealt with accordingly based on the current and previously acquired data. Most sensor-fusion techniques are involved in computer vision and try to gain more data about edges and other important features using larger more complicated systems, but sensor fusion gives us a good method to validate data information to guarantee driver safety.

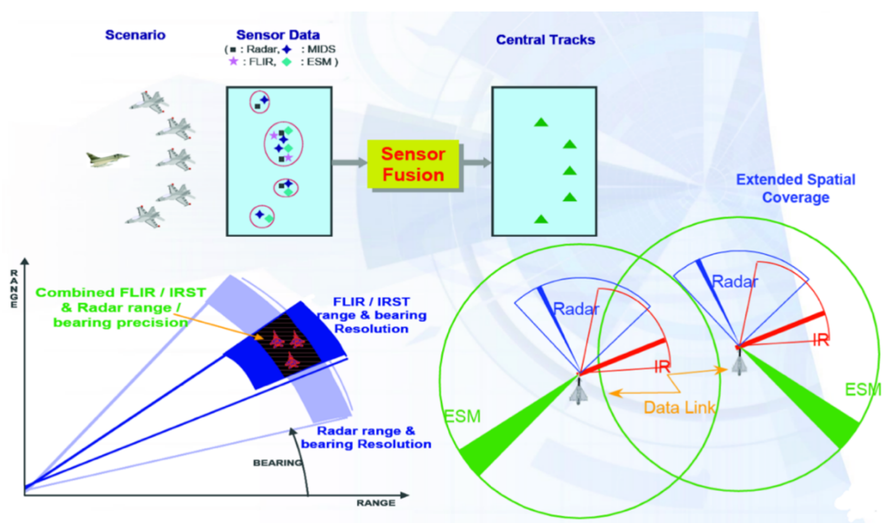


Figure 9: Sensor Fusion with RADAR and LiDAR (Public Domain)

4.1.2.5 Safety Concerns

An issue with LiDAR sensors is that they can cause damage to sensitive tissues so special precautions must be made. The first major precaution is that LiDAR systems are typically power limited so that the laser modules themselves cannot output damaging levels of laser radiation. The second precaution is that a LiDAR sensor typically pulses so that they do not overexpose the environment to this type of radiation for long periods of time, which is the same way that RADAR solves that problem since RADAR can expose people to enough RF radiation at short range that it can heat up objects. To aid in keeping the LiDAR safe for drivers and pedestrians, and to save on power consumption, we also want to sleep the sensors until they are required. We do this by using the RADAR as an interrupt for the device allowing us to only use the sensor when a relevant situation for its use is about to occur.

4.1.2.6 Environmental Constraints on LiDAR

The environmental effects on the LiDAR devices affect the operability and the reliability of the sensor. Most LiDAR sensors do not function well at high temperatures and because the stop signal needs to operate during the coldest and hottest times of the year, this can push the sensor to its limit. The black body temperature of an object in Florida can easily be 60°C which is the max limit on the operating range on our LiDAR sensor, so if it were to get any hotter at that point the sensor could be irreparably damaged and would need to be replaced. This is not an option for a system that is supposed to be inexpensive and a replacement for stop signs which need little to no maintenance. To solve this, we will most likely need to keep our system open enough for airflow and cooling elements so that the electronics do not get hot in the casing. The issues with the reliability of the sensor have been previously discussed but it is important to note that because of the roadways and the temperatures involved that if we are not careful the sensor can be completely unreliable if the sensor cannot get reflected signals back to it. If there is too much noise, or if the angle of reflectivity is off then the sensor could be useless. So, to mitigate this we will rely heavily on sensor fusion, the physics of optics, and geometry to make sure that our sensors work in the environment that they need to. Both issues could possibly be eliminated by using different sensors but that would drive costs up above our allocated budget for the project.

4.1.3 Microcontrollers

Microcontrollers are easily the most common type of computer in modern day computing. Computers are commonly thought of as large machines that host desktop operating systems that are capable of playing games and browsing the web. Today, however, most of the computing power in the world is controlled by microcontrollers. With the expansion of the Internet of Things (IoT), microcontrollers have become increasingly more common and necessary in daily life.

Microcontrollers, two of which are shown in Figure 10, are perfect for small, embedded applications that require low power components but also real-time analysis and control. In today’s world, microcontrollers exist in almost every electrical device we see. Computer monitors, keyboards, microwaves, refrigerators, cars - the list goes on. This speaks to the importance of these devices and how they can affect the world around them.

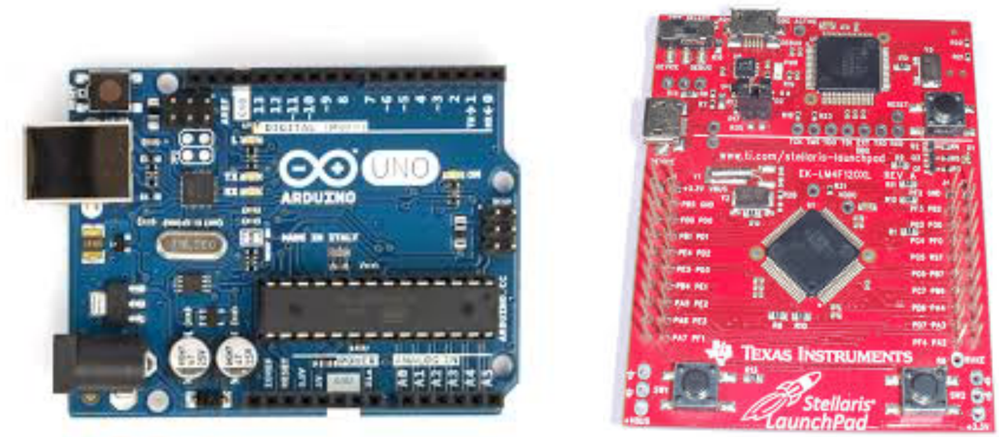


Figure 10: Arduino (Left) (Public Domain) - MSP432 (Right) (Public Domain)

Microcontrollers typically consist of many parts, the two most important being the microprocessor and the memory. The microprocessor and memory are the heart of a microcontroller and without either the system would not work. Then the microprocessor is there to execute that code in a meaningful way. Other parts can include timing crystals, USB ports, and Wi-Fi/Bluetooth chips. This gives microcontrollers flexibility to be changed and modified to fit their exact use case, which is why these devices are so common in embedded systems. A table of our design choices with respect to this is included in design section 5.2.2.1.

4.1.3.1 Microprocessors

Microcontrollers rely heavily on the processor they are designed for. Because of this, the microprocessor is typically the first component a microcontroller is designed with. Microprocessors have many different engineering standards, most of which deal with processing speed, number of cores, architecture, and form factor. When designing a microcontroller, all these factors must be evaluated as they can greatly affect the performance and size of the device.

Processing speed is consistent with most modern microprocessors. The range for processor speeds is from 4kHz to above 100MHz in some cases, most being under 48MHz for general processing. Compared to a normal CPU today which can be ran up to 5.0GHz or above this is drastically slower. Although this is a drawback to using a microprocessor, this is accepted because most embedded systems have no reason for requiring 5.0GHz speeds. To add to this the design of a microprocessor is typically not based on speed. Typically, microprocessors are designed around low input latency and low interrupt latency which allow the processor to react faster to changes in the environment and control signals that they are typically subject to.

4.1.3.2 Memory

The memory in microcontrollers has three different configurations. The first configuration is RAM, the second is ROM, and the third is Hybrid. The de facto standard for most microcontrollers is the hybrid memory architecture, exemplified in Figure 11, because it allows for the separation of memory components. The hybrid model uses ROM or Flash memory as the operating system and program storage. Then it uses the RAM as active program memory. This is done because RAM is volatile memory and cannot store information permanently. For a microcontroller, the program storage must be nonvolatile since the programs need to be re-run occasionally on timers so losing those instructions is not an option for continued use. Both RAM and ROM come in IC chips and other smaller components that are integrated onto microprocessors.

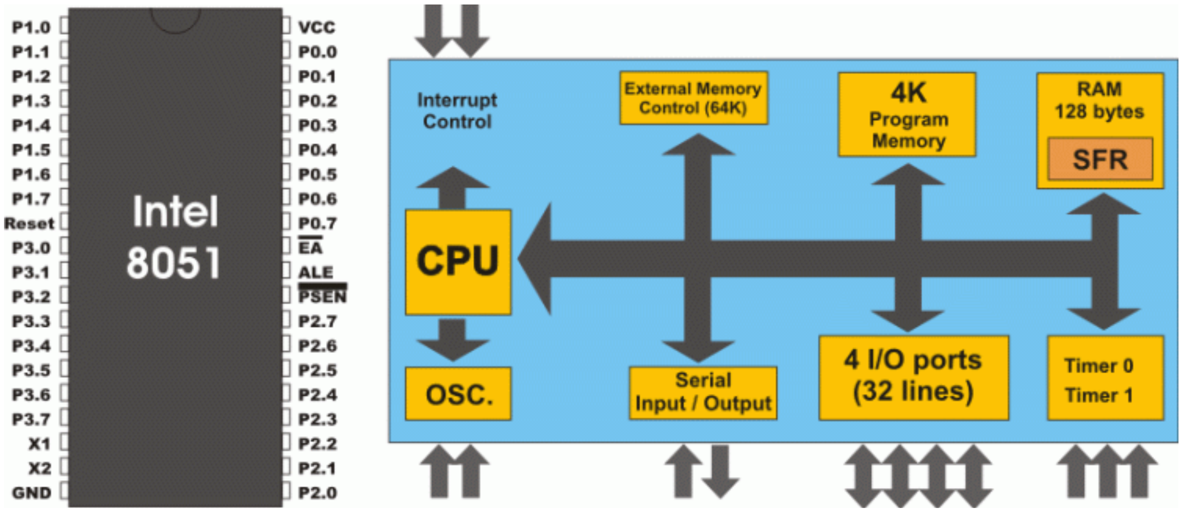


Figure 11: Intel 8051 Example Microcontroller Block Diagram (Public Domain)

ROM is typically nonvolatile, which serves as a good solution to this, but ROM and Flash memory are expensive to write to and are not good for memory that changes quickly. A hybrid solution is preferred. It allows the program to be stored so that it is not lost when the microcontroller powers off, and the normal memory for program execution is ran on RAM which is faster and more useful for objects and variable storage.

4.1.3.3 Low Power Mode

A large factor in microcontroller design is the ability to function in low power mode. This changes depending on the processor, but the idea is that a signal is sent to the processor to send the unit into what can be considered a sleep mode. When this is done the processor and any other components will be inactive and draw less, or no, power. Some processors can even draw as low as microwatts in this mode and this is important to most power limited systems. Typically, microcontrollers consume at least 1W or more of power depending on the processors and components involved, so being able to limit this on a system that uses batteries for example will always improve the usability and the uptime for that device.

4.1.3.4 Interrupts

Interrupts are one of the main driving features of embedded microcontrollers today. Interrupts allow for quick responses to important flags in software. Interrupts, as shown in

Figure 12, are configured and set in hardware, so their responses are near instantaneous and allow for error control and incredibly efficient responses even while the microprocessor is in sleep mode. The main uses for interrupts are to deal with hardware level errors or waking the processor from sleep mode on timers. This all depends on the system into which the microcontroller is being integrated. Generally, any time that the device can, it should be in sleep mode. This cuts down on power draw and wasted CPU time thus increasing the efficiency of the device in multiple domains. The drawback to this is that the CPU does take extra time to wake up, although the wakeup time for a microcontroller is typically very small is can be large enough that accidents can occur on a roadway, or time enough for a component to burn out if there is a voltage flag warning.

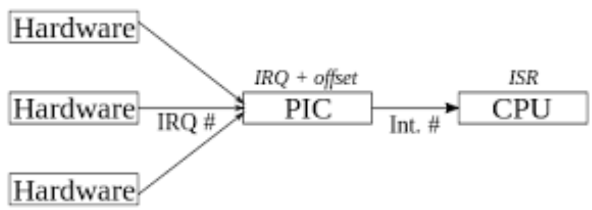


Figure 12: Hardware Interrupt Flow Diagram (Public Domain)

When an interrupt is encountered on a microcontroller device, what then happens is the device specifically goes into memory at a location stored in an interrupt vector table that is stored on the device memory. That memory location stores the ISR or 'interrupt service routine' which is a piece of code written by the embedded developer of the device to deal with that interrupt appropriately. Once that is done, the ISR exits and the microcontroller then returns to the main memory location where the device left off. This process is repeated anytime an interrupt is encountered but depending on the type of interrupt it may be possible that the ISR completely shuts down the device so that no harm may be caused to the system or people.

4.1.3.5 UART

Universal Asynchronous Receiver-Transmitter or UART is an asynchronous serial communication device. UART is asynchronous because, unlike other serial connection methods, it does not require a clock signal to sync the data between receiver and transmitter. UART on the sending device will send the signal sequentially and then the receiver will collect the bits sent in a shift register until the stop bit is received for that byte. This is done so that communication is cheaper for devices like microcontrollers. Less wires and connections mean there is less hardware to design and this simplifies that process and ultimately results in cheaper components. The downside to UART is that it requires one line for communication for each device, so it can get complicated in terms of hardware dealing with a multitude of devices.

Even though UART is asynchronous, this does not mean that one UART device can send to another without some configuration. When using UART, communication between the two devices must be set at the same baud rate, character length, parity, and stop bits. If that is not done, then the receiver will not be capable of reliably reading the data sent. Typically, this causes some type of "corruption" because the data received is being interpreted in a different way than it is intended to. An example of the UART message bit pattern is provided in Figure 13.

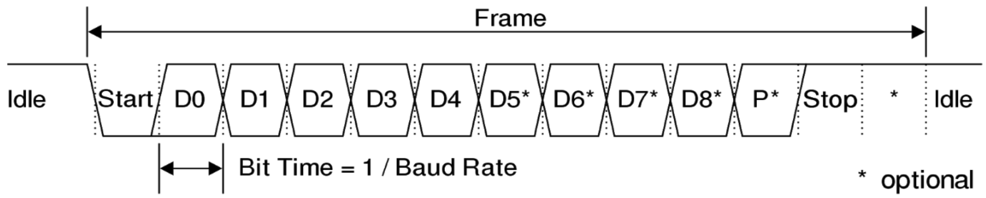


Figure 13: UART Message Bit Pattern (Public Domain)

4.1.3.6 SPI

Serial Peripheral Interface or SPI is a synchronous serial communication protocol that is commonly used in embedded systems applications just like UART. Unlike UART, SPI uses a master - slave architecture for connected devices. SPI specifically is a full duplex serial communication protocol, meaning that the connected devices can send data to the master device and receive data from the master device concurrently.

In SPI, each device uses the SCLK clock sent by the master device to sync their timings. With that there are 2 other inputs and one output, as illustrated in Figure 14. The MOSI or "master out - slave in" line is for sending data to the slave devices. The SS or "serial select" line, which must be an independent wire for each slave device, selects which slave device is to be addressed at that time. The one output for each slave device is the MISO or "master in - slave out" line and this is used so send data from the slave devices to the master device. Since all slave devices use the same MOSI and MISO busses, only one device may be active at a time. Otherwise, data will be corrupted and none of the active slaves will send data properly.

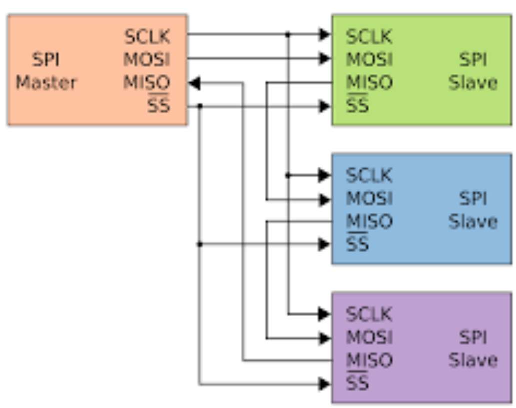


Figure 14: SPI Serial Interface Block Diagram (Public Domain)

SPI hardware can be achieved by using a processor, and two shift registers. The SCLK signal controls the shift rate in each shift register. The slave shift register receives bits from the master device on the MOSI line and at the other end of the slave shift register the last bit is sent out on the MISO line to another slave device, or back to master depending on the configuration. If there are multiple slave devices, then the signal is sent from the shift register to the next slave device. This is because these devices have to be chained together serially so this requires that data be passed from one slave device to the next in order to guarantee data is transferred to the correct device.

A downside to using the SPI protocol is that all flow control must be done by the master device, although this can be dealt with, it is sometimes inconvenient because you need to reroute clock signals to the SPI SCLK bus to guarantee it is timed correctly. Complexity is added to this protocol when different devices have different clock requirements and SCLK must be changed dynamically. One of the other downsides to using SPI is the increased size requirement due to the number of individual slave select lines that are necessary if you are using a lot of devices that use SPI as their communication interface.

4.1.3.7 I2C

I2C is a two-wire serial communication protocol that allows compatible devices to connect to each other much like UART or SPI. The difference with I2C is that addressing is different in I2C. Typically protocols like UART or SPI are for one device only or for chained devices in the case is SPI. I2C is different because each connected device must have a unique 7-bit or 10-bit address. This theoretically allows unlimited devices on a I2C bus. This behavior is incredibly beneficial for embedded systems that need to connect to many devices without requiring 2 or more GPIO pins for each device.

I2C has a max speed of 400kHz in 'fast mode', 100kHz in normal operating mode, and in some special cases there is an 'ultra-fast mode' that can go to 5MHz. This gives I2C some flexibility when dealing with faster devices and applications that require near instantaneous speeds. An illustration of an I2C configuration is given in Figure 15.

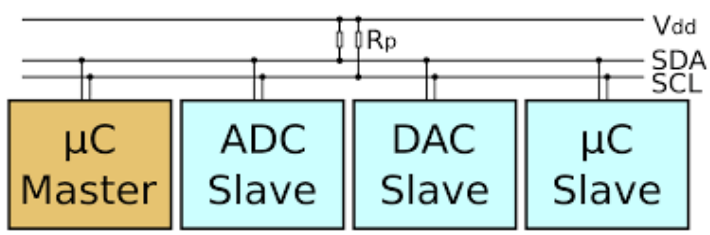


Figure 15: I2C Bus Diagram (Public Domain)

The I2C protocol itself is the following: the master device first generates and sends a 'start condition' which prompts all slave devices on the bus to get ready for a message from the master device. Then the master device sends the 7-bit or 10-bit device address so that the appropriate device knows it is being communicated to. Once this is done, an acknowledge signal is sent from that slave device letting the master device know that it exists. Then the master device sends a command signal letting the slave know that the master wants to use either write mode or read mode. Depending on what was chosen, the master device will either read from the slave device or write to that slave device causing it to do some action. Along with that functionality, I2C allows for consecutive start bits.

What this means is you can send a write command, then you can send another start command and switch to a read configuration. This allows for devices to be changed and then read from quickly and efficiently. After all this is done and the master device is done reading from or writing to the slave device, the master device will send a stop signal that will let the slave device know that data transmission if over and it may stop transmitting on that bus. A comparison of I2C, SPI, and UART is given in Table 1.

Table 1: Comparison of Communication Protocols

	UART	**SPI**	**I2C**
Speed	110bps – 256kbps	10Mbps	1Mbps
Number of Devices	1	Unlimited (Expensive)	Unlimited
Compatibility w/ sensors	Most sensors	Some sensors	Some sensors
Synchronous	No	Yes	Yes

4.1.3.8 JTAG

JTAG is an industry standard for testing and digital simulation for printed circuit boards. This standard was created by the Joint Test Action Group and has since been adopted by IEEE as a standard for electronic devices. Originally JTAG was used only for testing and debugging the sub blocks of integrated circuits, it also allowed for hardware monitoring of temperatures, clock speeds, voltages, and currents.

Now JTAG has been updated to not only do that but it also stores device programming hardware and can flash the ROM or flash memory on a microcontroller unit. This gives added functionality to users who wish to change the software on their system often. This interface is probably one of the more common ways to change software on embedded systems, but it has its drawbacks as it gives full control over the hardware and for secure systems JTAG may introduce security holes that are unacceptable.

4.1.4 Solar Panel

Living in a time when the total population is growing exponentially, the nation's power consumption has seen a large spike in demand. In the United states, the demand for energy had doubled in the past 30 years. Because the leading forms of power generation in the United States are coal and natural gas, the increase demand for power has given rise to a national concern for the environmental impact that the non-renewable power generation industries create. As a response to this, the power generation industries have ramped up their production of renewable energy production, which is accepted as an alternative that creates a smaller carbon footprint. One popular form of renewable energy is solar power. Solar power is popular because it only uses the sun's energy to produce electricity. Compared to the other forms of renewable energy such as wind and hydroelectric, solar power can be easily implemented on small scale applications.

Solar panels are made up of several photovoltaic cells. These cells can accept incoming photons and convert them to useable energy. As the photons, which are particles of energy, hit the surface of the photovoltaic cell at material specified wavelength, the energy of the photon is transferred to a pn-junction semiconductor. This allows the flow of electrons, or electric current, which can be utilized by electric devices. This phenomenon is called the photovoltaic effect. Figure 16 below demonstrates the photovoltaic effect through a PV cell.

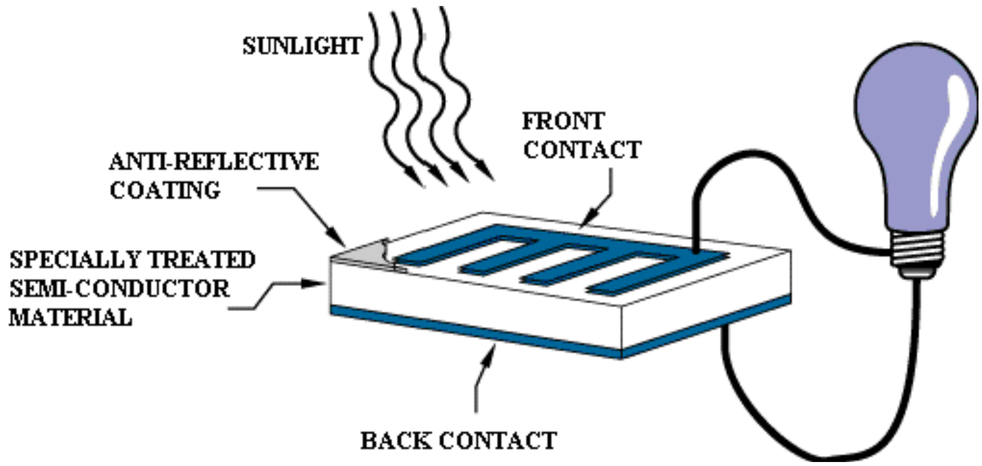


Figure 16: Photovoltaic Cell (Public Domain)

Semiconductors combine the properties of a conductor and an insulator into one device which at times can act as both. In photovoltaic cells, there are three layers of electron/hole concentrations which make up the semiconductor. The N-type of a semiconductor consists of a concentration of negatively charge electrons. The P-type of a semiconductor has an abundance of holes which are absences of negatively charged electrons. These two regions are separated by a depletion region which has no carriers of electrons or holes. When a power source with a large enough voltage is applied to the device, it allows free electrons from the N type region to pass through the depletion region and the P type region of the semiconductor. This flow of electrons creates the current that devices need to power them. When current is passing through the semiconductor, the material acts as a conductor. When there is no sunlight, or not enough sunlight to reach the required cut-in voltage, then the material acts as an insulator.

Solar panels are usually made from several solar cells. The difference between panels and cells is, solar panels have a protective backing, a weatherproof and transparent cover, and the appropriate wiring for an easy plug-and-play application. Alternatively, solar cells must be weatherproofed before installation and must have positive and negative contact patches opposed to protected wires. While creating solar panels from several solar cells requires additional work in installation, this provides the engineer with more flexibility to apply to their specific application. The engineer can position each solar cell closer together, covering them in one unified layer of resin or protective glass. Using separate solar cells also allows for more flexibility in wiring each cell up in series, parallel, or a combination of both. Solar cells can also be purchased at a fraction of the cost of a complete solar panel. For budget sensitive projects, wiring up individual solar cells may be a more practical alternative. Solar panels which are already protected and soldered together provide convenience, reliability, and time but the user is constricted by the specifications of its design.

4.1.4.1 Power Delivery

The most common photovoltaic cell is made of silicon semiconductors. This is due to their combination of effectiveness and low price point. These silicon photovoltaic cells require 1.1eV to turn on, and due to constraints of using a pn-junction semiconductor, providing additional energy would not provide a larger voltage. While the amount of voltage the solar panel can produce is constant, the amount of current which is produced can vary. The max current through the solar cell is dictated by the size of the panel. If the solar panel is larger, there is a greater surface area to fix additional solar cells; this would lead to more photons being accepted and a larger current output. Any additionally provided energy which could not be accepted by the solar cells would result in energy wasted as heat. Because of the specific requirement in energy to operate these solar cells, the solar cells also require a narrow range of frequencies. The appropriate frequency for these cells would be 886 nanometers. This can be found via Planck's equation, depicted in Formula 3. This relationship shows that energy and wavelength are inversely proportional.

$$E = h(Planck's\ constant) * f(frequency)$$

Formula 3: Planck's Formula

Most of the energy produced from the sun is not able to be used by the solar panel. Because the photovoltaic cells require 1.1eV of energy, with only a small band of variance, most of the photons are not be able to be used by the solar panel. The charge of a photon ranges as high as 3.6eV, so the spectrum of energies and wavelengths that the sun can provide is much larger than the spectrum that solar panels can accept. Both industrial and consumer grade solar cells range between 15%-22% efficiency. There are several factors that must be considered when designing/choosing a solar cell to be utilized by a solar panel. Solar panels are covered with a clear, weather-proof coating. These are usually a resin or a piece of glass. Having a nonreflective coating ensures that more of the photons that hit the glass pass through and not be reflected. The white lines that can be seen on solar panels are busbars. Busbars are small metal wires which allow current to pass through the solar cell. These bars reflect some of the photons that hit the surface, so the larger the busbars, the less efficient the solar panel is. All of these items must be considered when deciding on a solar cell to create a panel out of. Being mindful of these traits can result in a more efficient solar panel.

4.1.4.2 Solar Panel Mechanical Design

When considering building a solar panel from a collection of solar cells, one overlooked aspect is the solar panel encapsulation. It is important to use choose a protective coating that is nonreflective, so the photons can easily pass through. Many solar cells are coated in a resin. Resin is effective because of its ability to keep water off the cell. There remains little to no room for air, dust, or any intruding particles to land on the cell. The issue in coating a panel in resin is laying it over a large area while avoiding the emergence of air bubbles and other contaminants. An alternative to this is using a single plastic or glass sheet. Using this allows for a uniform, easily installed surface. Plastic is more prone to scratching and hazing, so glass is usually preferred between the two. Both plastic and glass sheets are more prone to allowing moisture in. Because it is difficult to create an airtight like resin creates, moisture can get under the plastic or glass sheet. Once moisture gets inside the glass, it could fog up, which would hinder a photon's ability to travel through it. To help with moisture issues on the plastic and glass panel applications, many solar cells are covered with an EVA film. EVA is ethylene vinyl acetate, which is a film that is heated onto a surface and shrinks to solar cells. Properly encapsulating solar cells ensures that the panel will both last a long time and efficiently deliver as much electricity as possible.

Solar panels thrive on direct sunlight, so ensuring they are located in an area free of overhead obstruction is imperative to their function. When using solar panels, it's important to understand the peak sun hours in the area of installation. Peak sun hours are the average amount of shining sun hours an area receives per day. These will vary by both region and season. Solar panels get peak solar radiation at solar noon when the sun is highest in the sky. In the sunny state of Florida, the residents benefit from 5.2-6.2 hours a day, which ranks it as the 7th most solar hours produced daily. While Florida can effectively utilize solar power, the amount of sun hours fluctuates on a day-to-day basis. Cloud coverage is a regular issue for solar panels, and their presence hinders the panel efficiency. While clouds and other opaque objects may block the view of direct sunlight to the solar panel, the panels will still function, as they do not block all photons from reaching the solar cells. Under cloud coverage, solar panels function at about 25% of its overall efficiency. If reliability and consistency are important to the application of the solar panels, having a battery backup or a connection to the grid is important to maintain functionality.

4.1.4.3 Types of Solar Panels

There are two types of silicon solar cells which are commonly created: Monocrystalline and Polycrystalline. Monocrystalline are made from silicon ingots. From these silicon ingots, slices are cut out to be made into solar cells. Silicon ingots are cylindrical, so it is common that monocrystalline solar cells have rounded edges as shown in Figure 17. Because these are made from the ingot, these are made out of the most pure, highest grade silicon. These high-grade panels yield the best output. Monocrystalline most commonly appear black to absorb as much natural sunlight as possible. Compared to polycrystalline, the monocrystalline are more efficient. Along with efficiency, monocrystalline are also more efficient in warm weather as temperatures rise. The largest downside to this type of solar panel is the price. Because of the process in production, monocrystalline are consistently more expensive.

Polycrystalline solar cells appeal to more budget-constrained applications. These panels can be obtained at a fraction of the cost because of the way they are produced. Polycrystalline solar cells are made from melted silicon which is poured into a perfectly square mold. This process wastes much less silicon than monocrystalline, which reduces the cost. The result is a less efficient solar panel which is not as resilient to high temperature weather conditions. Polycrystalline cells are commonly blue in color, which is due to the nonreflective coating applied to the cells. One less common alternative to monocrystalline and polycrystalline is thin film solar cells. These are thin materials, usually glass or plastic, which semiconductor materials are injected into. The result is a flexible, low cost solar panel. In Figure 17, the polycrystalline solar cell is distinguishable because of the several pieces of silicon, opposed to the monocrystalline panel on the right which is a solid piece of silicon.

Thin film solar panels have a lower concentration of silicon, and because of this, they are less efficient. Thin film panels usually operate with 11% to 14% efficiency which is significantly less than the monocrystalline and polycrystalline competitors. The appeal of these solar panels is the low price point they are offered at; however, with the advances in solar panel manufacturing, the price gap has been closing and thin film panels now are only seen on very low power implementations of houses which a large amount of roof space.

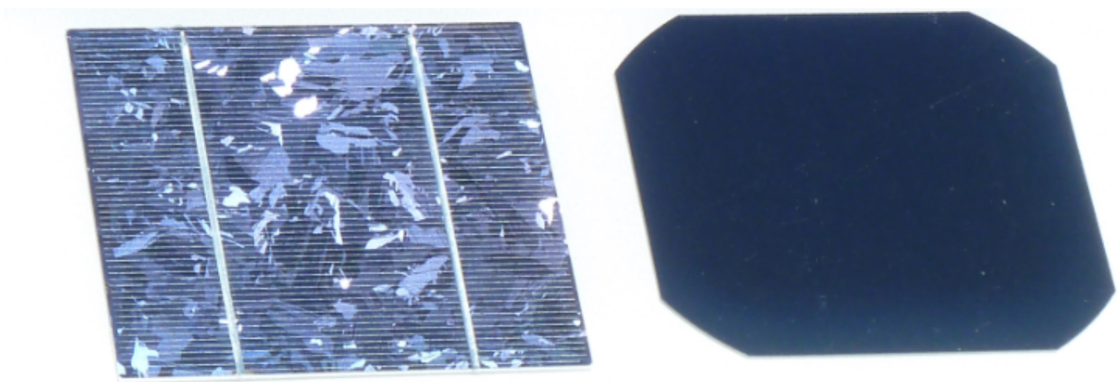


Figure 17: Polycrystalline and Monocrystalline solar cells (Public Domain)

The majority of solar cells are made of silicon because of the cost. When semiconductors are needed on a cost sensitive project, silicon is used unless the concern of speed drastically outweighs cost. The other solar panel option, which can be obtained at 1000 times the cost of a silicon solar panel, is gallium arsenide. Gallium arsenide has been around in the semiconductor industry longer than silicon but was quickly replaced once comparable silicon products were being produced at 1/1000 the cost. Gallium arsenide solar panels have been able to achieve 37% efficiency, compared to silicon's 22%. This is due to gallium arsenide's electron mobility. With a higher electron mobility, current can transfer faster through the semiconductor. This is the biggest benefit of gallium arsenide over silicon. Other benefits are gallium arsenide's effectiveness in low-light situations, its resilience to heat and moisture, and its flexibility. While these gains are substantial, gallium arsenide solar panels are not commonly sold on the consumer level. Gallium arsenide solar panels can be found on satellites, the Mars Rover, and military applications.

4.1.4.4 Solar Power Limitations

The issue that plagues solar powered devices is the same issue that power generation industries face when they attempt to apply solar power to the grid and other large-scale implementations. Solar panels take up a large amount of space compared to the amount of power generated. Compared to a gas turbine, a solar farm of the same output takes up a minimum of 10 times the space. When scaled to smaller applications, solar panels still struggle in providing enough electricity. In these cases, the limits of solar panels and their efficiencies must be taken into account.

4.1.5 Boost Converter (DC-to-DC amplifier)

To utilize the power that is being produced by the solar panel, the voltage of the solar panel must be greater than the voltage of the battery which is being charged. To do this, voltage must be stepped up or down to an appropriate magnitude. One way to achieve the desired voltage is by using a boost converter, illustrated in Figure 18, which is a DC-to-DC amplifier. This is a common technique to acquire the desired voltage in solar applications.

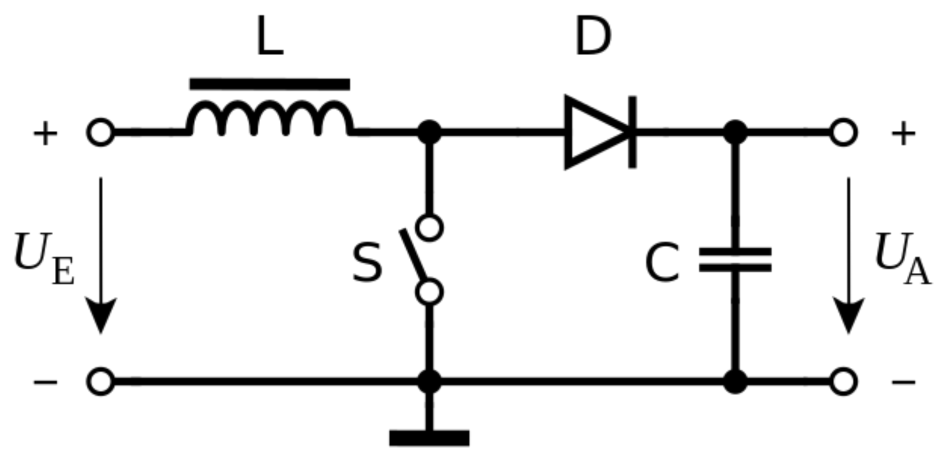


Figure 18: Boost Converter (Public Domain)

The main components of a boost converter are a transistor, inductor, and a diode. An inductor, which is the circuit element labeled “L” in Figure 18, is a passive circuit element which stores charge in the form of a magnetic field. As the strength of the magnetic field increases, the strength of the electric field also increases. Once the magnetic field stores enough flux and reaches a large enough magnitude, the inductor will allow a large enough current to pass through which will boost or step up the DC voltage using a MOSFET transistor. The MOSFET, which is a metal-oxide semiconductor field effect transistor represented by component “S” in Figure 18 acts as a switch which will increase the voltage and decrease the current. Once the amplified voltage has been reached, a diode is used as a switch to allow the current to bypass the MOSFET for further amplification. A diode is a voltage controlled, voltage device that is represented by component “D” in Figure 18; once a large enough cut-in voltage is reached, the diode acts as a switch, bridging both sides of its connection and allow for current to flow through it. In this case, the diode allows current to flow to the capacitor. The post amplified voltage is stored a capacitor, which is circuit element “C” in Figure 18. A capacitor is a device that stores electric charge in an insulator that is separated by two conductors. Once the stepped-up voltage has been stored in the capacitor, it will then stay there until it is added to the output voltage of the diode from the following cycle. The desired voltage amplification is commonly adjusted by using a potentiometer in series with the diode.

4.1.5.1 Boost Converter Efficiency

Boost converters are commonly seen in solar application, many of which are under tight power constraints due to the day to day inconsistency of solar energy available. Because of this, it is important to understand the power loss and the efficiency of charge converters. One source of loss in a charge converter comes from the conduction of current in the MOSFET. Because MOSFETs act as switches, a small amount of current is lost when passing through the transistor. Since boost converters increase voltages in steps, with each step being an iteration of the circuit, this power draw can add up to be a considerable total loss. Other factors of loss in boost converters are the resistance of both the MOSFET and inductor. Ultimately, most boost converters average above 90% efficiency depending on the gain needed per application. The larger the difference in input voltage vs. output voltage, the longer the current will have to utilize the MOSFET transistor to amplify the voltage. As the current is required to pass through the MOSFET, more loss is incurred which will lower the efficiency.

4.1.6 Battery

Wherever humanity goes, batteries seem to follow. The ability to create and store energy has given electronic devices the portability to be disconnected from the grid's 3-phase power and the reliability to have a bank of energy as an alternative source. By using a battery, devices can store energy and utilize it while it is not being produced. Storage of generated power is important for solar devices because solar panels can only create electricity under sunny conditions.

There are three main components of a battery: the cathode, the anode, and the electrolyte. The cathode is the positive terminal of the battery, while the anode is the negative terminal of the battery as seen in Figure 19. These two are separated by the electrolyte. The cathode and anode are each connected to dissimilar materials. This creates a difference in ions that the materials carry. Ions are molecules which have a positive or negative charge due to the gaining or loss of electrons. The anode is connected to a material which loses electrons in a process called oxidation. The cathode is connected to a material which gains electrons in the process of reduction. This is due to the high concentration of energy in the anode vs. the low concentration of energy in the cathode. The electrolyte serves to divide these two materials with a physical barrier and prevents a constant flow of electrons. By connecting the cathode and the anode, it creates a bridge which the electrons can pass through. The electrons will pass from areas of high concentration in the anode to the cathode which has a low concentration. The combination of oxidation and recombination is referred to as the Redox reaction and this will create a current value from the battery.

The voltage of a battery is dependent on its chemical makeup. Electrochemical reactions between the chemicals will decide the potential difference between the two materials. Having a larger quantity of the materials will result in a larger potential amperage. To obtain the correct voltage necessary, many devices will multiple batteries in series. Batteries such as the household AA and AAA are sold per cell, which is why devices often need several of them to operate. Other batteries have several cells packaged into one battery, such as the common household, rectangular 9v battery. When deciding on a battery to power an electronic device, the material used must be considered. When using a DC voltage source, the battery cell material will decide the voltage the battery will have. For example, when using 3.7v Li-ion battery cells, three can be connected in series to receive 11.1v, or 4 can be connected in series to receive 14.7v. To achieve 12v output out of these, the voltage would have to be amplified or stepped down, both of which will incur loss in the system.

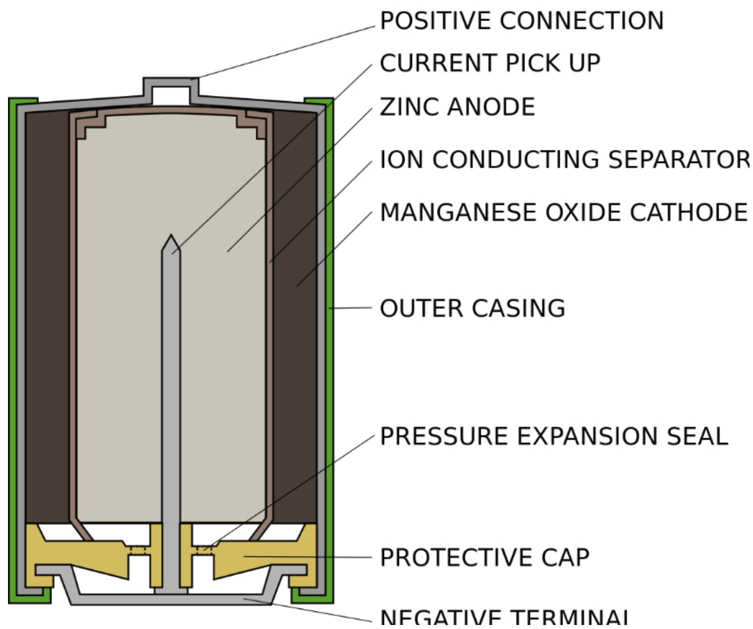


Figure 19: Anatomy of a battery (Public Domain)

4.1.6.1 Environmental Impact

In the field of solar power generation, the environmental impact and reduction of humanity's carbon footprint is an ethical dilemma when choosing to go off the grid. Avoiding using the grid's resources to power electronic devices will ultimately save power and use less nonrenewable resources. However, the production of large chemical batteries and their environmental impact strikes a debate on how green solar power is. For example, the sulphuric acid and lead in lead acid batteries is toxic and must be disposed correctly. Sulphuric acid is corrosive and lead is toxic, so exposure to living organisms puts them in danger. Toxic waste is just one-way batteries can be harmful to the environment.

While toxic waste is a large concern when producing finite life batteries on a large scale, the impact of recourse mining can also disturb habitats with valuable resources. Li-ion batteries, for example are seen as "nontoxic waste" by the government; however, the lithium for these batteries are mined by drilling into salt flats and injecting mineral rich brine into the surface. The minerals are left in these salt mines where water will evaporate and leave behind a concentrate of lithium which can be filtered out and used in Li-ion batteries. This disruption of marine ecosystems has been known to create an environmental imbalance. From this procedure, schools of dead fish and other marine wildlife have been found.

When deciding on a battery for an electronic device, it is impossible to find one that does not harm the environment. Instead, engineers have to settle on the least environmentally taxing. Whether it is production or disposal of batteries, the environmental impact must be considered when creating a device that may be marketed towards a "green" clientele.

4.1.6.2 Rechargeable Batteries

Solar panels can be used to provide current to a rechargeable battery which will "charge" it with energy. While rechargeable batteries can accept and output energy in the form of electrical current, they do not store electrical energy. Instead, batteries store chemical energy which can be converted back into electrical energy. Once the chemical energy of a battery is expended, rechargeable batteries have the capability to recharge to regain any lost or used chemical energy.

For non-rechargeable batteries, or primary cells, this chemical process can only go one way; the battery can only convert chemical energy into electrical energy. In rechargeable batteries, or secondary cells, electrical energy applied in the opposite direction of the battery to create additional chemical energy. When a rechargeable battery is connected to a power source or charger, it forces electrons to return from to the cathode back into the anode. This reversed movement of electrons can be seen in Figure 20. This cycle cannot go on forever; different materials can undo the chemical changes more efficiently than other. The amount of recharge cycles in a battery range from hundreds to thousands depending on the material and manufacturer. Once the electrons are back in the anode, the battery can be reused for additional cycles. Unfortunately, applying a charger to a primary cell will not yield the same results; because of the materials which primary cells are made out from, the battery is permanently damaged once electrons travel from anode to cathode. This means that they are not eligible to be safely and efficiently recharged.

Rechargeable batteries can be charged and discharged hundreds to thousands of times depending on the material they are made of. However, because the transfer of energy is not perfect, there will be loss after several cycles of discharge and recharge. Ultimately, these batteries can be recharged over-and-over until they are no longer able to power the devices they are being used for.

Batteries are made of a variety of materials, each with benefits and drawbacks. Some pivotal factors when deciding on a battery are voltage, amp hours, rechargeability, price, and durability to a variety of weather conditions. While the performance of different batteries may drastically vary due to size, material, and manufacturer, all batteries operate from the same conceptual properties.

4.1.6.2.1 Nickel Cadmium Batteries

Nickel Cadmium (NiCD) batteries are commonly used in small electronic devices due to their low resistance and high current values. These batteries do well in a large range of temperatures and can operate between -4°F and 140°F. This tolerance makes these batteries a practical option for outdoor applications where the weather is more consistent. NiCD batteries are secondary cell batteries that can be recharged several times at a low charging efficiency. NiCD batteries only convert around 75% of the energy given to it back into electrical energy. While these batteries can be discharged and recharged several times, NiCD batteries suffer from the memory effect. The memory effect occurs in batteries that require periodic full discharge to prolong their use. If NiCD batteries are recharged before being fully used, it will waste charge cycles and wear out the battery quicker. NiCD cells only have an electric potential of 1.2V. To obtain a total voltage of 12V, it requires 10 batteries in series. Batteries made from other materials have a larger potential difference than NiCD, which is why NiCD batteries are not used in as many higher voltage applications. Due to the low voltage of NiCD batteries and the complications in charging, NiCD batteries are not a popular choice for solar storage applications.

4.1.6.2.2 Lithium Ion Batteries

Lithium-Ion (Li-ion) is a popular secondary cell battery that is used in small electronics due to amount of energy they can produce, compared to their small cell size. Unlike NiCD batteries, Li-ion batteries do not suffer from the memory effect; therefore, these batteries do not have to be fully discharged before recharging them. Because of this, Li-ion batteries are effective in solar applications.

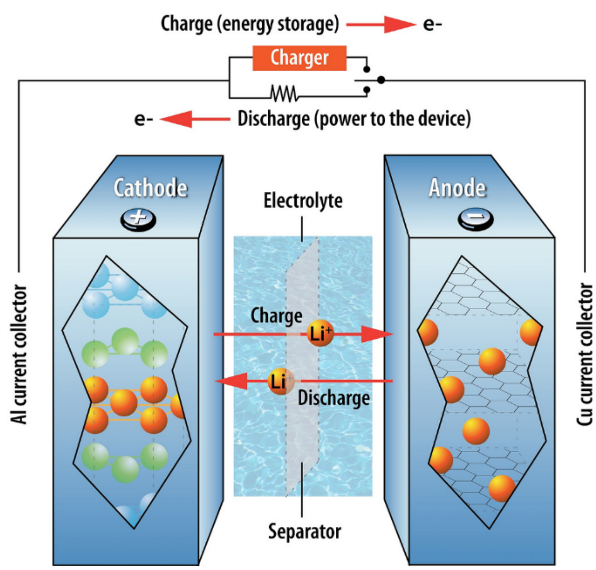


Figure 20: Redox Reaction of a Li-ion battery (Public Domain)

Input and output voltages must be closely monitored with Li-ion batteries because Li-ion batteries suffer from issues stemming from overcharging. Not only could this hurt a lithium ion battery's charge cycle count, but it could also contribute heat. To prevent this, lithium ion batteries have circuit protection, to stop these batteries from overcharging. Some common precautions to avoid issues due to overcharging are creating a circuit that does not allow too much current to pass through the device and mitigating excess current to a ground source. Additionally, some battery protection devices add modular vents to release any built-up gas. If a lithium ion battery is exposed to too much heat, they can experience thermal runaway. Thermal runaway occurs when a battery is heated to a point where it causes further increase of temperature due to a chemical reaction. Many cases of Li-ion thermal runaway occur from a short between the cathode and anode. Because the electrolyte has been engineered to be thinner, some manufactured lithium ion batteries have experienced failures in the electrolyte. This allows for the cathode and anode to quickly exchange electrons, expending all of the chemical energy into heat. This is the platform which catalysts thermal runaway. In thermal runaway, the device is not able to cool down enough to compensate for the quick change in temperature. Li-ion batteries have been observed to melt or explode, as seen by the Samsung's infamous Galaxy Note 7. Li-ion batteries can be applied to solar applications, but in environments where the battery may get warm, the possibility of the battery experiencing thermal runaway must be considered.

4.1.6.2.3 Lead Acid Batteries

The tried and true medium for storing energy created from solar panels is a lead acid battery. Lead acid batteries were the first rechargeable batteries on the market and are still used today due to price and consistency. Although more efficient batteries have been created since the implementation of lead acid batteries in solar, lead acid batteries are still used because of their middle ground between reliability and cost. Lead acid provides the lowest cost-to-watt relationship which is why they are popular among self-sustainability and off-grid projects. Compared to other secondary cell batteries, lead acid has a low amount of charge/discharge cycles. Because of this, in most cases using a lead acid battery over a Li-ion or NiCD would require a more regular replacement of batteries. Charging a lead acid battery is simple; however, it takes a long duration of time compared to more modern secondary cells. While it does not require extra circuit protect, and is not as prone to thermal runaway, lead acid batteries can take over 12 hours to fully charge. Lead acid batteries charge to around 75% of their capacity at a consistent rate, but then start to trickle charge. In a solar application, this may be problematic, because energy obtained by the solar cells are not available at all times of the day. If the battery does not accept all the current being produced from the solar cells, it will be wasted, which lowers the efficiency of the system. Figure 21, below, shows the rate of charge in a lead acid battery.

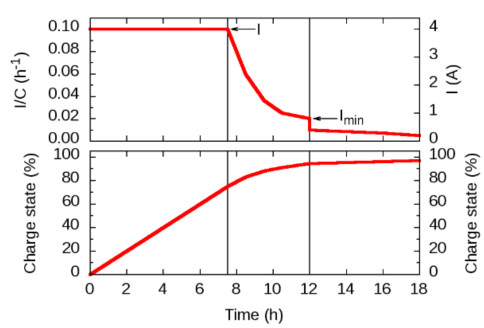


Figure 21: Lead Acid Rate of Charge (Public Domain)

For the first 75% of the battery recharge, the battery charges linearly, at an already slow rate. For the next 15% of charging capacity, the battery slowly accepts current at an exponentially decreasing rate. The battery finally crawls to 100% after several additional hours. If the battery is required to hold a large current value, lead acid batteries may not be a fitting selection.

There are a variety of batteries why fall under the lead acid category because of their chemical makeup. While each of these share traits, each battery has been engineered to prosper in different applications. Flooded lead acid batteries are filled with an electrolyte fluid where the cell plates are suspended in. This provides a high density of electrolytes for a large current value. The trade off to this is the potential for spilling the fluid out of the battery. Sealed lead acid batteries exist for this reason; however, the additional packaging occupies space and there is not as much volume for the electrolyte fluid. Gel batteries function similarly to flooded batteries, but the electrolyte solvent is more solid, gelatinous material. The production of these batteries can make them twice the cost of a traditional flooded battery. In Table 2, the three common batteries for solar storage are compared.

Table 2: Comparison of Battery Materials

Battery Type	Lead Acid	Nickel Cadmium	Lithium Ion
Price	$	$$	$$$
Cell voltage	2.0V	1.2V	3.7V
Life Cycles	200-300	1000	500-1000
Wh/kg	40	50	140
Charge Time (h)	8-16	1	2-4
Temp Range (F)	-4° - 122°	-4° - 140°	-4° - 149°

All batteries have trade-offs which must be considered when deciding on an energy storage device. With more appealing traits and characteristics comes with higher price points. Lithium ion batteries numerically are the most dominant; however, the issue of thermal runaway and requirement of additional integrated circuitry is enough to disinterest both consumers and engineers.

4.1.6.3 Battery Charging

Charging procedures are going to vary between battery material. Due to different materials, one battery may charge more efficiently due to their ability to create a reversal of chemical energy. Lead acid batteries will suffer from a loss in charge cycles and efficiency if they are not fully charged occasionally; however, partially charging the battery does not create any damage to the battery. Nickel Cadmium batteries do not need to be fully charged and can constantly be at states of partial charge without creating any damage. While several partial charges far apart do not cause any damage, several repeated partial charges in close succession can lead to a heat buildup which will both damage the battery and waste energy. Lithium ion batteries like to be partially charged; if Li-ion batteries can avoid fully charging, then they can avoid running a thermal runaway safety circuit which trickle charges the battery. With lithium ion batteries, partial charge does not degrade the cell.

Knowing how batteries handle inconsistent flows of charge is important when dealing with solar applications. Since sunlight is both intermittent and inconsistent, batteries may be constantly thrown between states of charging and not charging. When charging, batteries can fluctuate in temperature due to the chemical changes occurring inside. In a lead acid battery, once it has started to trickle charge, the battery will get warm. Since our solar device is going to be outside, this would ensure that the battery would not get too warm while charging in the Florida sun. Nickel Cadmium batteries get warm to the touch while charging and have to cool down to effectively draw power. This is problematic in a self-sustained solar project, as there is no other way to reliably provide electricity. Lithium ion batteries are not efficient in warmer weather conditions and must stay cool while being charged. These batteries will usually get slightly warm to the touch upon charging.

4.1.7 Charge Controller

Charge controllers are essential to solar powered projects to provide consistent voltage to the device. This is accomplished by applying the correct voltage to the battery. Solar panels, as stated earlier, provide inconsistent amounts of charge. This could be due to available sunlight, any obstructions, or glare that may reflect light hitting the surface. Because of this, having a medium between the solar panel and the battery ensures that the battery is getting provided a constant voltage. Using a solar charge controller gives solar powered devices reliability and sustainability. These devices take out inconsistencies in battery charge and are used as other renewable energy medians and seen in wind and hydroelectric power.

Charge controllers provide additional protection against reverse current. At night, solar panels can pass current in the opposite direction which will drain the battery. This draw is usually minor; however, this will cut down the efficiency and slowly drain the battery. In most charge controllers, engineers implement a transistor to resist current from going in the opposite direction. The transistor acts as a switch and it will only allow current to go from the positive terminal to the negative terminal.

4.1.7.1 Overcharging

Overcharging batteries can be detrimental to the device using the battery as well as the battery itself. Once a battery is fully charged, it cannot accept any additional energy. If more energy is applied to it, it will result in an increase in voltage of the battery. A larger voltage can create damage to the circuit and it will create damage to the battery due to overcharging. The charge controller acts as a hub between the battery, the solar panel, and the device. Because the charge controller is an interim step, it is able to modulate the current going to the battery. Once the battery is at capacity, the charge converter stops accepting charge from the solar panel. By not allowing the ends of the solar panels to make a connection, no current will flow; this will stop charging the battery.

4.1.7.2 Low Voltage Disconnect

Many secondary cell batteries will lose charge cycles if they are fully drained, or even drained past 50%. A secondary function of many charge controllers is a low voltage disconnect. The charge controller is monitoring the voltage of the battery, ensuring it does not drop to a minimum voltage level. If it does, it can disconnect the connection between the battery and the device. This can save the battery from any permanent damage.

While solar devices should not allow the battery to get to such a low voltage, the low voltage disconnect acts as a fail-safe. If the batteries being used are expensive or hard to obtain, this is a necessary safety precaution. Although disconnected from the device, the battery is still connected to the solar panel. Once the battery reaches an appropriate voltage, the charge controller will turn the device back on automatically.

4.1.7.3 Types of Charge Controllers

There are three main classifications for charge controllers currently available. When compared against each other, one charge controller is undoubtedly a better product. The Maximum Power Point Tracking charge (MPPT) charge controller is more efficient in storing energy; however, this optimization comes at a cost. MPPT charge controllers were created following Pulse Width Modulator (PWM) charge controllers, which lack efficiency. MPPT charge controllers are more expensive, partially due to their recent emergence into the market.

4.1.7.3.1 Single Stage Charge Controllers

Single stage charge controllers are the most rudimentary implementation to safely charge a battery without overcharging. These work as an on-off switch that either allows full flow of current or no flow off current. A problem with these single stage charge controllers is efficiency when charging the battery to its max chemical potential. Many secondary cell batteries used in solar applications require a slower trickle charge once they are 70-80% of their max capacity. Single stage charge controllers are the cheapest option to provide a safe solar power supply; however, they are not commonly used due to their effectiveness in charging and their tendency to degrade charge cycle capacity in batteries.

4.1.7.3.2 PWM Charge Controllers

Pulse width modulators are a budget friendly solution to having a charge controller in a solar device. PWM charge controllers build upon single stage charge controllers with an additional stage of charging. These charge controllers are a popular solution for solar powered devices due to their balance of budget and effectiveness.

PWM charge controllers will switch on once the battery drops to a minimum threshold. Once that threshold is reached, the charge controller will supply the maximum current the battery can accept until a safe maximum threshold is met. After this, the PWM slowly decrease the charge it provides to the battery until the battery is fully charged. The closer the battery gets to a full charge, the less current is provided. Slowly decreasing the charge given to the battery will provide less strain on the battery and allow it to stay at a constant maximum voltage when provided with additional current. Alternatively, in a single cycle system, the charge controller would wait until a threshold is met to allow for current to pass through.

4.1.7.3.3 MPPT Charge Controllers

The world saw the first MPPT charge controller in the 1980s, about 20 years after PWM charge controllers entered the market. These controllers are the most advanced due to their flexibility in solar panels to input. With other PWM charge controllers as well as single stage charge controllers, excess voltage will be lost upon supply to the battery. With an MPPT charge controller, excess voltage can be converted into additional current. Using MPPT charge controllers will allow for larger solar panels which may be to maximize usable solar panel space.

While the ability to convert excess voltage to current is a useful feature, it is not necessary for every environment. If the solar panels being used do not exceed the voltage of the battery, this will not be any more effective than a PWM charge controller. MPPT charge controllers can be four to five times the price of a PWM charge controller which is why they are not as widely used in every solar application.

4.1.8 Traffic Light Signal

The Situation-Aware Stop Signal uses red and yellow lights to control the flow traffic. Currently, other implementations of traffic lights use incandescent bulbs and LED bulbs. The signal displays multiple solid red lights which is used to stop traffic moving in that direction. Flashing yellow lights display the red one to tell drivers they can yield the intersection and proceed with caution.

4.1.8.1 Incandescent Light Bulbs

Incandescent bulbs are the most inefficient sources of electrically generated light currently still available on the market. Their persevered popularity is largely due to their price and reliability. Incandescent bulbs function by passing a large current value through a coil of tungsten wire which produces enough heat to create light. This process is neither efficient in creating heat nor light, and the majority of the energy is wasted using incandescent bulbs.

Despite the inefficiency of incandescent bulbs, they are still implemented throughout the US. Because traffic lights are connected to the grid, they are able to access as much power as necessary, so using an incandescent will not improve reliability. Incandescent bulbs can use either AC or DC power, so it eliminates any conversion of electricity from the grid's three phase system. They offer a low price point to implementation. In areas of large snowfall, incandescent bulbs prevent from buildup of snow on the traffic light lens. Because incandescent bulbs produce heat, they are able to melt any snow which would disrupt the visibility. With implementations of LED arrays in place of incandescent bulbs, areas that experience heavy snow storms have had buildup of snow on the traffic lens since they do not produce enough heat.

4.1.8.2 Compact Fluorescent Lamps

Compact fluorescent lamps (CFL) are both cost effective and efficient. These bulbs are almost as efficient as LED lights but are able to be obtained at a fraction of the cost. CFL bulbs are illuminated due to a dominantly chemical reaction. In a CFL bulb, a catalyst electric current is provided to initialize the operation of the light bulb. The bulb is filled with argon gas and mercury vapor and once current is passed through it, it will create ultraviolet light. The ultraviolet light will then illuminate the fluorescent coating along the sides of the bulb.

Unlike other forms of light production, this chemically driven illumination results in a slow initialization. Once CFL bulbs have been allowed to heat up, they function efficiently; however, they require a significant amount of power to start up. CFL bulbs are driven by a chemical reaction to ultimately produce light. Because of this, they are sensitive to outdoor environmental conditions. Specifically, CFL bulbs are not stable in high temperature applications. Most CFL bulbs not rated to withstand temperatures over 120°F. While the ambient temperature outside may not reach this on a day-to-day basis, it is achievable in a black box equipped with limited ventilation.

While CFL bulbs are a practical in a household application, their resistance to outdoor weather and their inability to react quickly make them an unattractive option for outdoor devices. Traffic lights need to be able to withstand harsh weather conditions in order to provide reliability. This may be an issue when using CFL light bulbs in an outdoor situation. CFL bulbs are currently not implemented in traffic lights, likely due to the previously stated issues.

4.1.8.3 Light Emitting Diode

Light emitting diodes (LED) are an energy efficient replacement to incandescent bulbs. LEDs are semiconductor devices that allows for flow of electrons or holes from an N-type region to a P-type region. Current is passed from the anode, through the semiconductor die, and finally to the cathode as seen in Figure 22. As electrons pass through the P-N junction of the semiconductor and recombine, energy is released in the form of a photon which will produce light. This will only occur once the diode has a large enough potential difference across it that is greater than the cut-in voltage. When a material creates light via applying an electrical current across it is referred to Electroluminescence.

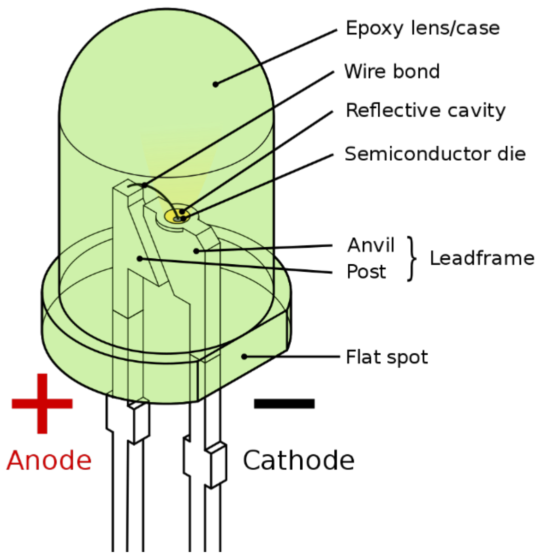


Figure 22: Diagram of a Light Emitting Diode (Public Domain)

The efficiency produced from an LED light is unmatched by any other bulb type, which is why most devices are incorporating LEDs if they are able to be included financially. Table 3 shows a comparison of the three most common light sources and their efficiencies. LED lights can be 5-6 times more efficient than the commonly used incandescent bulbs. Large LED manufacturers such as GE or Phillips will even outperform the average LED, producing over 100 lumens per watt. In devices that are constantly on and drawing power, it is important to have the highest possible efficiency. This will eliminate a large portion of operating fees for an intersection. According to WSDOT, a traffic light on average will cost $8,000 per year in just maintenance and electricity. By using energy efficient LEDs, the cost per year could significantly decreased.

Table 3: Comparison of Bulb Types

Bulb Style	Price	Lumen/Watt	Lifespan (h)
Incandescent	$	14.3	5,000
CFL	$$	57.14	10,000
LED	$$$	80	50,000-100,000

4.1.8.3.1 LED Lifespan

In order for traffic lights to be useful tools for preventing accidents and migrating traffic, they must have the utmost reliability. Upon the failure of any subsystem of the traffic light, an accident could occur. Because of this, it is important to understand the lifespan of every component being used in the device. LED lights outlive any other electric form of lighting on the market. This is due to their lack of heat production and low voltage requirements. LEDs will last between 50,000 hours, to upwards of 100,000 hours. Incandescent bulbs last only a fraction of that time with a 5,000-hour lifespan. Using LED lights in a traffic signal allows for less maintenance and added reliability in traffic system. The added lifespan can prevent against outages and downtime in traffic systems.

4.1.8.3.2 LED Brightness

In order for traffic lights to be useful tools for preventing accidents and migrating traffic, they must have the utmost reliability. Upon the failure of any subsystem of the traffic light, an accident could occur. Because of this, it is important to understand the lifespan of every component being used in the device. LED lights outlive any other electric form of lighting on the market. This is due to their lack of heat production and low voltage requirements. LEDs will last between 50,000 hours, to upwards of 100,000 hours. Incandescent bulbs last only a fraction of that time with a 5,000-hour lifespan. Using LED lights in a traffic signal allows for less maintenance and added reliability in traffic system. The added lifespan can prevent against outages and downtime in traffic systems.

To meet criteria given by the USDOT, there are a few options in obtaining brighter LED outputs. An LED will turn on once it has reached a large enough cut-in voltage; however, they will get brighter as more current is passed through the device. To accomplish this, a smaller resistor could be added in series with the resistor. This, however, will have its limits. If too large of a current is applied to the LED, it can cause the device to fail due to overheating. Furthermore, the brightness of an LED will start to taper once it has reached its limits. Another alternative to address a battery that is not getting enough current delivered is to supply the diode with a larger capacity battery.

Multiple LEDs can be integrated into a circuit to produce the desired candela output. By applying multiple LEDs in series, a larger light output can be achieved without diminishing the current going through the LED. When adding multiple LEDs in series, the circuit is limited to the voltage of the battery. Batteries tend to get exponentially more expensive as they rise in voltage, so as a compromise, LEDs would have to be arranged in a combination of series and parallel. Doing so, large LED arrays can be structured.

4.1.8.3.3 LED Colors

The color of an LED can be changed by increasing or decreasing the bandgap voltage of a semiconductor. Visible light lies on a spectrum between 1.6eV and 3.3eV. As seen in Figure 23, hues of red will be on the lower end of the eV spectrum while shades of blue and purple will be on the higher end of the electron spectrum. By manipulating the bandgap of the LED, the color can be changed. To achieve this, various materials are used to span across the color spectrum. Most of the visible colors are achieved by using diodes made from aluminum indium gallium phosphide (AlInGaP) or indium gallium nitride (InGaN). The colors being used in this project will be red and yellow to abide by existing traffic laws. These colors are on the lower end of the spectrum and will require a lower voltage to turn on.

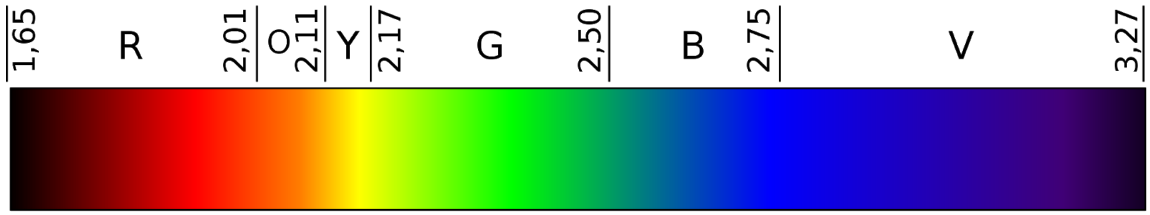


Figure 23: Color LED Bandgap Spectrum (Public Domain)

4.1.8.3.4 Environmental Impact

LED lights appeal to the environmentally conscious consumer due to its absence of natural gasses and mercury vapor which are used within CFLs. LEDs, unlike many incandescent bulbs and CFLs can be disposed without any additional precaution taken. This reduces any damage that could be done to the areas around waste management facilities. Often, proper recycling procedures are not taken with disposal of light bulbs. Since this is not a concern with LED bulbs, they create less of a strain on the environment.

4.1.8.3.5 LED Cost

The benefits that LED lights offer come at a premium price. These lights are much more expensive than both incandescent bulbs and CFLs. While LEDs are widely supported as the most efficient electrical light source, applying these lights on a large scale can get costly. LED lights can cost 5-8 times more than the traditional incandescent bulb.

LED lights have been dropping in cost in the past decade and have now become more and more affordable. LED lights were largely expensive due to the processes needed to create them. As manufacturing equipment improved and the demand for LED lights increased, the price has begun to drop.

4.2 Software

When dealing with any system, software is a huge factor. The first most crucial component of the software is the operating system. While some devices do not need an operating system, others may require one to guarantee that every component of the software acts on time and with a hard deadline. Then the next crucial component is how the system itself is designed. From design paradigm to language, to development methodology, each component determines how the software will be designed and how it will affect the system.

4.2.1 Operating Systems

A crucial consideration for this project is the platform upon which embedded code can run. For many embedded systems, it may be sufficient to load a single program onto the microcontroller and run it in a loop. Other systems require multiple programs, or processes, to run concurrently – or what may appear as concurrent to the user. This ability may be referred to as “multitasking.” Such multitasking requires a more sophisticated approach to loading and running programs. This problem is addressed by operating systems.

Operating systems schedule processes according to a certain set of rules. This rule-keeping is the basis upon which algorithms are written and implemented to allow processes to share time (what is called timesharing). Algorithms may be written to schedule processes according to duration, equal allotment of time, or priority, amongst other possible qualifications. Shortest Job First, Round-Robin, and Preemptive Priority are examples of these different algorithms used for the scheduling of processes, respectively.

For the Situation-Aware Stop Signal, it was very apparent that an operating system was needed due to the variety of threads required to monitor traffic in each direction of the intersection and to present warnings accordingly. It was also easily acknowledged that in a safety-critical application as in that of the stop signal, timing was of significant importance. Reaction time of the system had to be minimal and near-instant to be able to communicate with drivers in sufficient time. This introduced a need for a specific kind of operating system – one finely-tuned and capable of quick response to input.

4.2.1.1 Conventional Operating Systems vs. Real-time Operating Systems

Operating systems such as Windows, macOS, and Linux represent only a narrow selection of the great breadth of operating systems that run the devices in the world around us. While these operating systems seem to fit many needs, there are many cases for which these are not suitable. Such is the case in the world of embedded devices, whose small footprint, limited hardware, and real-time application require a different category of operating system. For these kinds of devices, a real-time operating system (RTOS) is more appropriate. A brief comparison of these two classes of operating system, conventional and real-time, will help narrow the focus of this segment of research.

Conventional operating systems such as Windows, macOS, and Linux are complex, heavyweight, and unsuited for real-time data processing in safety-critical systems. One might imagine how detrimental it would be to have a “blue screen of death” on a computer that is running important systems in a car or in an airplane. Due to the nature of their creation, conventional operating systems are not well-protected against the fatal errors common to the ways by which they schedule processes. Such errors can cause system hang up and significantly delayed response to input. In the end, this means that conventional operating systems are not sufficiently reliable when high reliability may very well be needed.

Real-time operating systems, on the other hand, are operating systems that are designed and built to respond to inputs extremely quickly and with a very high level of reliability. They are complex but lightweight, and they are finely-tuned such as to be able to change execution with little delay when a higher priority process is presented. While both real-time operating systems and conventional operating systems operate on a system of priorities, the former allows a guaranteed and predictable response for the preempting of lower-priority processes in favor of higher-priority processes. The ability to assure this kind of response characterizes real-time operating systems and sets them apart from their counterparts.

Because of their greater reliability and smaller footprints, real-time operating systems are most often the preferred choice for products whose timing of operation is critical or otherwise whose operation must guarantee human safety. Because the Situation-Aware Stop Signal is of this class of product, the remaining research on operating systems found in this section is focused on those that may be classified as real-time. Here, an overview of the attributes of these operating systems will first be given. Later, different RTOS options will be explored.

4.2.1.2 Attributes of Real-time Operating Systems

Because of their time-sensitive applications, real-time operating systems must perform quickly and with a high degree of reliability. They are intricately designed to provide both consistent and repeatable computation based on external and internal factors. They are also tailored to the hardware upon which they run. To ensure success of the missions with which they are tasked, real-time operating systems are built with considerations made in this regard. These considerations may be embodied in the following attributes:

1. Deterministic execution
2. Explicit definition of priority
3. Scalability
4. Compatibility
5. Power efficiency

Each of these attributes reveals important insight into the inner-workings of real-time operating systems and also provides a basis upon which comparisons may be made. While these characteristics are largely shared between operating systems of this type, implementations do vary. For further explanation of these attributes, an overview is provided below.

4.2.1.2.1 Deterministic Execution

Deterministic execution means guaranteed and unvarying execution of processes. The value behind real-time operating systems lies in their ability to provide a predictable sequence of execution. This suggests that both the scheduling and the completion of a high-priority process must be non-probabilistic, wherein a developer may assign the high priority and rightfully expect the scheduler to schedule the process in a defined manner and duration of time. If a real-time operating system is able to do this consistently, without prospect of failure, the execution of processes may be classified as deterministic. Figure 24 illustrates how a real-time operating system's scheduler ensures deterministic execution.

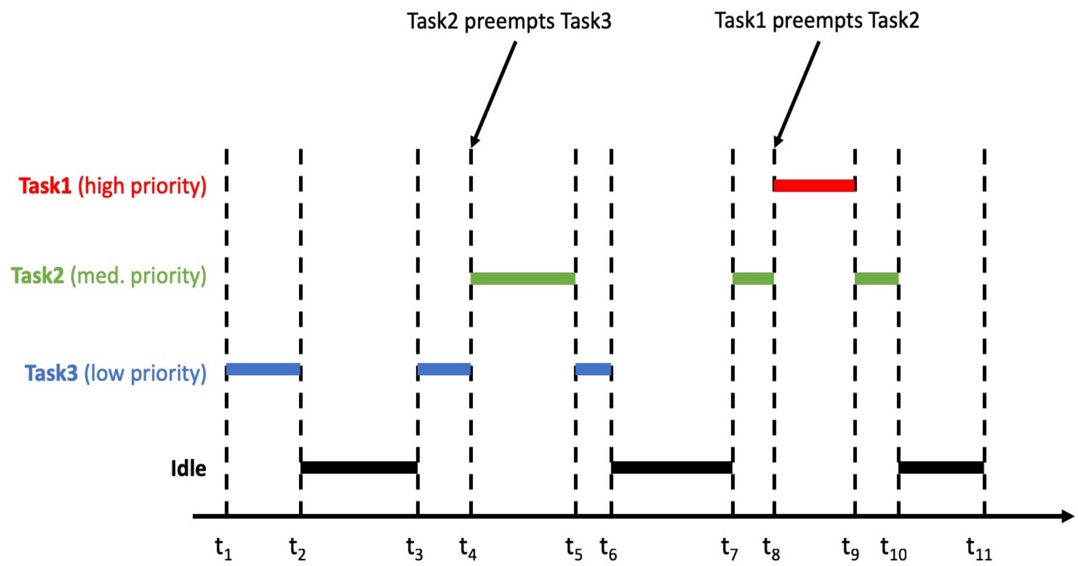


Figure 24: RTOS scheduling

4.2.1.2.2 Explicit Definition of Priority

Explicit definition of priority allows the developer to define specific degrees of priority and assign them to processes. The operating system then handles these priorities with designated behavior. The ability to explicitly define priorities allows the operating system to understand what is of critical importance and what is of lesser importance. It is able to clearly understand the correct way to schedule without having a need to make wrongful assumptions. Such assignment of priority is inherent of the development stage of the software pertaining to a project.

4.2.1.2.3 Scalability

Scalability refers to the ability of a real-time operating system to scale in executable size and in functionality to an application. Applications vary in their requirements, especially with regards to memory constraints, computation, and communication. Many real-time applications require a small memory footprint with little features other than the real-time scheduling of processes. Other applications, such as those characteristic of the Internet of Things, require complete networking stacks and other resource-demanding software components. With scalable real-time operating systems, the kernel may be packaged with only the necessary components for the application – be it few or many.

4.2.1.2.4 Compatibility

Compatibility suggests the ability of a real-time operating system to function on multiple platforms, or architectures. When a developer is tasked with the creation of a real-time system, hardware is considered as much as the software that it will run. To allow for flexibility in hardware design choice, a real-time operating system is often designed to support a variety of architectures. This may be seen as a high preference to developers who may be yet considering microcontrollers using different instruction set architectures such as ARM Cortex, MIPS, and Intel x86, amongst others. Typically, real-time operating systems are offered in different ports or otherwise implement a hardware abstraction layer (HAL).

4.2.1.2.5 Power Efficiency

Power efficiency is a focus of nearly every real-time operating system. Because many applications impose tight constraints on power, real-time operating systems have a need to operate efficiently and minimize unnecessary computation or execution. To do so, they may support low-power modes inherent of the hardware, as well as implement clever strategies to reduce their own management of the system. A common feature used to reduce power consumption in real-time operating systems is called a "tickless idle." By maintaining a tickless idle, an operating system refrains from using the MCU and checking in at small intervals called "ticks" during periods of complete idle. Figure 25 presents the difference between systems that do and do not implement a tickless idle.

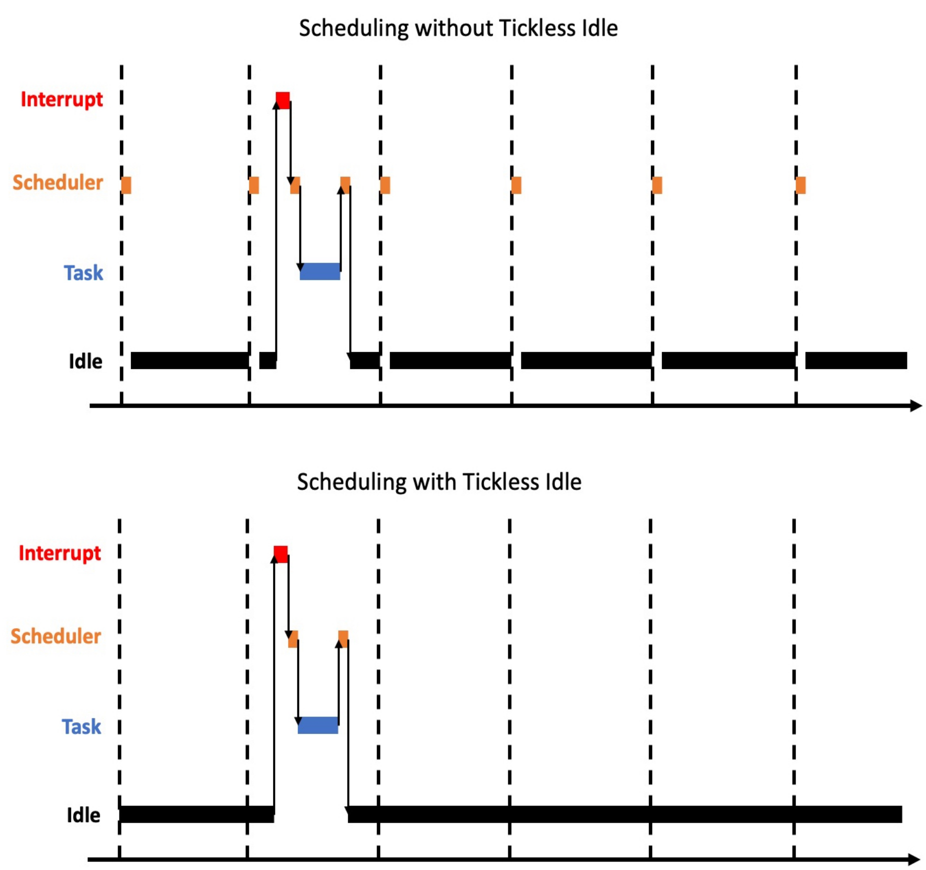


Figure 25: Effects of a tickless idle

4.2.1.3 Soft Real-time vs. Hard Real-time

Real-time operating systems may implement two kinds of real-time scheduling: soft real-time and hard real-time. Soft real-time means giving a best attempt to complete a process by a certain specified deadline. Hard real-time means guaranteeing the completion of a process by a certain specified deadline. In hard real-time, not completing a process by its deadline is considered total system failure. The selection of one of these two scheduling types depends largely on the nature of the application and the definition of system success.

Systems that implement soft real-time scheduling may strive to execute processes in a live setting but can afford occasional misses of deadlines or errors in execution. These systems include those facilitating live broadcasts and streaming of video, cellular voice communication, and weather sensing. While timely execution is desired, no critical threat is determined when infrequent mishaps occur. Often, these mishaps go unnoticed in this kind of system.

Contrarily, systems that implement hard real-time scheduling are absolutely time-critical. If a process is not executed in time, damage to the system may be incurred, mission failure may be imminent, or human lives may be at stake. Examples of this type of system include car safety systems, aircraft fly-by-wire control systems, and anti-missile defense systems. All of these present serious repercussions if deadlines are not met.

4.2.1.4 Real-time Operating System Options

The market for real-time operating systems has grown exponentially since the early 1980s. Today, many are offered that aim to meet the demands of systems both old and new. As has grown the variety of applications for embedded devices, the way real-time operating systems are developed and distributed has evolved. Where there was once a very limited array of commercial distributions, there is now an ever-growing number of options that are offered both commercially and as open source.

For the purposes of this project, an emphasis has been placed on real-time operating systems that are open source. The argument for an open source real-time operating system is much the same as the argument for open source software. Apart from a free price tag and a generous license, highly desired of a project of this type, an open source OS is one that is under the inspection of many. Many eyes are watchful of its code and many hands participate in its development. For a combination of these reasons, investigation has remained focused on open source options.

Research has yielded a select number of open source real-time operating systems that may be considered current and relevant to this project's needs. Four candidates chosen for further investigation are:

- FreeRTOS
- Zephyr™
- eCos
- TI-RTOS

4.2.1.4.1 FreeRTOS

FreeRTOS is an open source real-time operating system initially developed by Real Time Engineers Ltd. and handed over to Amazon Web Services in 2017. Its development began in 2003 with the work of its creator Richard Barry. FreeRTOS has now seen over 15 years of development and refinement, along with increased adoption amongst embedded software developers. It has come in at the top of its class in all EETimes Embedded Markets surveys since 2011 as either the most used or most considered real-time operating system. It is offered under the MIT license.

The operating system prizes itself in being a lightweight yet feature-rich platform. The source for FreeRTOS is contained in 3 C files, with an additional microcontroller-specific file needed for porting. The kernel binary averages between 6 and 12 KB, while a certain port reduces this number to less than 4 KB. The kernel schedules processes based on priority and does so by checking in at regular intervals called ticks. To save power, tick suppression, or a tickless idle, is implemented. As described in the overview of attributes of real-time operating systems, this feature may be an important consideration for projects requiring low power consumption.

FreeRTOS is fully-supported and documented, with an option provided to developers to upgrade to a commercial license for additional support and legal assurance. Its owners claim this to lower project risks as well as the total cost of ownership. While most developers may choose to remain with the free software and support made available by the developer community, this upgrade path could be a desired option for those that want to take their projects to a greater scale.

FreeRTOS supports more than 30 embedded system architectures, including ARM variants and TI MSP430. This compatibility has allowed for its use in various embedded applications.

4.2.1.4.2 Zephyr™

Zephyr™ is an open source real-time operating system initially began as a kernel developed by Wind River Systems and made into an open source project of The Linux Foundation in 2016. Although the project itself is new, its code has been developed over the course of more than 20 years. Wind River Systems, its originator, is well-known in the embedded marketplace. Wind River's real-time operating system VxWorks has remained an industry standard, powering vehicle systems and industrial robots, as well as the Mars Pathfinder and Mars Curiosity rovers. The kernel for Zephyr™ was derived from the VxWorks Microkernel Profile, intended by Wind River to be its own small-footprint real-time operating system. Zephyr™ inherits much from this valuable codebase and is offered under the Apache 2.0 license.

The operating system takes a successful platform and adapts it for the age of the Internet of Things where smaller, more power-efficient, and connected devices are prevalent. IoT is a specific focus for Zephyr™, and connectivity and security are at its core. It supports

Bluetooth, Bluetooth Low Energy, and Wi-Fi, as well as the standards defining IPv4, IPv6, and NFC. This it does while maintaining a small footprint. When built, the kernel binary may be as small as 8 KB.

Zephyr™ further distinguishes itself with multiple scheduling algorithms, including both cooperative and preemptive scheduling and Earliest Deadline First. These are paired with multiple queuing strategies based on linked lists, red/black trees, and multiple queues. Additional features of the operating system include thread-level memory protection, a virtual file system interface, and a Shell subsystem. Power consumption is also considered with support for a tickless idle.

Zephyr™ supports various architectures, including ARM Cortex-M, Intel x86, ARC, and RISC-V 32. These architectures cover a broad range of devices and allow the operating system to be deployed on many platforms including larger and more capable systems.

4.2.1.4.3 eCos

eCos, or the "Embedded Configurable Operating System," is an open source real-time operating system initially developed by Cygnus Solutions (bought by Red Hat) and ultimately transferred to the Free Software Foundation in 2004. Originally released in 1998, the operating system has been in development for nearly 20 years. Throughout the course of its life, it has been used in various applications including commonplace devices such as Sirius satellite receivers and NETGEAR routers. eCos has also been taken to space with its deployment in the data acquisition control system of the Alpha Magnetic Spectrometer, residing on the International Space Station. Currently, it is offered under a modified version of the GPL license (GNU General Public License).

The operating system may be identified by its focus on low interrupt latency, low overhead context switches, and a small memory footprint. It uses a multithreaded architecture to accomplish this, as well as a high level of configurability. This allows eCos to be scaled down to require as little as 1 KB of ROM and less than 600 bytes of RAM. This feat is achieved through the use of library linking and linker garbage collection, a feature included in the GNU compiler that was originally written for eCos.

The configurability of eCos sets it apart from many of its contenders. A Component Definition Language (CDL) allows management of configuration points throughout the system. These configuration points may be represented by "Packages" of a larger scale, such as a TCP/IP networking stack, or "Options" of a smaller scale, such as the size of the stack for the kernel idle thread. This fine control of functionality allows eCos to be built according to a very precise specification. By so doing, resources are conserved and not spent on unused features.

eCos supports multiple architectures including ARM, x86, MIPS, and PowerPC. These represent some of the most popular architectures in the modern day and put eCos in a unique position to answer the need for a real-time operating system on many platforms.

4.2.1.4.4 TI-RTOS

TI-RTOS is a real-time operating system developed by Texas Instruments and initially released in 2012. Although the operating system has only been in existence for a short time, many of its components existed long before its conception. SYS/BIOS, the real-time kernel of TI-RTOS, began in 1998 under the name DSP/BIOS. The Network Developer's Kit (NDK) and TI Wares, two other components of the system supporting TCP/IP and USB stacks, respectively, were started in 2001 and 2007. These have been integrated into one product made available freely to developers. Most components of TI-RTOS, including the kernel, file system, and networking stack, are provided under a BSD-like license.

Texas Instruments suggests that their operating system allows for a consistent platform, one that they say facilitates the porting of legacy applications to newer devices. The characteristics shared by TI microcontrollers and the individualized support provided by TI allow for guided transitions between product lines. Device-specific libraries are used to exploit functionality particular to the hardware and a graphical configuration tool is provided in Code Composer Studio, Texas Instruments' own Integrated Development Environment (IDE), to allow for the easy configuration of drivers required by an application. The SYS/BIOS kernel itself is scalable, meaning the operating system may be reduced to as small a size as the application permits.

As might be expected, TI-RTOS is designed specifically for Texas Instruments microcontrollers. This is a benefit for projects making use of such but rules out its deployment in other platforms. This may be a significant concern for developers who do not want to box themselves in to using only one microcontroller manufacturer, or who would otherwise want to maximize the compatibility of their system.

4.2.2 Programming Languages

Programming languages are an important part of the design process and as such each language has its pros and cons. There are two types of programming that are useful for microcontroller design which are procedural and object-oriented. Either will work for embedded design, but there are use cases for each and they affect the design of the system later.

When looking at what languages to choose, a programming language can be a compiled language, a scripting language, or assembly. These choices directly affect the speed and efficiency of a computing system and each type of language is used differently when designing a software system.

4.2.2.1 Procedural Programming Paradigm

Procedural programming is a type of structured programming that relies on routines and the conditions to flow the control of the program. The first programming languages of this type were Fortran, Cobol, and Basic. The modern languages that follow this procedural model are Pascal and C, which were created in the 1970's. Procedural programming languages are typically closer to the hardware of the system that the language is being used on. The control flow of a procedural language, shown in Figure 26, is dictated by conditions in which the system is found. The program can be seen as having distinct states depending on what conditions have been chosen. Usually a procedural program can be modeled by some sort of flowchart which dictates the high-level design and the main routines. Then smaller flowcharts are used to describe the behavior of the sub-routines which are referenced in the main routine.

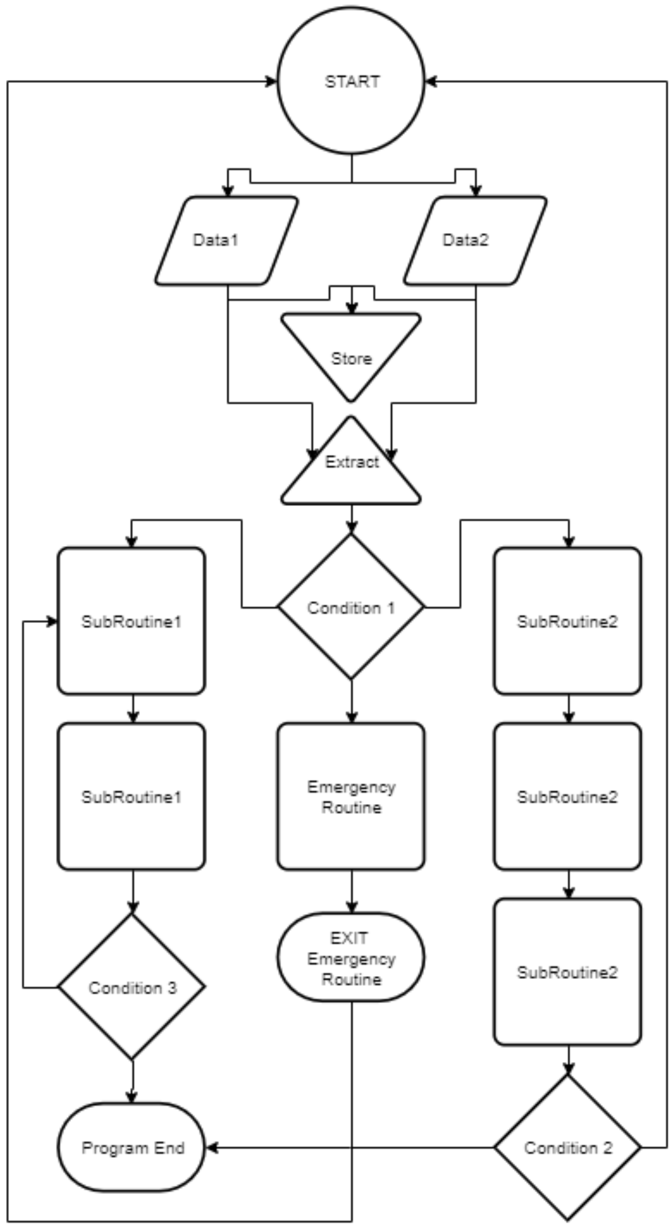


Figure 26: Procedural Programming Flowchart Example

Generally understanding memory architecture and how the CPU executes commands is helpful for these languages as they typically get translated or compiled into assembly language, then they are assembled into binary which the processor can understand. Because of this procedural programming is incredibly common for embedded systems and microcontrollers as they work natively on all microcontroller systems without extra configuration or the addition of operating systems or any other extra software to force compatibility.

For procedural programming modularity and scope are important factors. When programming with procedural languages memory is typically being directly accessed so minimizing the scope of individual functions is important to ensure the reliability of the program. This also helps with modularity as it lets the program run with only the necessary components and allows more efficient control schema which can ultimately lower runtimes and computational time.

4.2.2.2 Object-Oriented Programming Paradigm

Object-Oriented programming languages use a completely different design paradigm compared to procedural languages. Where procedural programming follows a flowchart like structure, object-oriented languages identify what objects are within a system and then break down all of the components into their respective objects. Just like a real-life object, usually objects have some actions they perform and some states they are in. Unlike procedural programming which follows a state diagram using true/false statements, an object-oriented definition of state is usually dictated by what the objects within the system are doing. This always changes the structure of a program to reflect the individual objects and then what they can do.

Usually object-oriented languages contain some sort of procedurally, but with most design patterns it is kept in what is called the main method or the main control of the system. With object-oriented languages modularity is usually innate to the components of the design, but object can also be related to each other and have what is called inheritance. Inheritance is simply about what an object is related to and the state that the object is in. This gives object-oriented languages amounts of flexibility that are not present in procedural languages because you can create new objects that inherit from other objects. Which allows that new object to take the functionality of that parent object extend it and modify it as needed. That behavior allows for programmers to write less code with more functionality. This is a desired aspect to programming as it allows for simple, modular, and expandable program design without changing the scope of the parent objects directly. Usually, object-oriented programming can be modeled by class diagrams, exemplified in Figure 27, showing the relationships between objects and how they interact with each other.

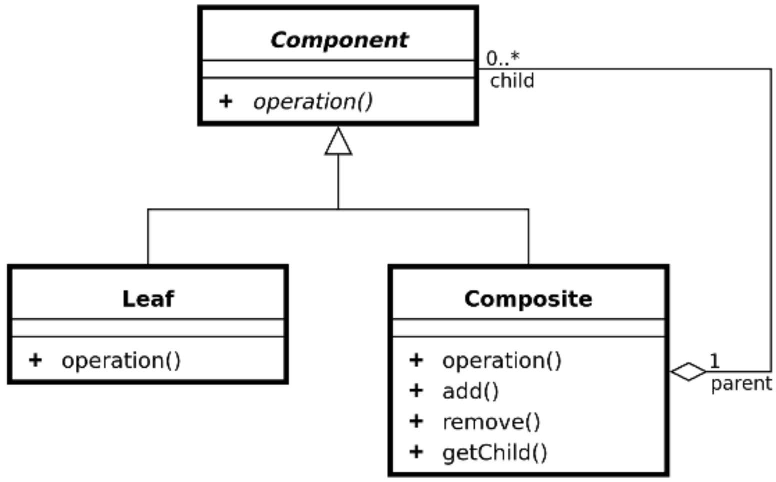


Figure 27: Object-Oriented Class Diagram in UML

4.2.2.3 Assembly Language

Assembly language is naturally procedural as it is the lowest level of programming without writing binary. It is naturally procedural because a computer will only execute instructions in a specific order and that order is exactly what was written in the assembly program. Assembly is converted directly into binary code that a processor will run by an assembler. Generally, assembly gives you direct control over relative addresses, the registers on the processor, and all input output devices. Because of that direct control assembly is different for every processor architecture and is not modular or compatible with any other system. This makes assembly a terrible language for writing a program for multiple processors and because of that it is highly discouraged in general computer systems. For embedded systems this is different though, because embedded systems usually are only going to be used for one specific function so using assembly is typically accepted. Although a compiled language would still be easier to use than trying to write an entire multifile program in assembly. Not only that, but a well written program in a compiled language will typically be more efficient in the processor than a program written in assembly because compilers in modern times are very good at converting code into the most efficient form possible. Twenty or thirty years ago this was not the case as compilers had not been developed as extensively until recently but in modern computing it will always be easier to program in a higher-level compiled language.

4.2.2.4 Compiled Languages

Compiled languages are one step above assembly usually. A compiled language has what the programmer writes in, then when the programmer goes to run that code, it gets compiled into assembly code via a compiler. After compilation then the compiled code, which is now in optimized assembly, is then sent to the assembler to be assembled into a binary executable which can then be ran or flashed to a device. Because of this, compiled languages are incredibly fast and will always run faster than a scripting language which works differently.

4.2.2.4.1 C

C is a procedural language that allows the programmer to program systems easily and reliably. Although somewhat complex, C is a common language for hardware level control over a system. That is important for an embedded system as it gives the programmer access to what memory is being used at what time. On top of this, because C is a compiled language it can be easily flashed to a processor as an executable binary that makes the system incredibly fast and accurate. Generally, C is good for creating efficient data structures and procedural programs that work based off of the inputs to the system. The C standard does not allow for multithreading and multiprocessing. This can be a problem for many modern systems that need multiple threads running concurrently to stay within time constraints.

4.2.2.4.2 C++

C++ is an object-oriented and procedural language that extends the functionality of the C language. C++ follows a lot of the same ideas as C, allowing for pointers and hardware level access to memory. C++ is also more common in many cases since it allows the programmer to take a more object-oriented approach to solving problems which is useful for most programs. Allowing the programmer to create interfaces to more complicated components allows the program design to be greatly simplified. With this though adds some complexity to the language. One unique feature to C++ is namespaces, this allows the programmer to create multiple groupings of variables and functions that extend the usability of the language. In C, once a variable is declared using the name 'value' for example. That variable is unable to have its type changed to anything else. Namespaces solve this problem and allow the programmer to create a namespace for collections of variables being used. So, you can have a namespace that uses integers for example. Then you can have a namespace that uses floating point values. This allows C++ to have more variables per program and helps significantly with readability since you can have more variables with the same name, just a different namespace. C++ following the updated standard supports threads and multiprocessing unlike the C standard which is important for systems that need concurrent processing.

4.2.2.4.3 Golang

Golang is an object-oriented and procedural language that has native support for threads and processes unlike C and C++. Native methods for multiprocessing, multithreading, and networking were included in the inception of the language which has made the language incredibly robust. Golang has taken inspiration from many different languages and tried to improve and use all of those aspects except for a few that the creators deemed detrimental to the language. Mainly inheritance and exceptions, the creators wanted to use implementation instead of inheritance. So instead of an object inheriting from a parent class, it just implements an interface which allows for objects to have the same methods but different inner implementations. This does force code reuse typically which can be seen as a downside but overall it allows for a more robust language because it forces programmers to create new methods for different objects and this increases the likelihood of finding new problems with the old behavior that would have normally been ignored originally. Golang also features garbage collection and does not support pointer arithmetic. This is different than C and C++ which allow for direct memory management and can affect the design of a program. Garbage collection in Golang is significantly better than in Java, but the lack of pointer arithmetic and direct memory management can sometimes hinder a system that is actively trying to manage memory. This is especially important for systems with limited memory. For a system with a large amount of RAM, that behavior may be okay, but for a system without it, direct memory management is important because you can control exactly when pieces of memory are used and freed allowing for incredibly efficient usage of low amounts of memory. This is important for embedded systems that generally have access to very low amounts of memory (in the order of 10's of kilobytes or less).

4.2.2.4.4 Java

Java is an object-oriented language that utilized a virtual machine to run. Java is incredibly useful for large systems and has support for multithreading, but not multiprocessing. Java is a bit different than C, C++, or Golang, as it runs on its own virtual machine. This makes java incredibly useful for running on multiple operating systems without needing to change any code. This type of behavior is good for general computing, although for embedded systems this type of architecture is not useful since the code is specific to the embedded hardware on the device itself. Another issue with Java is that although incredibly extendable, it is much slower than C or C++ because Java does not get compiled to assembly. It gets compiled to bytecode, which is then run through the java virtual machine which translates the byte code into instructions that the computer is capable of running. This is okay for non-time sensitive programs, but for a system that needs to run in real time this behavior is typically too slow. With that, Java does have a garbage collector like Golang, but the garbage collector is not optimized properly for real time systems and can perform collection at inopportune times and cause failures in the system. For a system that needs to run 24/7 at near 100% accuracy, this is unacceptable, and because of that Java is not used as an embedded systems language. Overall, Java as a general-purpose language for soft real time systems and personal computer systems works wonderfully and offers a range of extendibility that many languages do not offer.

4.2.2.5 Interpreted Languages

Interpreted languages execute instructions at runtime and are by nature significantly slower than their compiled counter parts. Usually interpreted languages are also ran in some sort of virtual environment so that the language can handle the added computation and methods involved. Typically, these languages do not just run line by line as it is impossible to do so and too inefficient, so they use a type of intermediate step between compiling and running.

4.2.2.5.1 Python

Python is a commonly used high level, object-oriented, and procedural general programming language that emphasizes readability and simplicity. Python is an implicitly and dynamically typed language meaning that the types of variable depends what is passed to them and they can change. This means a variable can be created as an integer, then subsequently changed to a string for example. This behavior makes python easy to work with as a programmer, but it can introduce some typing problems with large programs if the programmer is not aware that a specific variable was changed somewhere else in the program structure. Python, although a scripting language, does use an intermediate representation by compiling the program each time a program is changed before execution. This cuts down on the run time of the program significantly to the point where any normal user would not notice a difference in the program being ran. But for time sensitive application it still is not preferable since the time between actions and compiling can be shortened significantly. A comparison of Python and the aforementioned languages is found in Table 4 below.

Table 4: Comparison of Programming Languages

	C/C++	Golang	Java	Python
Works w/ MCU	Yes	Yes	No	No
Compiled	Yes	Yes	~Yes	No
Speed*	1	2	3	4
Reliability*	2	3	4	1

* 1 (Best) – 4 (Worst)

5 Design

After thorough research of all possible options and solutions to our problem, we designed our Situation-Aware Stop Signal to be an effective, low-cost device. Hardware, software, power, and mechanical design were all aspects of the project that had to be considered. Each of these subsystems were dependent on each other and had to be designed concurrently. Having multiple constraints over each subsystem required several viable solutions to our problem. Overall, the best products and designs were implemented and created to optimally satisfy our project goals.

5.1 Power Design

The power system design is the most crucial, limiting factor of our project. The power supply system must efficiently produce enough power to both operate and charge the device. The power supply system is split into three major components: the battery, the charge controller, and the solar panel. The solar panel generates electricity using energy from the sun. This energy is then used to recharge the battery, and the device is fed from the charge of the battery. Because of inconsistent charging rates, the charge and output of the battery is regulated by the charge controller. Ultimately, this results in a self-sustaining DC voltage source. The benefit of this is the low cost of operation and installation. Comparable devices would require an electrical connection to the grid, which is a cost this product avoids.

5.1.1 Solar Panel Design

The efficiency and effectiveness of our power supply system is going to be dictated by the solar panel design and execution. To fully cover the surface of our device, the solar panel used is assembled from several solar cells instead of purchasing an already assembled solar panel. By assembling our own solar panel, we were given the opportunity to maximize the surface area used by our solar panel. This allowed us to design the solar panel around the size of the device due to the flexibility of design.

Silicon solar panels is used in this device due to the market availability and the cost per cell. While other material solar panels are on the market, many are too expensive for the budget constraints of this project and not available for consumer purchase. Silicon is a common element, which is easily able to be mined; thus, the cost of these cells is affordable enough to be implemented on small, lower budget applications.

Monocrystalline solar cells can produce more power compared to polycrystalline cells. This is due to the purity of production. Because of our need to produce a large amount of power via solar panels, using monocrystalline cells allow this project to easily meet our power objectives. The cell we have chosen to create our panel out of is the SunPower C60. SunPower currently produces the most efficient silicon solar cell on the market that is available to consumers. While the quality does come with a price premium, the price of solar cells has been dropping as technology has advanced, and they can now be obtained at an affordable cost.

5.1.1.1 Solar Panel Layout

Each solar cell is 125mm in length by 125mm in width. Our device has a surface with the length and width of 25 inches to accommodate having a grid of 25 solar cells. By arranging the solar cells in a square grid allows for optimization of space on surface of the object. SunPower solar cells, unlike most solar cells on the market have their anode and cathode contacts both on the back of the solar panel. The elimination of any contacts or busbars along the visible surface of the solar cell allows the solar cell to accept more photons; thus, being a more efficient panel. Solar cells with silver busbars tend to reflect a fraction of the photons that hit the surface, which may have been absorbed with a panel that is free of any obstruction.

The 25 solar cells are all arranged in series. Since each of the solar cells provide a small voltage, they will collectively be large enough to charge a battery. The contact patches are on the back of the SunPower c60s, and a dog bone style contact is soldered onto 6 points of each cell. Two sides of the solar panel have three contacts each. On one side will be the cathode, and the other side will be the anode. This is because each cell contains 3 cells in parallel. After all the cells are connected, the three contact patches are connected in parallel which will add up their current values collectively. Once all the panels are wired in series, wire leads are connected to opposite sides of the solar panel to allow for easier installment to the charge controller.

Unlike solar panels, individual solar cells must be handled with care. There is no structural backing to prevent the panel from damage, and there is no resin or glass that prevents the front from being scratched. Grease and oils from human contact can degrade the life of a solar cell, so gloves must be worn during assembly. SunPower cells, unlike others on the market, are constructed of a semi-flexible material. While these cells cannot be purposefully bent, having a flexible panel lowers the chance of breaking a cell upon assembly. Most solar panels are light and brittle.

5.1.1.2 Solar Panel Physical Design

The solar cells are secured onto glass sheets that are cut to the same size as the solar panels. After assembly of the array of solar cells, they are then sandwiched between two clear pieces of glass. The glass panel was then cut with only 12.5cm of overhang on each side. This allowed for a large enough area for good adhesion between the two pieces. Glass provides a scratch resistant surface that will easily allow photons to travel through. Once the glass is secured, silicon will be used to seal the two pieces of glass together. The panels are each only 150µm thick, so these cells lay flat and allow for the glass panels to be easily secured together. After secured together, the panel was fashioned at the top of the device. The panel is stationary and flat to provide optimum sunlight exposure during any time of the day with minimum obstruction.

The two pieces of glass must be waterproof, to prevent moisture to get in. Moisture does not only damage the solar cells over time, but it also fogs up the glass which would prevent the solar panel from operating at its full potential. For additional protection against harsh weather conditions, EVA film may be used. The EVA film creates a protective barrier over the solar panel that resides under the glass.

The solar panel outputs a low voltage DC signal, so the solar panel is fashioned close to the charge controller. The charge controller is an intermediate between the device, the battery, and the solar panel. By minimizing the distance between the devices, the voltage from the solar panel is less likely to experience a drop. The solar panel outputs 14.5V which is enough to charge the battery; however, a decrease in potential difference would result in a slower battery charge.

5.1.1.3 Solar Panel Specifications

The solar cells being used will each provide 0.58V and 5.93A. This results in each solar cell producing a theoretical 3.43W. These values are under ideal conditions that the panel were tested under, so for design calculations, we are using 80% of what SunPower's datasheet specifies. By connecting 25 panels in series, the output of the solar panel is estimated to have a voltage of 14.5V and a current of 4.74A. The voltage value of 14.5V was picked to have a large enough potential difference to charge an 11.1V - 12V battery. If loss is incurred before the charge controller, a boost converter can be used to step up the voltage to an appropriate level.

The current value being assumed is 80% of what is listed in the datasheet while the voltage is estimated to have little a negligible loss. Because solar panels are semiconductor devices, they will only allow for current to pass through once a cut-in voltage has been reached. Under the assumption of 80% efficiency, the solar panel provides over 68 watts to the battery with the potential of producing over 85 watts under ideal conditions. While the device will only be actively consuming less than 15 watts, the surplus power being produced is to charge the onboard battery.

5.1.1.4 Comparable Products

Multiple solar panel qualities were considered when selecting a solar panel for our device. Among these traits were price, efficiency, quality, reliability, and panel layout. Our focus when designing our solar panel was to produce as much power possible within our budget, which made price and efficiency the most important qualities.

While our team looked through dozens of solar panels and solar cells, the five listed in Table 4 were decided to be the best candidates for our project. With budget being a large constraint in the project, the large price increase of solar panels vs. solar cells has to be addressed. The appeal of solar panels is the reliability of a mass-produced manufactured product as well as the plug-and-play ease of install.

The best two options for solar panels were produced by Seed Studio and SparkFun. Both companies are reputable providers of electronic components, so their thoroughness and accuracy of datasheets can be relied on. The solar panels made by SparkFun provided a 19% efficiency which is high compared to most of the competition in the marketplace. This panel was coated in a waterproof urethane coating. Waterproof coating is an essential feature which solar cells do not provide. Because this was an already assembled solar panel, it also had both 3.5mm and 5.5mm barrel plug adapters. This makes for easy testing and installation. These were the leading reasons to consider this product. Ultimately, we did not decide chose this product as our solar device. This was due to the comparably low power production as well as the total price. The solar panel created by Seed studio offered similar benefits as the panel made by SparkFun. This product offered an easy installation at a high price point. Due to the lower power production, this product was not the best option for our solar panel.

Using multiple solar cells to create one solar panel is a budget effective route which can provide similar or better results than using a pre-assembled solar panel. The SunPower C60 provides the highest efficiency of any consumer grade solar cell. While more effective Gallium Arsenide cells exist, these can be thousands of dollars more expensive and are difficult to obtain in small quantities. SunPower cells offer a unique look, with no visible busbar lines running along the surface of the cell; this is because all of the lines are on the back of the cell. By positioning the busbars to the back of the cell, less photons are reflected by the white or silver lines and more photons will be accepted by the cell. All of the connections are hidden on the back, which allows the cells to be in close proximity of each other and provides an attractive product. SunPower also creates solar panels with their cells already connected and protected; however, these are panels that are too large for our application and do not comply to our budget constraints. A unique feature SunPower cells provide is their flexibility. SunPower cells are flexible up to 30°. Our panels is positioned on a flat surface; however, the additional flexibility makes these cells less brittle and less likely to be broken during the assembly process.

Misol solar cells are a monocrystalline solar cell sold on Amazon at a more affordable price point than SunPower cells. These offered a similar output to SunPower cells; however, due to their lack of brand recognition and technical datasheets, we decided not to go down this route. Our solar cells are connected in series to create the appropriate voltage to charge our battery, so having incorrect data and solar cells could result in a large difference in output. With the cells we were considering, it would require about 25 cells to cover our area. While these were offered at a lower price than the SunPower cells, we decided not to use these due to the reliability and less power production.

A variety of cheaper solar cells were available for purchase from Amazon, eBay, and Alibaba. There were several which looked similar and claimed to have comparable features. This brings forth a reliability dilemma, which was the biggest reason why we decided to further pursue these products. The Aoshiki Mini Solar Cell was one which appealed to us due to its price point and efficiency. Our device surface would be able to be fully covered in solar cells for less than $40. These panels were much smaller than any of the other ones considered. At only 2 inches in length and width, we would need to connect

144 of these together. Each solar cell has six points of connection; this would result in over 800 solder connections. Due to the reliability and size of these cells, this was not a reasonable option for the scale of our project. These and aforementioned cells are compared in Table 5.

Table 5: Comparison of Solar Panels and Cells

Product	SunPower C60	Misol Solar cells	Seeed Studio POW92145O	Aoshiki Mini Solar Cell	SparkFun PRT-13784
Price	$97.50	$77.97	$199.00	$39.50	$474.00
Power generated (625 sq. in)	80W	70W	50W	66W	54W
Efficiency	22%	17.8%	17%	17%	19%
Layout	Solar Cell	Solar Cell	Solar Panel	Solar Cell	Solar Panel
Waterproof	No	No	Yes	No	Yes

5.1.2 Battery Design

The battery of the device was a deciding factor as to which components are able to be used in this device. The size of battery dictates how long the device can run without being recharged. Because of this, multiple options were considered upon the task of choosing a battery. To account for the battery demands of the device, the charging rates required, and the budget of the project, we decided to use the GTK 3S, a lithium ion battery obtained from AliExpress.

5.1.2.1 Battery Type

There are multiple chemical combinations which are used to make a battery. Each battery uses two elements which are separated by an electrode. Once a load is connected between the two chemicals, a reaction occurs which allows electrons to flow and current to travel. To fit our application, lead acid and lithium ion were amongst the most discussed. Lithium ion batteries offered more watts per kilogram than lead acid which is necessary when confined to a small space. Lithium batteries also charge faster than lead acid batteries. Because our device is powered from a limited amount of sun hours, utilizing as much of the energy brought in from the solar panel is important to maintain sustainability. Lithium batteries outshine lead acid in most ways besides the cost. Lithium ion batteries can be four times the price of lead acid batteries. Ultimately, we decided to use a lithium ion battery over a lead acid due to the various benefits they offer. To offset the price difference, we sourced our battery from China instead of a manufacturer in the United States.

5.1.2.2 Product Specifications

The largest consumption of power in our product can be accredited to the large 200mm traffic signals. These can amount to about 70% of the power consumption because they must be constantly running. The traffic light is designed to have the battery intake and capacity to operate throughout the day and night. The lights on this device consume 12V each while the rest of the components only use 5V. To accommodate for the large power consumption of the lights, we decided to choose an 11.1V lithium ion battery. For the minor power-consuming products, the voltage is stepped down using voltage regulators. The output current of the device is calculated to be 1.5A. To compensate for this, we decided on a battery with a maximum current rating of 2A.

The capacity of the battery ensures that the device is able to run constantly. To provide reliability to the device, we decided to choose a battery that would provide enough power to run without any sunlight for two days straight. This would be for extreme conditions because, even when it is overcast, a solar panel is able to produce a sizeable amount of energy. Our battery has a capacity of 20Ah which meets the current demand of our device.

Our device is suspended in the air which had to be planned around. On a windy day, traffic lights can be seen swaying in the wind. Because they do not use any visual sensors, they would not experience a noticeable negative effect. Our device utilizes LiDAR and RADAR sensors, so we have to keep weight in mind with our product. Our battery weighs less than three pounds which is only a fraction of what a similarly sized lead acid battery would weigh.

5.1.2.3 Comparable Products

There were many batteries considered to power the device. There were many fitting options due to the flexibility of using lead acid or lithium ion. The ExpertPower EXP12200 is a lead acid battery that offered a large capacity with a low price point. This battery was highly considered; however, due to its high weight and the slow charge rate of lead acid batteries, it was deemed an inadequate fit for our device. A comparison of this and similar products is given in Table 6 below.

Table 6: Comparison of Batteries

Product	ExpertPower EXP12200	UPG 85980	GTK C12	GTK 3S
Type	Lead Acid	Lead Acid	Lithium Ion	Lithium Ion
Price	$40.00	$65.00	$64.90	$87.00
Voltage(v)	12	12	11.1	11.1
Amp-hours	20	35	10	20
Weight(lbs.)	12.5	22.6	1.7	2.7

The UPG 85980 as seen in Table 6 shows a capacity almost double the size of the other batteries being compared. This battery, however, weighed over 20 pounds, which would add substantial weight to the overall product. Additional capacity would not be a large benefit over the battery chosen because lead acid batteries cannot dip below 50% charge without receiving damage. The GTK C12 was a similar battery to the GTK 3S chosen. This battery was offered at a lower price-point but was half the capacity. 10Ah would not last a full two days without charge; thus, it was not a good fit for our product.

5.1.3 Charge Controller Design

Charge controllers play an important role in a solar-powered product in ensuring reliability, efficiency, and longevity. Charge controllers supply reliability by providing a constant voltage from the battery, even while the device is charging. Batteries only charge when a larger potential difference is applied across them. This effectively increase the voltage of the battery; however, if a charge controller is utilized, it regulates the output voltage during charging periods. Charge controllers add sustainability to the device by allowing the battery to optimally recharge while receiving energy from the sun. By utilizing a charge controller, the lifespan of the battery can be improved. Charge controllers improve the lifespan of the battery by trickle charging after the battery is nearing maximum capacity as well as providing protection against overcharging. To satisfy all of the requirements of our device and project, we decided to use the PowMr 30A charge controller. This controller satisfied the needs of our device and was within our budget.

5.1.3.1 Efficiency

After being presented with a solar-powered project under tight power constraints, optimizing energy obtained from the sun was imperative towards having a functioning prototype. To achieve this goal, we decided to use an MPPT style charge controller over a PWM charge controller. MPPT charge controllers offer efficiencies over 95%, and the one we decided on has a max efficiency of 98%. This is, however, under ideal conditions; regardless, MPPT style charge controllers have a higher efficiency rating due to their ability to utilize excess voltage and produce more current.

5.1.3.2 Device Specifications

The charge controller is an intermediate between the battery, the solar panels, and the device, so it must meet the specifications of each. Solar-powered devices have a large range in sizes, from small devices to those powering an entire house. Because of this, charge controllers are produced to operate under multiple combinations of input and output voltages. For our application, our solar panel produces 15 volts during operation. Our charge controller is rated to be able to step this down to 12 volts during operation which is a key feature of this device. The output of our battery is 12 volts due to the chemical properties of the lithium ion batteries. The PowMr supports this output voltage which makes it a viable option for our device. Charge controllers have an input range of 1A to 80A. Choosing a charge controller that has a large enough current rating to ensure that no current is ultimately grounded and that the charge controller can handle the load being input into the system. Our solar panel outputs 6A of current and the charge controller is rated to withstand 30A. We decided on a charge controller with a larger amp capacity to allow for future expansions of solar panels without affecting other aspects of the power delivery system.

Our device is powered from a lithium ion battery. Lithium ion batteries require an additional stage of circuit protection to ensure they are safe and not every charge controller offers this. This was a limiting factor in deciding which charge controller to use for our project. Most charge controllers are marketed towards larger-scale projects. These usually use lead acid batteries, so lithium ion support is not very common. The PowMr has support for lithium ion amongst several other battery types which make it a fitting choice for our device.

5.1.3.3 Debugging Display

Because we decided to purchase a charge controller rather than build one, we purchased a controller with a display that showed the status of the battery and charge controller. This would make debugging easier if any unforeseen issues were encountered. Our charge controller displays the input voltage and current, the output voltage, the battery level, and the temperature. Comparable products without this debugging screen could be obtained at a fraction of the cost; however, this device would show the estimated battery level which is an important statistic that could not be easily obtained without extra tools. The device is going to be outdoors where high temperatures are of concern; this device includes a cooling fan to ensure the charge controller does not overheat and it digitally monitors the temperatures.

The S.A.S.S. requires a constant output voltage to ensure that the voltage regulators respond accordingly and the 12V output lights display at an appropriate brightness. The charge controller includes an output voltage display value which helped in confirming that the device is getting the proper voltage supply.

5.1.3.4 Comparable Products

Ultimately, the two outstanding factors in choosing the charge controller for our device were budget and efficiency. Table 7 below shows a comparison of charge controllers that were discussed. Charge controllers are often the source of the most power loss in a solar-powered device, so having an MPPT charge controller with high efficiency was highly sought after if it was within budget. The PowMr was the cheapest MPPT charge controller that was able to be purchased that fulfilled all of our requirements. A comparison of the best options that we saw are below and they include four charge controllers:

- Renology Rover. – Expensive but overall a solid option
- PowMr MPPT-30A – Cheaper than the Rover for more options
- GreeSonic MPPT1575 – Cheaper than the PowMr but less options and power
- Binen-20 – Cheaper than all the others but significantly less efficient

Table 7: Comparison of Charge Controllers

Product	Renology Rover	PowMr MPPT-30A	GreeSonic MPPT1575	Binen 20A
Price	$150.59	$69.45	$49.59	$16.99
Style	MPPT	MPPT	MPPT	PWM
Current Rating	40A	30A	15A	20A
Output Voltage	12/24v	12/24/36/48v	12/24	12/24v
Debugging Display	Yes	Yes	No	Yes

The Renology Rover charge controller would be a charge controller that may be further pursued if our device was put into production. This charge controller had a tracking efficiency of over 99% which is one of the highest on the market available to consumers. This device also had Bluetooth pairing capabilities, so it is able to be monitored wirelessly from a mobile device. Because our device is suspended in the center of an intersection, having this capability would be useful under conditions of device failure. This device, however, was over twice the price of the charge controller that was chosen. The additional features could be utilized under a larger production budget; however, budget constraints could not account for the premium price these features demanded.

The GreeSonic MPPT1575 satisfied most of the requirements of the project. This device had a high level of waterproofing and was more affordable than the device that was chosen. This device, however, fell short due to its lack of displays. The charge controller display would allow the device to operate with a level of confidence, as it would display the operating conditions of the device and the signal. This charge controller also offered an efficiency of over 99% which is higher than the charge controller we decided on.

The Binen 20A charge controller was considered due to its good reviews and low price point. This device was a PWM charge controller; thus, it does not have as high of an efficiency as the others being compared. The efficiency of this device was not specified, but similar PWM charge controllers offer efficiencies of around 75%. Due to the power constraints of the project, we decided that this was not an adequate fit for our operation.

5.2 Hardware Design

Hardware includes the electrical components which input and output signals as well as compute any processes being carried out from the device. The input of our device includes a sensor fusion of RADAR and LiDAR modules. For each of these devices, the range, durability, ability to function outdoors, and their communication ability affect our decision making for input sensors.

The output of the device is a visual LED signal that coordinates traffic. Careful decision making was required to ensure that all MUTCD laws and regulations. Because the output lights are the primary power consumers, having these as efficient as possible would eliminate most of the excessive power consumption.

5.2.1 PCB Design

One of the key implementations of hardware design is the PCB or Printed Circuit Board. The way the PCB is designed can either make or break a project. An optimized design can decrease resistance, improve efficiency and help reduce excess heat output. In order to successfully create the Situational Aware Stop Signal that meets all design requirements, there are many techniques to follow that could both save energy and protect all the components.

The best PCB design tips or “The Golden Rules of PCB Design” are incorporated throughout the electronics industry and will be followed in this project. When designing the PCB schematic, one thing many PCB design programs do not consider is the size of each component. While some computer-aided PCB design software will have a few different sizes which can help when distinguishing the larger components with more pins from the smaller components, the components are often not to the exact scale they will be in the finished board. This can cause a spacing and routing issue for designers who think they have enough room for a certain component. A tool to prevent this issue in most schematic design software is adjustable grid lines. Fitting the size of the grid lines to match the general size of the components being used can help the designer avoid layout issues and maximize the board space.

The next step when adding components to a design is to always keep components as close together as possible (as long as there are no interference or electromagnetic field issues). This tip can increase efficiency of the design as well as keep the routes/tracks shorter, which helps decrease unexpected resistance in the network. If related components are grouped properly and separately from other components they will not interact with, there will be a better testing environment for making sure the design is working properly. Testing individual networks can help pinpoint the exact point of the error and limit the possible causes for that error.

Creating a ground plane on a PCB is becoming a more common design feature on many PCBs. This ground plane is usually made up of poured copper. This copper can also be used to create connections on the board. Because of this feature, this copper pour is now being used as a power plane. Instead of using small pads to connect components that require a lot of power, copper pour ensures that the power will flow quickly and effectively without increasing impedance. This ground plane can also have a positive impact on the number of connections that need to be made. Adding the power plane limits the amount of return paths for each component. Running multiple lines to the same area is now possible and can improve the design’s speed while allowing for more components on the board.

While many data sheets provide a range or list of values for minor components such as resistors or capacitors, there is usually one or two values that respond better to any one specific design. Limiting the ranges of these component values or just selecting one value can create consistency, better organization, and higher performance. Consistency in range selecting can also give the board a higher chance of success. If one component value is on the larger size and an adjacent component value is on the smaller side of the range, this could potentially cause unforeseeable issues in the circuit.

PCB software usually include checks or simulations to test the schematic before creating the actual board. These simulations, if available, are always beneficial to run when designing a PCB. These tests allow designers to catch errors in both the designing and routing/building stages before sending the PCB to the manufacturer. While most manufacturing companies will run other tests, they will most likely not check if the design performs as expected. As a result, that is up to the designer to check before printing.

5.2.1.1 Designing for Efficiency

When simulating, the wires connecting to pins or components produce no resistance or impedance; however, this is rarely true for wires on a PCB board. Most wires carry a resistance and if that resistance isn't expected, it can have a negative effect on any board design. Formula 4, found below, may be used to help determine the resistance based on the type of connection being used. This formula is based on the resistivity of the material used for the trace, along with the size of the trace (its length, thickness, and width).

$$R = (\rho * l) * (th * w)$$

Formula 4: Resistance of PCB Connections

Not only will a highly resistive wire create an additional resistance, but it may also cause a loss of power. Because this project specifically has significant power restrictions, even this small loss of power can affect the success of the entire system. Limiting power losses wherever possible can contribute to the project's overall efficiency.

Decoupling capacitors can improve efficiency within the circuit. The capacitors can help eliminate noise throughout the board as well as help create independence between networks so non-related subsystems of the board do not rely on each other. As long as suggested and standard values are used for the capacitors, they will help to improve the circuit and maximize efficiency. The placement of the capacitors is also an important way to add efficiency to a design. Moving these capacitors closer to the power and ground pins can reduce inductance.

One of the largest errors in PCB design is the inconsistency of the component data or pin layout and the PCB design layout. Making sure both the datasheet requirements and the schematic match the board layout will limit if not eliminate errors in the board design. Double checking placement of components, trace connections, and hole counts before submitting the design will also improve the design's success and limit the amount of time spent on extra testing.

Noise is a common occurrence in PCB design and limiting it can be difficult at times. In the case of this project, limiting noise is crucial in making sure both the RADAR and LiDAR signals are unaffected by noise from the surrounding circuit. This LiDAR module, in addition to other components, is easily affected by surrounding noise. To limit this noise, separating a trace that will create noise from one that is easily prone to noise can limit the noise growth. A rule of thumb when placing traces that are easily affected by noise is that a noisy trace shall be placed a distance of three times the width of that trace away from the noise affected trace. Running adjacent traces in parallel can also cause an excessive amount of noise as well as create a capacitance between these traces. This created capacitance can couple the circuits or make them dependent on each other. This can cause major problems in the circuit, especially if these parallel traces are in different networks or unrelated to each other. Traces with higher frequencies can also add noise to a network, especially if a component in that network is more prone to malfunctioning under noisy conditions. These issues highlight the issue of placing components too close together. Figure 28 shows our final rendition of our PCB which was designed in EagleCAD.

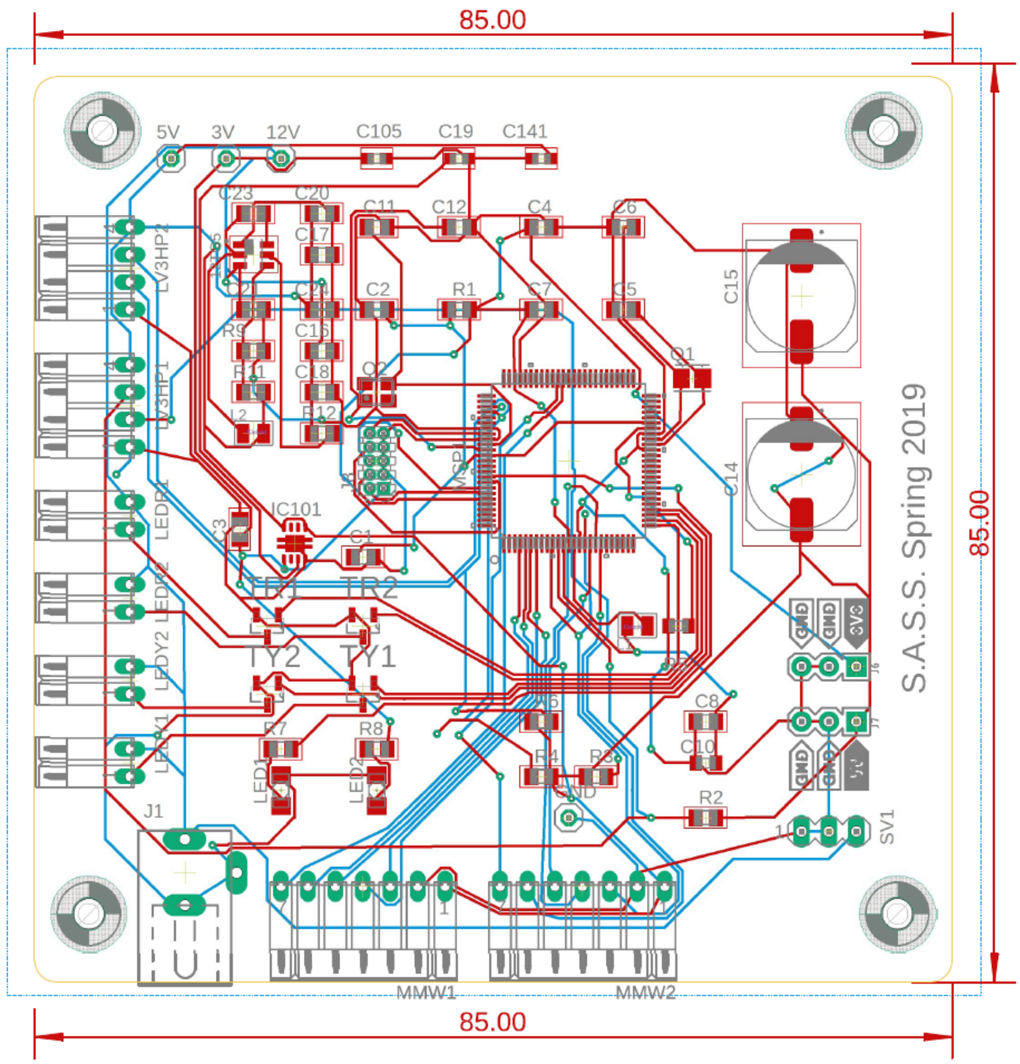


Figure 28: S.A.S.S. PCB Design

5.2.1.2 Designing for Testing

Because this project is time sensitive and must be completed quickly, it is important that the PCB design works the first time. Ordering more PCBs wastes both time and money. In order to ensure that the PCB works the first time it is printed, it will be designed for testing. Designing a board for easy testing allows the designer to check every network individually and helps speed up the testing process.

It is important to understand and recognize the devices that will be used on a PCB before creating the board. Understanding how each component will react with another can help avoid issues with the board later down the road. This will also improve the testability of the board. It is wise to create a “safe zone” on the board specifically for unusually large or tall components. These components are more likely to interfere with both testing and other smaller components on the board. For example, in the case of this project, a LiDAR sensor will be connected to the board. Because this module sends out a concentrated light signal, keeping this module near the sides of the board (away from most parts) will help the LiDAR’s signal travel further without interference and cause less interference with parts in the middle of the board. Keeping the LiDAR module away from most of the major components will also make testing easier. In addition to the lack of interference the distance provides, if the module is placed away from the test pads, it will not hinder other components having access to those test pads.

While the PCB design is up to the discretion of the designer, there are some important guidelines to follow when it comes to spacing and wire thickness. When it comes to testing, it is important to keep testing points approximately 2 mm away from each other. If these test points are too close together, interference becomes more likely and the results of each test become less accurate. Additionally, test pads should be placed around 3 mm from the edge of the board. This prevents the frame of the PCB from affecting the tests taking place. It is important to note that, when using and testing surface mount devices, increasing the size of the test pads will provide a better connection. Since the design for this project will include surface-mounted components, it is important to account for the size of the test pads in the design. The optimal size of these test pads is 40 mI, while the minimum is 25 mI, to ensure optimal test conditions 40 mI test pads are beneficial to include. These test points cannot be too close to any one component because they will not perform effectively if soldered or if they get ink on them.

The most important thing to remember when designing for testing is to know the total load current of each network on the PCB. Providing too much or too little current to a component could fry the board or cause nothing to turn on. Testing this could increase the chances of the board working or improve understanding of why it isn’t working. Separating the board into a network and grouping components that work together creates easy accessibility for testing and creates shorter connections which can help improve the speed. Creating multiple and easily accessible power nodes helps provide a way to test the current along multiple points of the board. Because this specific project requires speed and accuracy when adjusting the lights on the stop signal, evenly-distributed testing nodes are necessary. These test points cannot be too close to any one component because they will not perform effectively if soldered or if they get ink on them.

5.2.1.3 Heat Dissipation

Another concern for this project is the overall temperature of the product. Since this device will be outside all day in the Florida heat, it is also vital to minimize heat output due to each component of both the PCB and other hardware inside the signal. While most components produce minimal heat, this small output could be amplified due to the surrounding humidity and high temperature. In order to limit this heat on the PCB, we reduced the size of the regulators.

Keeping components that are sensitive to heat or produce a large amount of heat away from each other and close to heat dissipating elements is the first step to minimizing heat in the design. High temperatures can also affect the efficiency and accuracy of the design. Because this design includes MOSFETs, voltage regulators, resistors, and other heat generating devices, it is important to place them both far away from each other and far away from components that are easily affected by excess heat, such as capacitors. Additionally, minimizing the amount of current flowing through the components that are prone to generating a lot of heat will help prevent those components from affecting others close by. Because the heat released by these components directly depends on the amount of current traveling through them, minimizing current where possible can decrease the heat output of the entire board.

In addition to other heatsinks and fans, there are other small design adjustments that can help increase heat dissipation throughout the system. As mentioned before, increasing the size of the pads under surface mount components can improve their connection and testing ability. Similarly, increasing the amount of conductive material (in this case, copper) around the surface mounted devices can also help to provide extra surface area for heat to dissipate. Thermal relief is creating a connection between a copper pour and a PCB pad. This connection can make soldering easier while restricting heat flow throughout the connection.

When creating a multi-layered PCB board, certain traces will be cooled differently than others depending on both placement and what layer they are on. Because external (top or bottom) layers of the PCB are often cooler and the traces on these layers can cool easier, it is important to place heat producing components on these layers. This collection of heat occurs between layers of the board due to the fact that the current must travel between layers of copper before reaching a component. Higher frequency current loops can produce a larger amount of heat than lower frequency current loops due to the larger amount of current traveling through those loops over a certain period of time. As a result, these high frequency networks should be kept close to each other and have smaller traces. These small loops can also help reduce inductance and resistance of the overall network.

5.2.1.4 Vias

Another way to reduce heat dissipation throughout the PCB is the use of vias. Vias are connections between pads on two different layers of the PCB. The connections are made using conductive holes made of electroplating. This connection between boards can not only improve the design but can also help move excess heat from one side of the board to the other. Using multiple larger vias between middle layers of the board can help remove some of the extra heat generated from middle board layers back to the external layers where the heat can be dissipated more quickly. This transfer of heat can help contribute to a more reliable board.

It is important to note that, like traces, vias also carry a small inductance and resistance due to their conductive material. This resistance can be minimized, however, by using multiple large vias. Also, connecting vias to the ground plane can reduce the inductance of that layer, especially when connecting capacitors. The use of vias between planes can also reduce the noise level in the circuit.

Full tenting is a process that occurs when solder covers both the ring and hole on both sides of the via. This is not a good idea when creating vias for testing the design because the vias that are tented are not able to be easily accessed when testing. Vias, however, will need some form of protection. If full tenting isn't used, creating a solder mask can also protect the via from damage. Because this design required ample testing to ensure the accuracy of the PCB tenting will not be used on vias whose network must be tested multiple times.

When using a solder mask or other via protection, it is important to create the same protection on both sides of the via. If one side of the via is has a solder finish and the other does not, this can cause connection issues and cancel out the protection placed on the via. It is also important to only use vias between layers of the same type and material. This miss match in layer types can affect resistances, voltages and power drops between layers and can also ruin the board's circuit along with its via if not properly protected.

5.2.1.5 PCB Type

With the growing need for PCBs and the advancement in today's technology, different types of PCBs are now becoming more readily available for consumers. PCBs are being made more flexible and less rigid. These PCBs are made of a completely different material and layer design that the rigid PCBs. These boards have been known to improve efficiency if used correctly. Identifying the type of board needed is crucial to the overall design and success of the board. The board type can also affect the design and the way the components are placed.

There are many important board factors that depend on the type of board used. For example, switching to a different type of board can alter the layer count, amount of trace space, hole size, and amount of copper used. If one of these factors change, it is likely that the entire board will have to be redesigned. Even on the small chance the board does not have to be updated, the board properties can affect other aspects of the circuit such as the speed of the design.

While flexible PCBs may be the next great thing in PCB development, unfortunately the budget and use of the PCB limits the type of PCB that can be purchased for this project. As a result, the type of PCB that was used is a rigid-PCB.

5.2.2 Embedded System Design

To design an embedded system the function must first be tied to the requirements. Once the functionality is tied to the engineering requirements and the constraints are analyzed then the embedded hardware can be chosen and then integrated together into a PCB and the resulting system. For us, the three main constraints on system design are power consumption, processing time efficiency, and budget. Each of these affects all aspects of our design, but each constraint also affects the other. Processing time efficiency directly affects the power consumption because the faster a processor can calculate the necessary information needed to make a decision and then make that decision means there is more time for that microcontroller to sleep and save power. Which directly affects our power usage throughout the day and benefits the system as a whole since it runs on a very limited power supply.

5.2.2.1 Microcontroller

Our criteria for choosing a microcontroller/microprocessor include power usage, speed, memory size, and price. Our main constraint on our choice of microcontroller is power. We are trying to stay within 12 watts of usage during normal operation to guarantee that the system can run throughout the day and night. To guarantee this we want to choose a microcontroller design that is power efficient. The microcontroller being low-power gives us room to work with the other components and their power constraints.

The design cannot be solely focused on power efficiency since the system is to be placed in intersections and should be capable of responding to observed situations in real-time to guarantee driver safety. Not only do we need a design that is power efficient, we also need a design that can also perform adequately to read and interpret sensor data provided by the RADAR and LiDAR modules. Out of our options we manually tested the TI MSP430G2553, an Arduino Mega, the Arduino Uno, and a TI MSP432P401R. Each of these works well as standalone microcontrollers, but we are testing them on their communication interfaces and their processing speed while reading and interpreting data concurrently. The Arduino Mega and the TI MSP432P401R worked the best with our testing benchmark, but this was expected because they have the more powerful processors compared to the other choices in our list.

We chose the MSP432P401R due to its extra memory, processing power, and its proprietary low-power mode. TI's built in low-power mode functionality is important for our design because we want to sleep and wake the device at regular intervals so that it can save as much power as possible. Table 8 below, as seen below shows a comparison of some of the processors we have discussed and tested.

Table 8: Decision Matrix for Microcontrollers

Processor	**MSP430G2553**	**MSP432P401R**	**ATmega328P**	**Atmega2560**
Processor Speed	16MHz	48MHz	48MHz	48MHz
Flash Mem.	16KB	256KB	32KB	256KB
RAM	512B	64KB	2KB	8KB
Voltage Rating	5V	1.62V – 3.7V	5V – 12V	5V – 12V
No. GPIO	15	48	15	58
RTOS	2	3	0	0
LPM	Yes	Yes	No	No
Price	$2.20	$6.08	$1.28	$4.32

5.2.2.2 Interfacing and Programming the MSP432P401R

There are 3 possible ways to interface with the MSP432P401R, the first is to use a MSP432P401R evaluation board and an extra connector on our board. That method lets us use another device as the means of accessing the hardware which is good for security but is not necessarily the most preferable way to program and access the MSP432P401R. The second way is to include the XDS module that is also on the evaluation board for the MSP432P401R, this is the nicest way to allow our microcontroller to be programmed, but overall it is not the best because this requires the design of an entire new section that is separate from our microcontroller design which adds complexity and size to the PCB design that is not needed once in production. The third and final option is the J-Link Segger, with the 'Flash Programmer' or 'UniFlash' software that allows developers to flash the memory onboard a TI device with an image or binary file. This option is just as versatile as using the evaluation board as a programming and communication means but it is more expensive as a base version of the J-Link Segger costs at $378. The reason that the J-Link Segger is so expensive is because it is a dedicated debugging tool for all TI devices, so not only would the Segger be capable of flashing and debugging the MSP432P401R, but it would also be capable of flashing and debugging the mmWave sensor evaluation modules. This gives the Segger versatility that is incredibly useful for our system. Sadly, for our project using a J-Link Segger is outside of our budget so we relied on the evaluation board to flash. Table 9, below, compares the discussed methods of access.

Table 9: Methods of Accessing MSP432P401R Comparison

Product	Via Eval Board	Via Segger	Via additional XDS module
Versatility	2	1	X
Ease of Use	2	3	1
Debugging Capable	Yes	Yes	Yes
Flashing Capable	Yes	Yes	Yes
Easy to Implement	Yes	Yes	No, need to add to PCB

5.2.2.3 Embedded Configurations

When looking at the entire system for a product like this, there are many ways that the system can be configured. Each way has its own advantages and disadvantages, but they do offer the same functionality at the end. Each different configuration affects the control scheme of the system, the programming, and the other hardware involved, so this must be decided first in the design process after designing the overall look and functionality of the product. Overview of the different configurations are given below, beginning with Figure 29.

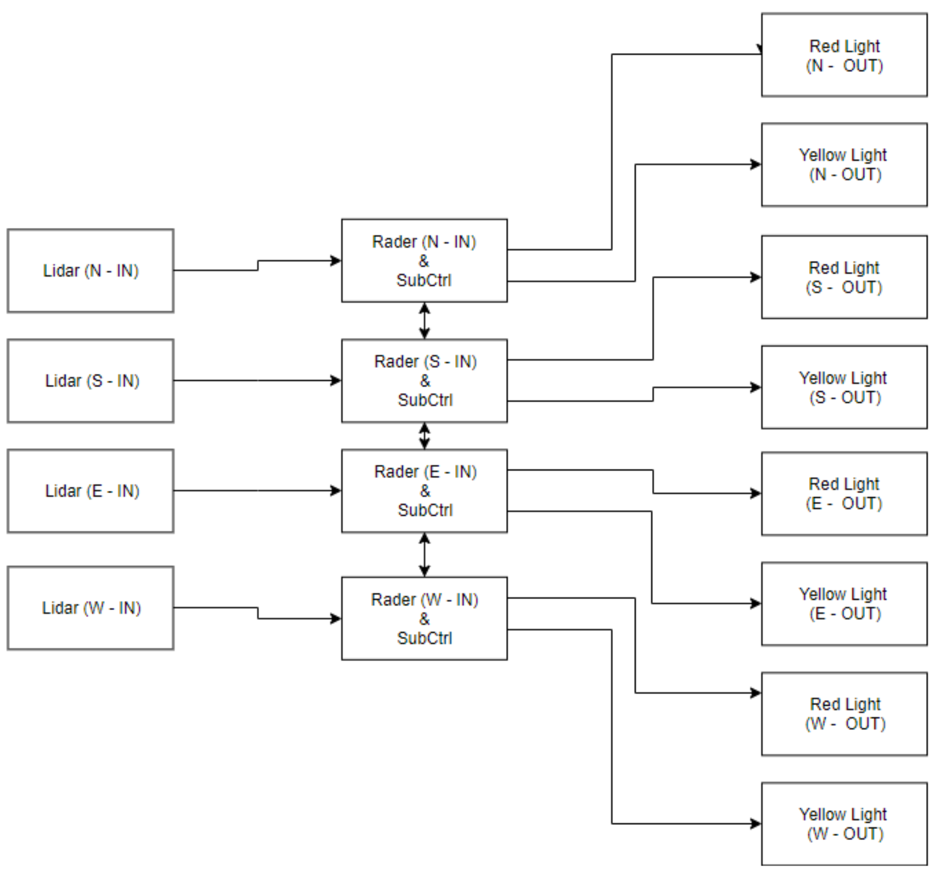


Figure 29: Embedded Design Block Diagram 1

One of the designs we considered was the above diagram. This design was a simple design with each means of control dictated by each individual side of the product. This design does have a few highlights that would make it viable for a system in production. First it alleviates the need for a main microcontroller module. That allows the package to be smaller and could allow for larger batteries or larger lights. With that if the product designer wanted to use cheaper RADAR sensors or the mmWave sensors, or even just the mmWave evaluation boards, each of them could be a good option for a production machine because it would reduce the design effort dramatically. The best option out of those would be to make a custom board with a RADAR sensor, it could be a normal RADAR sensor or the mmWave chip, the LiDAR sensor, and the microcontroller all in one package.

In this design, to use the mmWave evaluation boards we needed to use them as RADAR-only because they are capable of processing data just like our microcontroller unit would, so we would need to make custom boards with the mmWave RADAR module and possibly our own microcontroller to simplify the design so that it is usable. This type of design, although it does allow those custom boards to work together would be very energy inefficient compared to the next two designs. This is because there are more microcontroller units, which ultimately use more energy because all four of those modules would need to be active at all times during the day. Even if the microcontrollers draw very little power in sleep mode, having all 4 in a system at once is drawing a notable amount of power that is taking away from the rest of the system. Even though this design is not energy efficient, it is however faster than using just one controller. Since each sub-controller can individually process each side. This allows for “multithreading” in a sense since there are four processors doing the necessary computations to calculate safe speeds and manage the control of the system with that.

One reason we decided against this as the design of the PCB would be unnecessarily complicated just to get the mmWave RADAR sensor working correctly on a custom board when the development boards are available. Another drawback to this design is that it is not modular and the software for a design like this is complicated because each sub-control module must communicate with all other sub-controls. This behavior is programmable, but it is likely that a system designed this way would be prone to causing catastrophic errors just because of a slight timing mismatch. Due to all these issues, as a team we moved away from this design as it is more expensive, less power efficient, and has more complicated hardware and software that would not benefit the design of the system. For this reason, we considered another design, depicted in Figure 30.

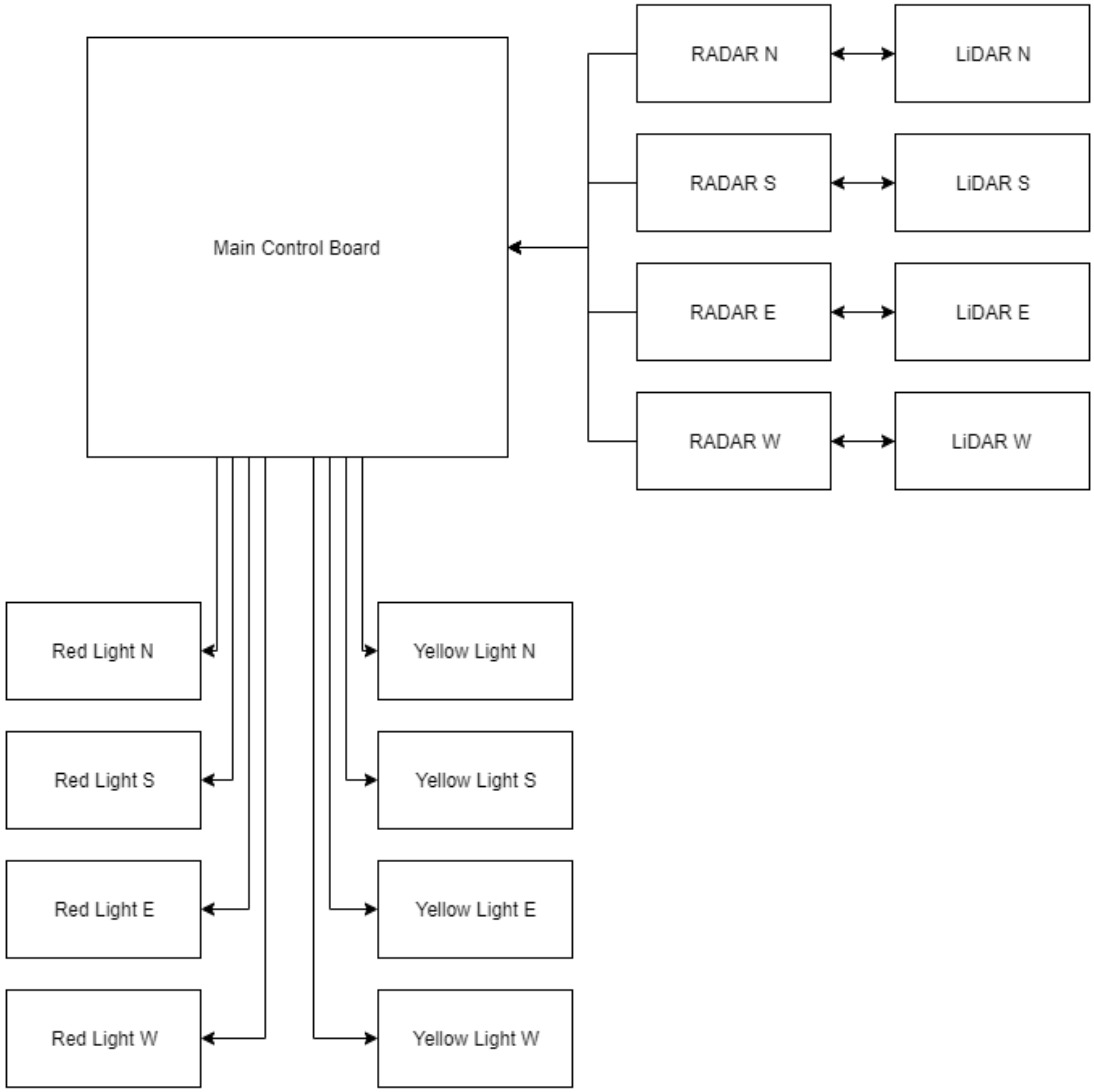


Figure 30: Embedded Design Block Diagram 2

This design was a good compromise of data processing distribution and locality. The highlight of this design was the LiDAR being integrated or at least connected to the mmWave sensor board which condensed all of the sensors into individualized sensor modules within the system.

In this design, we would allow the mmWave sensors (which contain Arm Cortex M4F Processors) to also control and read the information sent by the LiDAR sensors on the system. This offloads processing to those sensors who can then do some preprocessing before sending that information to the main control board that we are designing. This is good for a fully custom design, but for our overall design this configuration is not easily achievable because the mmWave sensors are limited. We decided that this would showcase the project and idea better, and to keep as much processing as possible on the main control board. With this type of design, the main board makes a decision after taking data from the sensor modules and then changes the lights for each side of the product accordingly. A similar data flow occurs in a final design explored, shown in Figure 31.

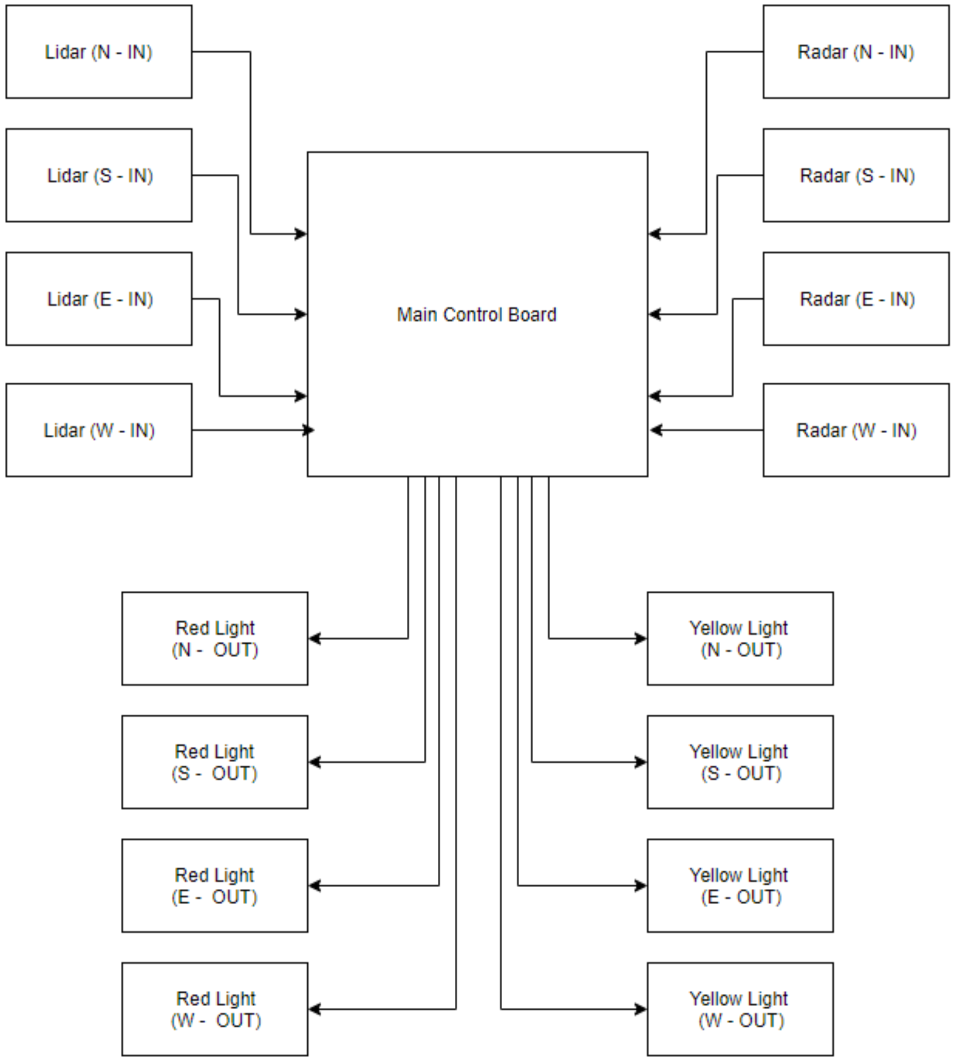


Figure 31: Embedded System Block Diagram 3

The above embedded systems block diagram showcases the most promising design we have created for the system, and it is the design we have chosen for this product. It allows for modularity and robustness that is incredibly important in a viable product. This design also allows us to control all aspects of the design and only use the components we deem necessary. It also permits 3rd party part replacements and a reusable design model in case the RADAR sensors, LiDAR sensors, or lights need to be replaced by other components. Because of this, our processor does have to do more since there is no preprocessing done of the sensor data. This gives the main control board more authority over what is done with the signals being received. In this case, we can use the LiDAR only when the RADAR detects an object, allowing us to save power and processing. When the LiDAR and RADAR are both active, the processor can then use that data to determine the proper action of the system without having to completely rely on the RADAR module to relay the correct information at all times. That behavior is very beneficial to the system because it gives the system the ability to go into a safety mode that bypasses the RADAR and LiDAR and falls back to a default mode that we decide.

5.2.3 Sensor Design

A major requirement of the Situational Aware Stop Signal is accuracy. In order to successfully manage traffic and prevent car accidents, this device is able to sense which direction a car is coming from and that car's location. While both RADAR and LiDAR sensors can detect this information on their own, each sensor has a different specialty. Using both LiDAR and RADAR sensors creates a sensor fusion that allows us to draw from the strengths of each sensor. This idea also creates necessary redundancies that can help prevent accidents by making sure the data received from one sensor aligns with data received from the other sensor. For example, because RADAR can usually have a higher accuracy over a wider range than LiDAR, this device uses the data gathered from the RADAR to find the car's general location on the intersection. Using the RADAR's wide range can help the device sense more cars over a distance and can pick up cars that are not able to be seen by the LiDAR module. Even though LiDAR has a smaller beam width, it can accurately pinpoint the car's exact location. Using two different types of sensors in the Situational Aware Stop Signal creates two different ways to get data that results in a more successful traffic management device.

In addition to the LiDAR module being used, a RADAR sensor will be used to detect oncoming traffic and trigger the LiDAR to turn on and pick up more precise readings than the RADAR would alone. The RADAR sensor will have a wider beam width to detect more objects over a wider range than the LiDAR module would. The requirements for this RADAR device included compactness, a large beam width to detect cars, speed, accuracy, and at a reasonable cost. As mentioned in the RADAR design section, different types of RADAR sensors have different wavelengths. The sensor's wavelength affects its performance and gives the sensor different strengths and weaknesses. Below in Table 10 is a comparison chart that presents the different sensors that were considered for this project.

Table 10: Comparison of RADAR Modules

	Hyperikon Microwave Motion Sensor	**IWR1642 Evaluation Module**	**TIDEP-0090**	**IWR1642AQAGABLR (MCU)**
Size (m)	0.07x0.06	0.01x0.01	0.01x0.01	0.01x0.001
Cost ($)	38.95	310.48	300	33.91
Range (m)	16	70	70	N/A
Beam width (degrees)	360	Approximately less than 90	Not specified	N/A
Power (W)	1.5	2	2	0.06

5.2.3.1 RADAR Design

The RADAR sensor module selected was the IWR1642 Evaluation Module board. This board uses mmwaves to detect objects with a frequency of over 100 GHz, the smaller wave size provides a more accurate and speedy result when detecting objects. As a result, we focused mainly on mmWave sensors when looking for a RADAR module. This sensor's large range and great beam width in a small package made it a great choice for this project. It is important to note that this board has also been used previously in other traffic monitoring projects, which helped solidify the choice. When looking at the other options, even though the Hyperikon Microwave Motion Sensor had the largest beam width, a problem arises when placing the sensor at an angle, which is needed for this project. Similar to the LiDAR with a 360-degree beam width, this module is not practical for this application. The other remaining choices all use the IWR1642 MCU and functioned very similarly. While purchasing the MCU alone would cost significantly less than buying the evaluation module, designing a RADAR board would be difficult and too time consuming to successfully complete within the time frame of this project. Ultimately, the choice was made when the IWR1642 Evaluation Board was donated to the group by the members of the Texas Instruments Lab.

5.2.3.1.1 Communication Design

Because this evaluation board already has an MCU, LDO, and other additional components on the board to process its data, the only connections required where the power and communication connections to this board. When powering this module, it is important that the datasheet is followed properly and the MCU receives approximately 3V. Because the solar panel will be supplying around 12V, this module will not be able to get its power directly from the panel. Instead, a voltage regulator on the PCB will decrease the voltage down to an acceptable value to power this MCU without frying the board.

This board allows multiple different ways and channels to send data through. Originally, the idea was to collect the data from the sensors on this module and send that information directly to the PCBs MCU (the MSP432) for decoding and processing. After, the MSP432 would send a signal back to the RADAR with instructions for what the RADAR should do next. This, however, raised concerns for the speed for the design and its need for a round robin software design. When taking these concerns into consideration, it was decided to have the IWR1642 MCU process the data gathered from the mmWave sensor on the evaluation board, then send an interrupt back to the MCU on the PCB. This helped create a more efficient and more accurate design for current and future versions of the Situation-Aware Stop Signal.

Both the MSP432 and the IWR1642 have multiple types of communication can then be used between modules, however, in order to keep the software design less convoluted, each sensor will use a different communication type and bus. This allows the MSP432 to multitask better and send instructions more efficiently, which helps to optimize the software design. Because the LiDAR module could only communicate via I2C, the mmWave will send an interrupt to the PCB via a GPIO pin.

5.2.3.1.2 RADAR Drawbacks

Unfortunately, there were still some issues and concerns that this RADAR module presents to the Situation-Aware Stop Signal. While the drawbacks of mmWave sensors, versus other sensors is in the RADAR design section, the drawbacks of the specific evaluation board will be discussed here.

Even though this RADAR sensor comes with an evaluation board that can process and translate the data received from the sensor, this module is not perfect. Like the LiDAR sensor, the data received can be negatively affected by weather and other obstructions. For example, while most of our hardware elements will be inside the plastic encasing, both the LiDAR and RADAR sensors must be outside of the plastic in order to get any readings from oncoming traffic. While placing the weather protected LiDAR module outside did raise some concerns, there will be more issues and a greater likelihood of issues when placing the evaluation board outside, allowing it to be vulnerable to harsh weather conditions. Any rain or even harsh sunlight will damage the module immediately if not placed in a protective casing. This casing is an additional design component that, if it fails to protect the evaluation board, will negatively alter the results of the RADAR sensor.

Additionally, due to the small size of the antenna and connections from the mmWave evaluation board to the PCB, placement of this sensor is extremely important and sensitive. For example, placing the board closer to the PCB will limit the extra resistance and time used to send data between boards; however, placing the antenna too far from the edge of the structure will limit its view of oncoming traffic. Also, if the antenna is not placed at the proper angle, the results will also be off and result in incorrect data being transmitted. Using the incorrect angle could result in the RADAR not being able to detect any moving objects as well as detect objects that aren't useful (such as animals in the grass). Placement of one sensor could also affect the results of another. If the RADAR sensors are placed too close to each other, they may detect the same vehicle twice which causes unnecessary triggering of the PCB MCU. Another issue arises when the RADAR sensors are placed too far away from each other. In that case, a car could drive by undetected, which may lead to an accident or even a fatality.

5.2.3.2 LiDAR Design

As previously stated, the LiDAR module will be used mainly to get more precise readings that will help the device to determine where a car is, where the car is coming from, and if the car is stopped at the stop bar. In order to do this, the LiDAR module must be very accurate, have a relatively large detection range, not require a large amount of power, and remain in budget. Because there are many different types of LiDAR modules available, such as rangefinders and laser scanners, a table was made to determine which specific type of LiDAR will be used. In Table 11, different types of LiDAR modules are compared based on the requirements listed above.

Table 11: Comparison of LiDAR Modules

Product	**Slamtec RPLIDAR A3 360° Laser Scanner**	**LIDAR-LITE V3**	**Halo XL450-7 Rangefinder**	**TF02 - LiDAR Rangefinder**	**TF MINI MICRO LIDAR MODULE**	**YD LIDAR G4 360° Laser Scanner**
Cost ($)	599	158	62	99	49	324
Rate (Hz)	16000	270	Not specified	100	100	9000
Range (m)	25	40	411	22	3-7	16
Power Required (W)	0.12	0.675	0.01	0.6	0.12	0.1
Size (in)	3.0 x 1.6 x 3.0	0.8 x 1.9 x 1.6	6.0 x 5.0 x 3.0	2.7 x 1.8 x 1.0	1.7 x 0.6 x 0.6	2.8 x 1.6 x 2.8

As the table presents, the LiDAR modules shown above each have different strengths and weaknesses. Choosing the perfect module for the Situational Aware Stop Signal required us to figure out what the top priorities were and the main function of the LiDAR.

Two rangefinder LiDAR sensors were considered for this project. While these products clearly have the best range, with a little more research, it was found that while the length of the range was great, the width of the range was not large enough to detect multiple lanes of cars. Each of the rangefinders must also be disassembled and reverse-engineered to determine how this module could meet the needs of the LiDAR sensor we need. As a result, these sensors were not chosen for the project.

The second types of LiDAR sensors compared in this matrix are the 360° laser scanners. Even though these modules had a full rotation of sensing range, because our device will he hanging from the air, both the sensors must be placed at an angle to properly detect cars. When placed at an angle, the laser scanners' 360° range would become less useful for this application. As a result, we would still need multiple LiDAR sensors and the cost of one laser scanner is more than double the price of some of the other LiDAR sensors. These devices can only detect objects in a closer range (at most 25m) but have the widest beam width at 360 degrees. Because these devices had an extremely fast frequency, the price was over our budget.

The last two remaining sensor modules are the LIDAR-LITE V3 and the TF MINI MICRO LIDAR MODULE. Each module utilizes a different form of detecting obstacles. The TF MINI uses phase detection, which directs the light in order to measure its distance. This measurement type can be very slow. The LITE V3 takes measurements of the time of flight, which bases the distance measurement of the speed of light. This measuring method is much faster and has now become cheaper with new technology. The TF MINI has a much smaller range at only 3-7 meters while the LITE V3 has a range up to 40 meters. Also, the TF MINI is over $100 cheaper than the LITE V3. The TF MINI uses a LED as its light source to detect objects. While the LED is a cheaper option, it is not as effective in outdoor environments or when detecting non-flat surfaces, such as car windshields. The LITE V3, however, uses a laser diode to detect objects. Even if this component is more expensive, it will be much faster and more efficient than using an LED.

The LiDAR that best meets these requirements is the LIDAR-LITE V3 made by Garmin. Even though this module requires the most power of those LiDAR modules that were compared, this module has the best range and smallest size. The power required is still relatively small when compared to other components and will not be an issue. This module also has a low power mode that can lower the amount of power needed by 1 - 2 watts. Additionally, because the LiDAR module will not constantly be on, even more power is saved. Other than the rangefinder, this module has the farthest distance range. This sensor also has a maximum operating temperature at 60 degrees Celsius, which is well within the average temperature in Orlando, Florida, which is where the device will be tested. While most electronics have a similar maximum operating temperature, it is important that the RADAR module be able to function outside without being damaged by humidity. The housing, as shown in the image below protects the sensor from the outdoor elements which is a necessity for outdoor products such as the Situation-Aware Stop Signal.

Unlike the RADAR evaluation board, this LiDAR module is not equipped with its own MCU and requires an external power source. Because of this, the LiDAR module must be connected to our designed PCB board in order to receive the required amount of power as well as the software instructions from the MCU.

5.2.3.2.1 Communication Design

There are two major connection configurations for this RADAR module. The first one uses I2C communication, while the second one uses PWM instead of I2C. Once the sensor is triggered the PWM mode will measure the laser distance then output a signal with a pulse that is proportional to the distance measured. The I2C works as expected using a slave/master technique. While the module is equipped to handle both options, the I2C communication will be more useful in this case. Since other components will also be using I2C and the MCU can use that form of communication, I2C will be used. The I2C communication will help perform better fast scanning applications, this could be helpful when there are multiple cars entering the intersection at once. Additionally, a single I2C bus can run multiple devices at once, which important for creating multi-way intersection traffic control devices.

Because this device includes both LiDAR and RADAR sensors, that trigger each other, there must be a way for them to communicate. To accommodate for this, the RADAR evaluation board will process its own data while the PCB that is designed will process the data gathered from the LiDAR. The communication will take place through the MCU on the PCB that we have designed. Both the LiDAR and RADAR will be connected to the MCU where the LiDAR will send its data. If a car is detected by the RADAR sensor for example, the data will be processed on the evaluation board, then be sent back to the MCU. There, the MCU will turn the LiDAR on. The LiDAR will then provide more details on the car's location based on the readings it sends to the MCU.

5.2.3.2.2 LiDAR Drawbacks

While this LiDAR module is the best option of those found in research, there are still some drawbacks that need to be addressed, fixed, and realized before the prototype is developed. While most sensors will be able to detect different types of objects, the shape of the object will affect the accuracy of the results. Unfortunately, most modules have trouble detecting objects on specular surfaces. Specular surfaces are very smooth and as a result reflect energy instead of evenly dispersing it. These types of surfaces make it hard for the sensor to recognize the distance from the sensor to these surfaces. As a result, the reflected beams will be much smaller and could completely miss the sensor, going undetected. The issue here is that most cars have reflective or glass windshields, which are common specular surfaces. In order to prepare for this issue, the LiDAR will be tested multiple times on different types of surfaces to see how the LiDAR is affected and what adjustments to make. One way to account for this issue is to compare it to the data gathered from the RADAR. That way, if the LiDAR is unable to gather data for any reason, the data gathered from the RADAR will be used to determine the approximate location of approaching car. This data can help determine if the approaching car is a threat or not and will trigger the lights to change color.

This device is supposed to work effectively outside in all weather conditions, as a normal stop sign would. The issue with this, is both the LiDAR and RADAR sensors are sensitive to humidity, water, and any other type of intense weather conditions. For example, rain filled with dirt particles (water with a high turbidity) can hinder the dispersion of the wave by either absorbing it or reflecting it off of the water particle. In order to prevent this absorption or reflection, the sensor will be angled so the amount of dispersion is limited. To do this, the sensor must be angled so the laser is viewed from the normal or tangential of the object that it will be trying to sense. Otherwise, the energy will be reflected instead of being dispersed. If the energy is reflected, the waves will not hit the sensor and will produce no readings.

Because the LITE V3 is enclosed in a weather proof housing, the sensor should be somewhat protected from any harmful weather. Also, the placement of the sensor on the device will help protect the sensor and the other electronics from the weather and humidity. In this project the sensors will both be placed towards the lower half of the device or under the device itself. Using this orientation, the solar panel, which is on top, will protect the sensor from weather as well as intense sunlight. While the solar panel will be absorbing a large amount of sunlight, that light absorbed by the panel will not reach the sensor or electronics inside the device. This may give off excess heat, but the fans and heatsinks will help contribute to dissipating this heat so the PCB and other elements do not overheat. Since the housing will be much larger than the components and PCB itself, there is more room for airflow and less areas for heat to collect inside the housing. The placement of the LiDAR will also affect the wave dissipation, and if the module is too low, the wave will not be able to reach the object or reflect off of the object incorrectly.

5.2.4 Traffic Light Signal Design

The output light of our device is meant to give a traffic signal from several meters away. To do this, we used LED light arrays that are compliant with MUTCD traffic standards. The traffic standards ensure that the lights are visible in the day and the night from a range of angles. These lights consume most of the power being used by the device; thus, we used LEDs to reduce power consumption as much as possible.

The traffic lights used for this product are made by Wei De Mei and sold by Amazon. These lights fit the US MUTCD regulations of traffic light sizes as well as the constraints of our project. Traffic lights in the United States are either 200mm or 300mm depending on the size of the intersection. The S.A.S.S. is marketed towards smaller intersections; thus, the 200mm light was used. Because the 200mm light occupies less space, it requires less LED lights inside and consumes less power. A lower consumption of power helped meet the power constraints of the project, making it more self-sustainable. A clear plastic lens covers the LED array and protects the circuit from the elements.

Joining the clear lens to the back of the traffic light casing is a rubber seal that prevents moisture from getting into the device. The life of the circuit can be exponentially decreased when exposed to moisture; thus, extensive waterproofing has to be used to ensure that longevity of the device is preserved. The Wei De Mei light specifies that their LED arrays are capable of lasting over 80,000 hours. Because these traffic lights are constantly on, the LEDs chosen are projected to function for the duration of about nine years.

5.2.4.1 LED Specifications

The traffic lights made by Wei De Mei each have 90 5mm LEDs that fill up the 200mm light. While the circuit could be easily replicated, with time constraints we decided to buy a pre-assembled light. In future iterations of this device, a custom PCB would be made for these to save on the overall cost of the device.

The traffic lights produced by Wei De Mei are specified to operate at 12V and 667mA which would result in an 8-watt power consumption. At 8 watts, this is only a fraction of the output needed to power a comparable incandescent traffic light. Additionally, after testing the LEDs, the actual product only consumed 480mA of current during operation. Having a smaller output current to the LEDs will decrease the overall power consumption of the device and store more residual energy in the onboard battery. These lights are rated to run from a DC power source which was an important feature due to the type of power source being used on the project. Most traffic lights are powered by AC electricity provided by the grid; thus, they include a rectification circuit to convert the AC power to DC. Other devices that include this rectification circuit would have to be modified to apply to our product.

5.2.4.2 Comparable Products

Comparable products are offered throughout eBay, Amazon, and AliExpress. All of these are of similar quality and seemingly under-engineered from China. After narrowing down the light selection to only those that fit our criteria, there were not many options. High quality LED traffic modules are offered by companies based in the United States; however, these companies avoid business with small customers and do not usually provide products on the consumer level. Because the lights were of questionable quality, we decided to source our lights for our project from Amazon due to their fast shipping and question-free return policy. The majority of the LED lights offered from these sellers did not include official datasheets, which further questioned their reliability.
The main lights on the system are controlled via our main board after getting input from the RADAR and LiDAR sensors. The design of the lights is simply a state machine based on the processed inputs from the LiDAR and RADAR which compare the calculated speed, acceleration, and distance values to reference values that are calculated and programmed into the system based on the intersection that the system is placed in. In a production model there would be in total 8 outer lights, and most likely numerous inner LED's that could be switched on using a debugging mode that would show signals inside of the embedded system. This functionality makes testing easier and allows for simpler maintenance later on if the system ends up in a failure or error mode. In designing the lighting system, we are not designing the light module itself since we want to stay compliant with current laws and lights used in intersections within the United States, but we are designing the way they are controlled and how that affects traffic. Table 12 displays a comparison of suitable LED lights for our project.

Table 12: Comparison of LED traffic lights

Product	BBM JD200-3-R	Primematik	LeoTek IL6-P3	Wei De Mei
Price	$45.79	$50.10	Not Available	$39.99
Power Consumption (W)	5W	5.8W	2.4W	5W
Voltage (V)	12-24V	12-24V	10-28V	12V

Ultimately, we decided on the lights produced by Wei De Mei due to their obtainability in low quantities and prices. A highly considered option was offered by LeoTek. The LeoTek IL6-P3 offered a light output high enough to comply to government standards; however, these were not easily obtained on the consumer level. These lights offered a high output while only consuming 2.4W which is half of the consumption of the lights we have obtained. In future production of the device, LeoTek lights would be further considered if the minimum quantity of purchase was reached.

Primematik LEDs and BBM JD200-3-R were both highly considered because they met the requirements of the project and were within budget. Each of these lights fit our power consumption and ran off of DC power, which was a requirement for our design. Due to the price difference, however, these were not chosen to be used on our project.

5.3 Software Design

Since we used a microcontroller as the base for our design, we designed our product to be a full embedded system that would be integrable into currently existing systems. With designing a full software system, you have to first break the design into multiple parts and decide how the software components should first act. Once the design team has decided how the software should interact with the other components, they are able to design the behavior of each component. As this is done, each component gets focused on in the design before ever being written. This allows developers to determine how their software should interact and how it should behave; then finally they are able to determine the inner functionality that must be written to achieve those things. This is done upfront and then reevaluated at each iteration of the software to allow for fluid design and quick bug fixes early on before the entire software has been written.

In the Situation-Aware Stop Signal, advanced sensors and a self-sustaining power system are paired with complex and multitasking software. This software underlies all operations of the Situation-Aware Stop Signal, dictating its ability to interpret sensor data and respond in an intelligent and proactive manner. An interesting challenge confronted by the stop signal was the need to ingest two kinds of sensor data and make sense of it as one whole. This process was a primary concern for the software implementation of the project. Such analysis as well as a consideration for the scheduling of vehicles at the intersection represent the keys to decision-making in the stop signal software.

In this section of the document, a brief overview of the parts and tools selected for the development of software will be given. This will be followed by a comprehensive explanation of the architectural design of the stop signal software, describing the structure of key software components. At the continuation of the section, a comprehensive explanation of the detailed design of software will be shared, detailing the behavior of software components. At the conclusion of these writings, the reader should have an understanding of all software and its implementation. Should this be successful, a reader well-versed in programming should be able to follow the guidance provided herein and write functional software that meets the specified requirements of the stop signal.

5.3.1 Part Selection

Software design for the Situation-Aware Stop Signal included the selection of two primary components: operating system and programming language. These components represent the platform upon which the logic of the stop signal program is implemented. Because of the complexity of the behavior required by the stop signal, careful consideration was made for each and various options were explored. Research pertaining to this exploration of operating systems and programming languages is found in Section 4.2 of this document. After research was performed in both of these areas, final selections were made by the team. Explanations of the rationale behind these decisions are included in the following subsections.

5.3.1.1 Selection of Operating System

Because of a variety of system inputs and the need to process data in real time, it was determined early on that the Situation-Aware Stop Signal software should run on top of a type of operating system, particularly what is known as a real-time operating system or RTOS. Operating systems in general take upon themselves the task of scheduling processes to make possible multitasking in systems where multiple actions need to be taken at once. Real-time operating systems take this further by implementing a strict system of priorities and guaranteeing the deterministic runtime of high-priority processes. More so, many real-time operating systems are made to be lightweight and implemented in embedded systems.

Four real-time operating systems were investigated as candidates for the platform of the Situation-Aware Stop Signal software. They are FreeRTOS, Zephyr™, eCos, and TI-RTOS. These were selected from amongst their peers due to their relevance in modern embedded computing as well as their open source provision. Each of the operating systems was studied and noted for its qualities in the following areas, considered important for the stop signal platform:

- Distribution type and license
- Years in development
- Kernel size
- Scalability
- Power management
- Compatibility

A comparison of these qualities is given in Table 13 below, while an overview in each area is given at the continuation of the section. In each overview, an acknowledgement of qualities desired of the stop signal platform will be made, followed then by an explanation of the qualities of the real-time operating system candidates. Following the overviews, a conclusion is made with justification given for the selection of one of the four candidates.

Table 13: Operating System Comparison

Product	FreeRTOS	Zephyr™	eCos	TI-RTOS
Distribution Type	Open Source	Open Source	Open Source	Open Source
License	MIT	Apache 2.0	GPL	BSD
Years in Development	15+	20+	20	20
Average Kernel Size (KB)	6-12	8	1	Unspecified
Easily Scalable	Yes	Yes	Yes	Yes

Implements Power Management	Yes	Yes	Yes	Yes
Compatibility	Various platforms supported	Various platforms supported	Various platforms supported	Limited to TI platforms

5.3.1.1.1 Distribution Type and License

It was desired that the operating system chosen for the stop signal platform be open source and given under a license that permits free use and modification for both developmental and commercial purposes. This was to greatly reduce the cost of the software system, while facilitating the production of the stop signal should it be brought to market. Open source software is also well-reviewed with many contributors, including large and successful organizations, donating to its codebase. Such usually results in quality products with plentiful documentation.

All four candidates were selected because of their open source distribution. These, however, differ in licensing and corresponding permissions. FreeRTOS and Zephyr™ are offered under the MIT and Apache 2.0 licenses, respectively, while eCos and TI-RTOS are offered under variants of the GPL and BSD licenses. Of the four different kinds of licenses, the MIT and BSD licenses seem to be the least restricting by only requiring the inclusion of a copyright notice in forked software. The GPL license requires the complete disclosure of source code of licensed works and modifications, while Apache 2.0 asks for the statement of changes along with the inclusion of the license and copyright notice. Here, FreeRTOS and TI-RTOS may be the most preferred, with Zephyr™ not far behind. The disclosure of source code required of the GPL license may be a concern in the case of eCos.

5.3.1.1.2 Years in Development

A desired attribute of the stop signal operating system was that it be proven with a long development history. While this is not to say strictly that older software is preferable to newer software, it means to suggest that more mature and tested code is sought as a foundation for the software of the project rather than new code that lacks the tests of time and field experience. Because of the safety-critical nature of the Situation-Aware Stop Signal, the underlying software platform needed to be the most robust and reliable. A record of success was desired of the operating system to fill this role.

The most senior of the four candidates in its code history is Zephyr™. While the project itself was introduced in 2016, it inherits from a codebase of Wind River System's VxWorks that goes back more than 20 years. The next most mature operating systems are eCos and TI-RTOS, both reaching their 20-year milestone in 2018. FreeRTOS is the youngest of the four, began just over 15 years ago.

5.3.1.1.3 Kernel Size

The kernel size of the real-time operating system to be employed by the stop signal was to be small, preferably under 10 KB. Because the stop signal software was to be run on a microcontroller, memory was considered a precious resource. Because of the particular memory constraints imposed by the selected hardware (64 KB of RAM in the MSP432P401R), the operating system could not require a significant amount of system resources. An appropriate operating system was to have a capable yet compact kernel.

The smallest average compact kernel size found of the four candidates is that of eCos at an approximated 1 KB. It distances itself greatly from the other candidates, FreeRTOS having an average compact kernel size between 6 and 12 KB and Zephyr™ having such a size of 8 KB. A measure of the kernel size of TI-RTOS was not found in its documentation. In this comparison, eCos seems to have an advantage over its peers.

5.3.1.1.4 Scalability

In the same sentiment as kernel size, another attribute desired of the stop signal operating system was an ability to scale to the need of the project and reduce the inclusion of unnecessary features. In a system where memory usage is a critical matter, it is important that resources not be spent on unused utilities. The ability to reduce the operating system kernel to a package of only necessary features allows for lower memory consumption and a more optimal software deployment. The size of the operating system becomes directly proportional to the capabilities it provides.

All four candidates support this kind of scalability, or modularity. FreeRTOS provides core functionality while supporting add-on features such as a networking stack. Zephyr™ allows fine-tuning of the kernel by using a utility called Kconfig. eCos allows for similar fine-tuning using a Component Definition Language (CDL) and the management of configuration points across multiple levels of the operating system. TI-RTOS exhibits a modular architecture like the others with its own management of software add-ons. Of the four, Zephyr™ and eCos seem to allow the greatest level of granularity when it comes to the configuration of the kernel itself.

5.3.1.1.5 Power Management

Because the Situation-Aware Stop Signal was to be solar-powered, the operating system running it needed to implement power management features to reduce power consumption when at all possible. This might have implied direct interaction with the low power modes of the MCU or the operating system's own reduction of processing at times of idle. Regardless, the stop signal operating system was to be mindful of the power it consumes and capable of changing its consumption based on activity. This was to result in noticeable power savings in the whole of the stop signal system when compared to the power draw of an always-active MCU.

Each of the four candidates implements power management, being designed for embedded systems. FreeRTOS, Zephyr™, and TI-RTOS all use tick suppression, otherwise known as a tickless idle, to refrain from checking in during idle periods. eCos has limited power management features, supporting four defined modes of operation but with no mention of tick suppression. These findings suggest the use of the first three while introducing a significant concern for eCos.

5.3.1.1.6 Compatibility

A final consideration for the stop signal operating system was that it should be as platform-independent as would reasonably be possible. This means to suggest that the operating system should be compatible with multiple platforms or architectures. While the particular MCU and corresponding instruction set architecture to be used in the Situation-Aware Stop Signal may have already been decided, it was important that the software be portable should any hardware changes be needed and a change in instruction set architecture be decided. By supporting multiple architectures, an operating system allows for the consistent execution of software on various types of hardware.

All candidates except TI-RTOS have a wide gamut of supported platforms and architectures. FreeRTOS supports multiple ARM variants, as well as TI MSP430 - reaching a total of 30 supported architectures. Zephyr supports ARM Cortex-M, Intel x86, ARC, and RISC-V 32, amongst others. eCos supports a variety of popular architectures, as well, including ARM, x86, MIPS, and PowerPC. TI-RTOS supports a number of TI architectures including MSP430, as well as ARM Cortex-M, but is limited to TI MCUs. This limitation puts TI-RTOS behind its competitors and made it a less attractive option for application in the Situation-Aware Stop Signal.

5.3.1.1.7 Conclusion

It was determined that the operating system powering the Situation-Aware Stop Signal should be one that is robust and capable of driving safety-critical software. It should be mature, have a small footprint, and exhibit characteristics that make it scalable, energy-efficient, and compatible with a variety of platforms. After analysis of these qualities in four competent candidates, we initially decided to select Zephyr™ as the operating system to run the Situation-Aware Stop Signal. The selection of Zephyr™ was made with much thought given to the maturity of its codebase and the capabilities that allow it to hold its own amongst its competitors.

When we began using Zephyr™ at the end of the first semester of Senior Design, it became apparent that it did not sufficiently support other design decisions we had made. Specifically, we found that it did not support needed communication protocols I2C and SPI on the MSP432P401R and had limited support for the C++ programming language. Here, we realized we had made more independent decisions in selecting microcontroller, operating system, and programming language rather than considering them as a whole. To correct course, other candidates were again compared and TI-RTOS was swiftly chosen as the operating system to run the stop signal software. TI-RTOS has ample support for the

MSP432P401R and other TI microcontrollers and has extensive compatibility with the C++ programming language.

5.3.1.2 Selection of Programming Languages

The first step of developing the software for a project is deciding which tool is best for designing the system. For a microcontroller, the best way to interface with the device is to program as close to the hardware as possible. Assembly would be the best option for this but, generally, programming a full program in assembly that interprets RADAR and LiDAR data and then uses that information to classify an object is too complex and time-consuming. Because of this, we plan to use a slightly higher-level, compiled language C++. Using C++ is the best option for a microcontroller because it allows for the hardware flexibility of the assembly language but abstracts enough of the hardware so that the language is simpler. C++ was compared with a variety of other languages, the findings of which are found below in Table 14.

Table 14: Programming Language Comparison

Product	C	C++	Cython*	Golang	Assembly
Speed	2	3	5	4	1
Libraries	4	1	2	3	5
Hardware Access	2	3	5	4	1
Zephyr Compatibility	Yes	Yes	X	X	Yes
Object Oriented	No	Yes	Yes	Yes	No
Multithreaded	Not standard	Standard post C++11	Yes, but not C standard	No	No
Personal Knowledge on Language	1	5	2	3	4

In addition to this, C++ has pre-built first-party and third-party libraries. This is probably the greatest benefit to using C++ as it has pre-built functions for many different types of microcontrollers. This gives C++ flexibility and allows for the product designers to worry about coding the main algorithm for the software and not get bogged down in how to exactly implement a queue in hardware or control specific GPIO ports through memory addresses. Doing this lets the programmer abstract the hardware out into classes that will handle the hardware level accesses. Then the upper level control software can be roughly independent of that even if the lower level hardware changes. If those lower level classes change then the system will reliably work.

We chose C++ over C because C++ has full support for the C language embedded within it. By using C++, we add functionality, class structures, inheritance, and simpler data referencing with an object-oriented approach instead of C's common place procedural approach. This gives our product the capability to grow if we decide to add hardware or if the design was to ever be modified in the future.

Using any other compiled language is an option, but generally most other languages are not as efficient as C++ and our engineering requirements dictate that efficiency is key. Golang could be a good option as it natively supports processes and would allow for proper multi-processing on a single core machine. The real-time operating system Zephyr, however, does not support Golang as a compile-able language. This is also an issue with the MSP432P401R microcontroller that we are planning on using which can only be programmed with TI Assembly or C/C++ with TI's custom C/C++ header file that is non-transferrable to Golang.

5.3.2 Architectural Design

Architectural design in terms of software, is the design of how each individual software component interacts with the other components of the system and the hardware. This is typically the first step in software design as the developers need to determine how they want to implement the system on a higher level based on the requirements and constraints on the project. Usually, the best approach for an embedded system is to start from the bottom and work your way up to the main functionality of the software. This approach lets us design our interfaces that let us access the hardware. From there we determine how the higher-level components of the system will access those hardware control methods to achieve our project goal.

The architectural design of the stop signal software describes the overall structure of the different components that are to comprise the software itself. Like other aspects of the project, software may be divided up into separate yet cooperative and functional pieces. Because an object-oriented language has been chosen, it is easy to abstract certain functionality of the software into individual classes. These classes, independent or related, form the components of the software that describe certain domains of functionality.

Software used in the Situation-Aware Stop Signal is organized into multiple components to allow for concentrated development in areas of particular application. An overview of these components will first be made with individual attention given in following subsections. Of particular note is the use of UML diagrams to describe the structure of the software. Here, deployment, component, and class diagrams are used to visualize the various pieces of software, down to classes and their corresponding fields and methods.

5.3.2.1 System Structure

In the Situation-Aware Stop Signal, functionality may be intelligibly organized into areas of operation. Because the stop signal is both a collector and a processor of data, it is designed to operate in a sequential manner (albeit multiple operations may be taking place at once). As would be characteristic of this kind of design pattern, input is taken in, processed, and then followed by a response determined by the system. The stop signal software is organized in this manner, with components dedicated to each phase of data consumption: ingestion, analysis, and response.

In the stop signal software, these phases are represented by the components Object Detection (OD), Object Classification (OC), and Traffic and Light Control (TLC). A fourth component, Low-level Hardware Abstraction (LLHA), is also included to provide wrapper classes for hardware components such as the RADAR and LiDAR sensors. These four components collectively abstract the operation of the Situation-Aware Stop Signal into manageable groups of software.

The software components described correspond directly to the hardware that composes the stop signal. The deployment diagram in Figure 32 illustrates the grouping of hardware into the different software components. It may immediately be made clear that certain hardware pertains to more than one software component. Such is the case for the RADAR modules, which participate in the preliminary Object Detection and continue to report data in Object Classification.

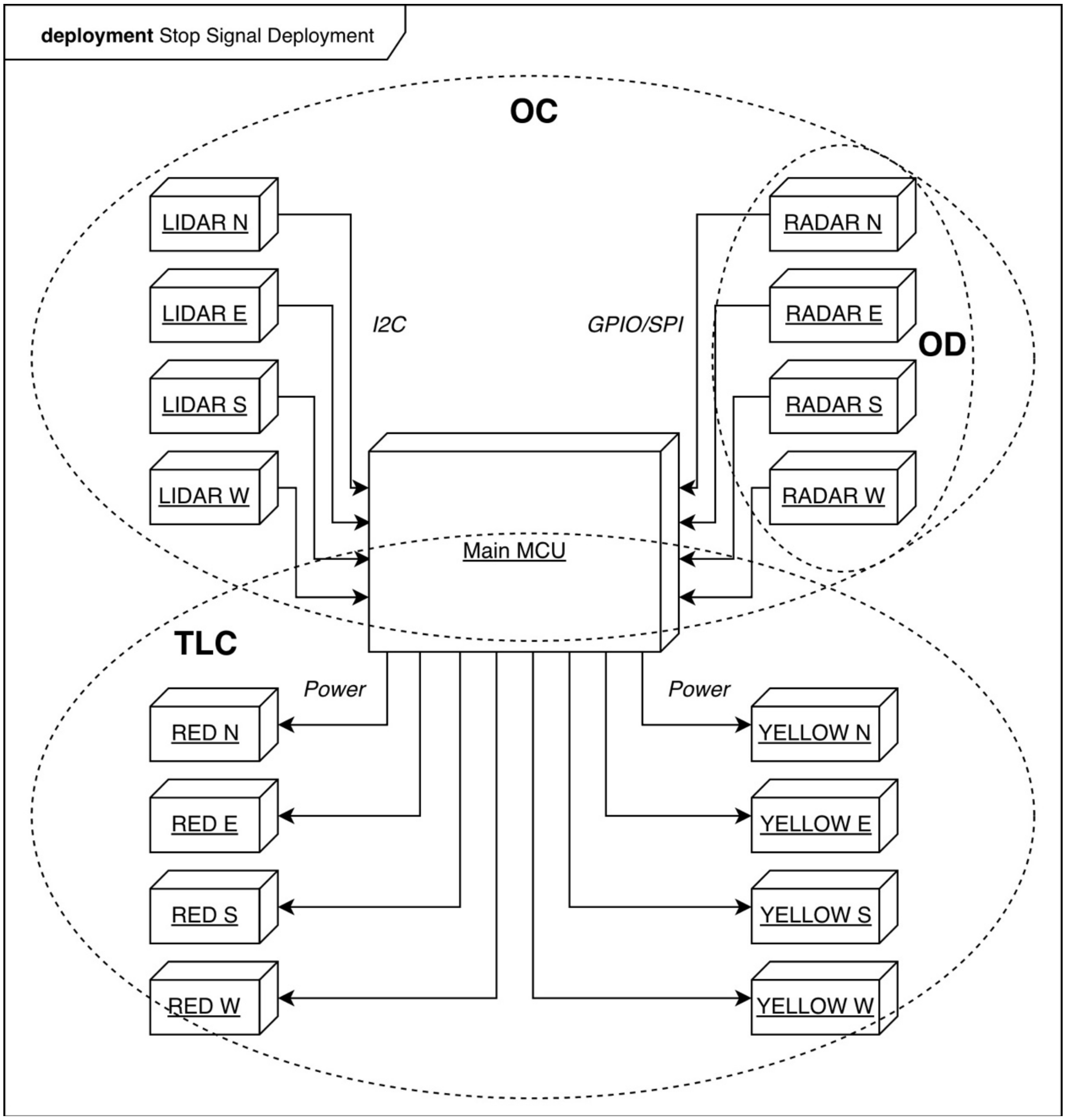


Figure 32: Deployment Diagram

Although Object Detection, Object Classification, Traffic and Light Control, and Low-level Hardware Abstraction represent individual components, these are not isolated from one another and there do exist dependencies. This, of course, must be so due to the nature of data consumption. The component diagram in Figure 33 shows the relationships between the four components of the stop signal software. Of the four components, the only one to be found independent is Low-level Hardware Abstraction. The three remaining components exhibit dependencies on one another or on Low-level Hardware Abstraction.

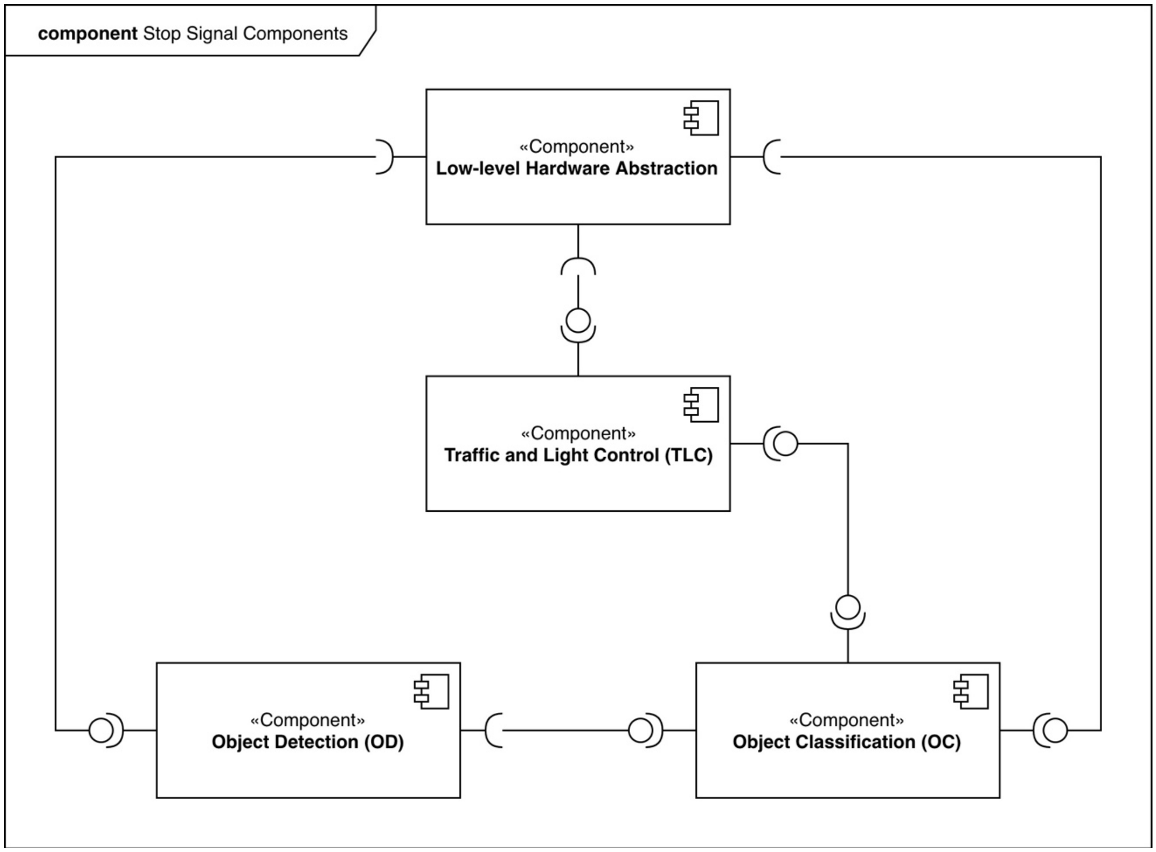


Figure 33: Component Diagram

A review of the relationships between software components allows us to formulate how interactions and message passing will work between the various classes of the components. These classes define the software interfaces available when writing code for the stop signal and each contain fields and methods that define their own structure and behavior. A description of these classes and their fields and methods, with a particular focus on structure, will be given hereafter. Behavior of these classes will be more directly described in the detailed design section of software design.

5.3.2.2 Low-level Hardware Abstraction

In working with an embedded system, dealing with the hardware is probably the most demanding task. For the software side, it is important that we have a convention in place that allows us to abstract the hardware so that we can develop a system that is relatively independent. This makes for good design and allows for flexibility if a system like this were put into place in real intersections.

An embedded system can be coded very specifically using C or C++ and would work well, but many of the pins and how to access the system would be static. Thus, there would be no good way to change the software without going into the code and changing values that are hard coded at the top of a file. Generally, this is not good design as it forces the developer to have to recompile the code after every change. For an embedded system, this has to be done regardless but the code should still be modular and changeable in case the design is to ever be modified. For real life products, modularity and the ability to change as needed are huge benefits to software design since they allow engineers and software developers to change components and code as necessary.

The design of our hardware abstraction layer is based on an interface to sensor modules that then creates objects for the different sensors of the system, illustrated in Figure 34 below. If a component were to be added, then a developer would just need to implement a sensor interface. On interfacing with a component, the communication protocol needs to be defined, as well as GPIO pins and interrupts that the hardware component needs. This will be done for other hardware components, as well, so any software controllable component that we use should have a class or interface that is to be used to help implement that functionality. This gives the embedded system flexibility and also makes the code readable and explicitly clear which is important because safety requires code that can be easily understood so that the design can be verified as needed quickly and easily.

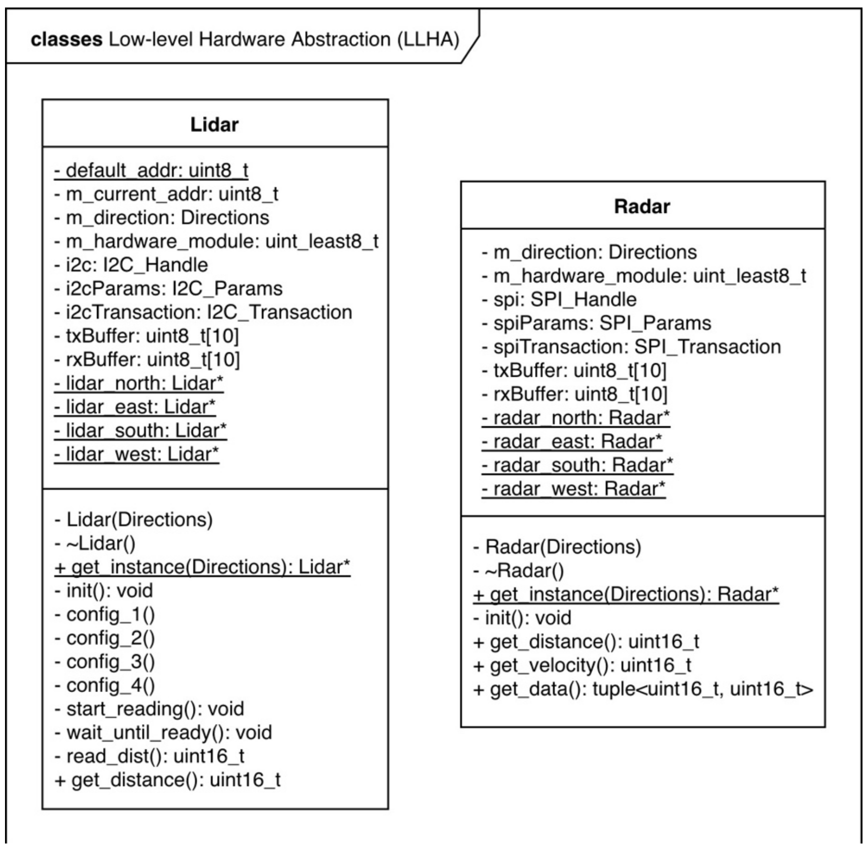


Figure 34: Low-level Hardware Abstraction Class Diagram

For the motherboard of our system, there are multiple modules that control individual components or groups of components. Our design leverages this so that each RADAR and LiDAR module has an object which represents it, and which contains the data it sends to the main motherboard. That abstraction lets us isolate the data based on the direction from which the oncoming vehicle is arriving. Knowing that gives us the ability to control the lights that are onboard the system, as well, since we then know from which side a vehicle is approaching. The lights are configured similarly to the sensors; the lights themselves are the output of our system so we use the information gathered by the sensors to control the power to the lights through the GPIO.

Using this approach also allows us to make another hardware abstraction, albeit not on our system. It lets us create vehicle objects for the vehicles that arrive in the intersection. This lets us monitor a specific vehicle's speed, acceleration, and stop time which gives us all of the information we need to make an accurate prediction of driver safety at any moment in time. More on this will be covered by the detailed design section since this relates to the behavior of the device.

5.3.2.3 Object Detection (OD)

In terms of architectural design, object detection is probably the simplest aspect of the design because this component of the software system is largely done on the mmWave as a pre-processing and power control method to detect moving vehicles. The first detection in OD is done with the IWR1642 mmWave RADAR device. Once a vehicle is detected the mmWave evaluation board will then begin classifying the vehicle as safe or unsafe. Once a classification has been made then the system will send a GPIO signal back to the S.A.S.S. board which will indicate whether the vehicle is unsafe. Then the second phase of vehicle detection relies on the LiDAR device. Once a vehicle approaches the stop bar, the LiDAR will then detect that change and begin scheduling the vehicle once it is stopped. Once the vehicle leaves then the LiDAR is also responsible for understanding that the vehicle is no longer present at the stop bar and relaying that information back to the S.A.S.S. board.

5.3.2.4 Object Classification (OC)

Object Classification envelopes all software that handles the recognition of objects whose presence has been reported by Object Detection. It contains the logic that defines whether or not vehicles are safe at the intersection. This involves consistent measurement of key information, such as velocity and distance, that indicate when a vehicle is no longer within a safe stopping distance. Raw data, notably, is captured through the assistance of Low-level Hardware Abstraction. This and all other pertinent information about objects at the intersection is requested and recorded within designated classes in Object Classification.

The two main classes used in Object Classification are Classifier and DetectedObject, depicted in Figure 35 below. The Classifier class is intended to be instantiated for each side of the intersection and allows access to a list of methods that perform classification and tracking of objects. Used within dedicated threads, it is designed to be used for normal monitoring of the intersection, as well as emergency monitoring during periods when threats are detected. It is directly dependent on the DetectedObject class, itself lending to the creation of a DetectedObject instance when it finds and classifies a relevant object.

The DetectedObject class is really an abstract class, or what may be considered an "interface" in other vocabulary. It is realized and extended by the Vehicle and Other subclasses. These allow for the creation of software objects to represent the real-world objects that the system classifies and tracks. Setter and getter methods are provided to both update and access key information, including velocity, acceleration, direction, and distance. In the Vehicle subclass, additional methods allow for the setting and getting of the time at which a vehicle stops. Other subclasses, in addition to the Vehicle subclass, may be implemented for future types of objects, such as pedestrians and animals.

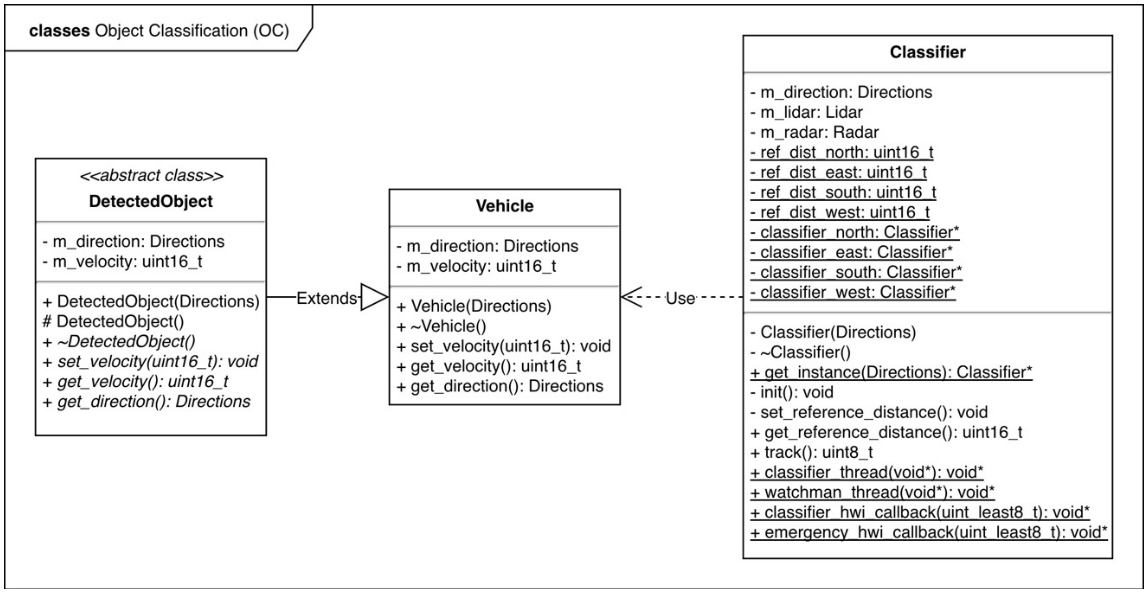


Figure 35: Object Classification Class Diagram

5.3.2.5 Traffic and Light Control (TLC)

Traffic and Light Control envelopes all software that controls the lighting of the stop signal, with and without input from Object Classification. It is responsible for facilitating both the automated and manual control of each set of lights comprising the stop signal system. Traffic and Light Control makes direct use of Low-level Hardware Abstraction to create an accessible software interface to the physical lights. This functionality is contained in a single class that may be utilized throughout the stop signal software.

The main class used in Traffic and Light Control is Scheduler, depicted in Figure 36 below. It is intended to be instantiated once and used constantly throughout normal operation in a dedicated thread. It is directly responsible for managing a first-in first-out (FIFO) queue that receives Vehicle objects, representing stopped vehicles, from Object Classification. In TI-RTOS, this queue is represented by a deque object (standard in C++), allowing for the passing of data between threads and between an ISR and a thread. Methods in the Scheduler class are designed to interact with this queue and use built-in methods in the Lights class to signal one driver to cross at a time or to signal all drivers to stop in the case of a threat. A hardware interrupt is raised when this occurs and is the source by which the stop signal program knows to change its mode of operation. In emergency mode, the Lights class is utilized to turn all lights red until a threat is no longer present at the intersection.

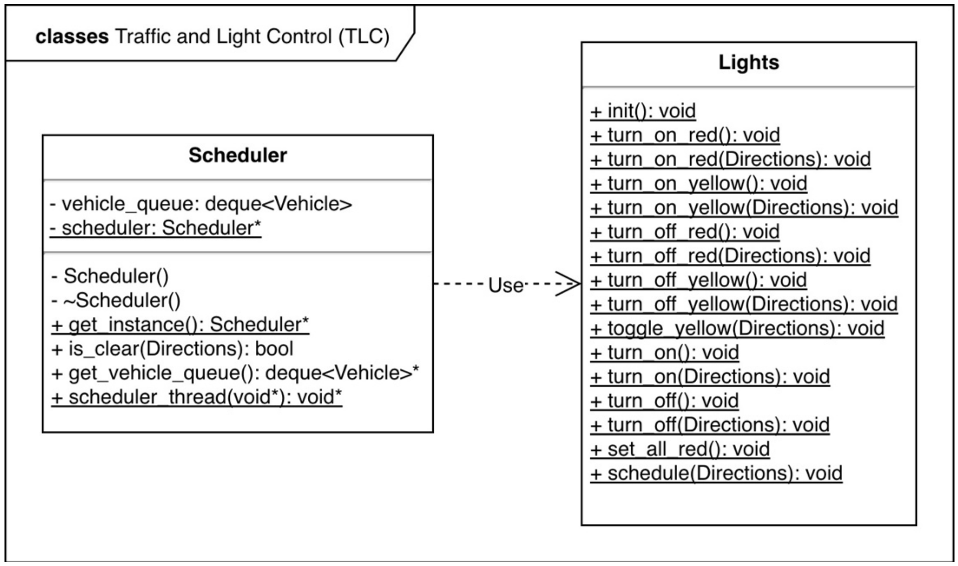


Figure 36: Traffic and Light Control Class Diagram

5.3.3 Detailed Design

The detailed design of stop signal software describes the overall behavior of the different components that are to comprise the software itself. Whereas an overview of the system structure has previously been given, a more detailed explanation of the functions of each component will be given hereafter. Detailed design is intended to expound the algorithms used in project software and explain the decision-making that leads to certain courses of action. Where architectural design gives the developer an idea of what software is to be written, detailed design is documented to give the developer an idea of how software is to be written.

As in architectural design, the use of UML diagrams may be noted. Here, they will be used to describe the behavior of the software. Sequence and state transition diagrams, otherwise known as state machines, will be used to visualize the operation and decision-making of the different software components. An overview of general system behavior will first be given, followed by a detailed explanation of the behavior of each software component.

5.3.3.1 System Behavior

The behavior of the stop signal software is defined by the three primary objectives of the overall system: Prevention of Threat, Protection from Threat, and Scheduling of Traffic. Each software component detailed previously in architectural design is to be implemented in a manner that allows for the efficient and timely realization of these objectives. The segmentation of the software into components and ultimately classes helps isolate the development of algorithms and other complex decision-making that is inherent of the Situation-Aware Stop Signal. These, although developed independently, are designed to collectively bring about desired effects in the system.

Behavior of the stop signal system is based largely on sensor data. Nearly all actions taken are influenced by the information gathered at very short intervals from the array of RADAR and LiDAR sensors on the device. To distribute the work of these sensors and the processing that is needed to interpret their data, the stop signal software follows a "hand-off" strategy. In this approach, data is captured and qualified at different stages. If certain criteria are met, information is "handed off" from one stage to another. Object Detection, Object Classification, and Traffic and Light Control may each represent a stage in the flow of data. These stages and the flow of data between them are best represented by the sequence diagram illustrated in Figure 37 below.

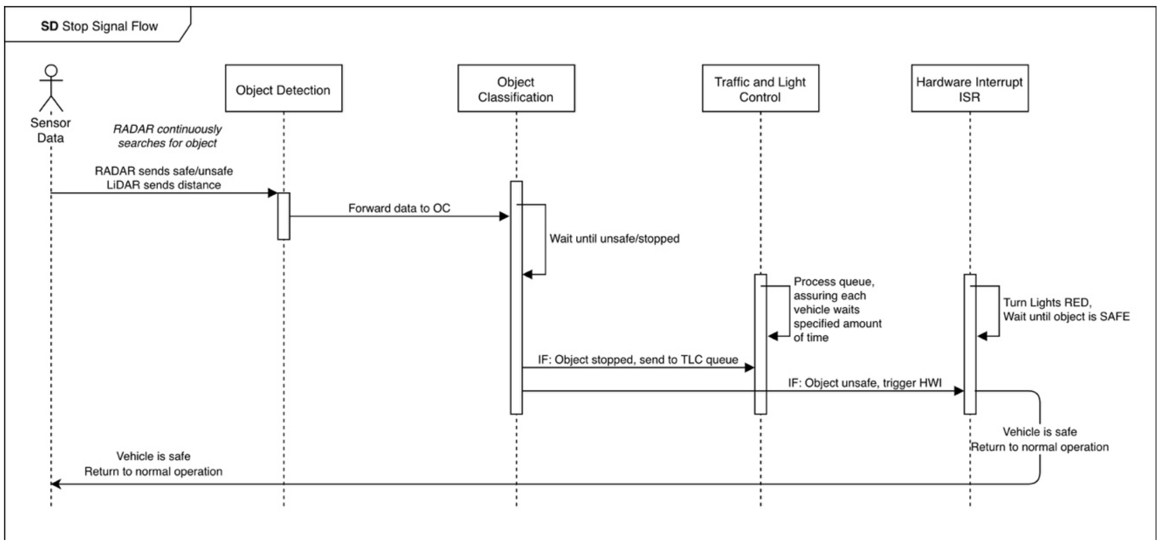


Figure 37: Sequence Diagram

Indicated in the sequence diagram is the origin of sensor data and the subsequent destinations to which it is passed. It may be noted that data is gathered not once but multiple times in the course of the stop signal sequence. It is collected consistently and to varying degrees depending on the presence of objects at the intersection. An understanding of how and when data is collected and passed is critical for proper behavior of the system.

Sensor data is first gathered by RADAR and LiDAR and sent to Object Detection. Object Detection continues to request data from RADAR and LiDAR until qualifying moving and/or stopped objects are found. Because of the variability of time between object appearances at the intersection, software has been designed to allow for both shorter and longer polling periods. This could serve as a means of saving power for when such is possible.

When Object Detection finds an object from an analysis of the data provided by RADAR and/or LiDAR, it hands off execution to Object Classification. During Object Classification, both RADAR and LiDAR are used to provision data for the classifying and tracking of the object. Here, moving vehicles are classified as being safe or unsafe and stationary (stopped) vehicles are monitored until the continuation of their journey. Sensor fusion was originally intended to be used here to allow redundancy in data collection. This would have used data provided from both the RADAR and LiDAR and considered both to make a more accurate analysis of the current situation for every given object. Instead, due to design constraints imposed by the selection of the LiDAR sensors, RADAR is uniquely dedicated to track moving objects while LiDAR is used to monitor those that are stationary.

When the vehicle comes to a stop, data and execution is handed off to Traffic and Light Control. Traffic and Light Control manages the vehicle for the remainder of its presence at the intersection, ensuring a timely stop and the safe continuation of its travel. Here, the previously passed-in data is used to determine which vehicles are present and the order by which to schedule them. Data sent from Object Classification may be continuously added to the queue managed by Traffic and Light Control as vehicles continue to stop at different sides of the intersection.

Because of the need to process data captured from all sides of the Situation-Aware Stop Signal, multithreading is used. Multithreading allows the sequence depicted above to be followed on each side of the intersection, with Object Detection and Object Classification being performed independently for each direction of traffic. A dedicated thread for Traffic and Light Control, represented programmatically by an instantiation of the Scheduler class, is shared between multiple OC threads. This includes the data comprising the FIFO queue from which traffic is scheduled. TI-RTOS, the real-time operating system serving as the platform for the stop signal software, makes this multithreading and data passing possible within the embedded system.

5.3.3.2 Low-level Hardware Abstraction

The detailed design of the hardware abstraction is directly tied to the hardware of the microcontroller and our PCB. As shown in the architectural design section, data is separated out into the correct classes for each major component. Each of these components needs a way to relate to the hardware which requires addresses, pin outs, and low-level control signals. For our design, we created low-level drivers for the LiDAR and RADAR systems to create a custom data protocol so that we can easily transfer information from the sensors to our board. This mainly consists of header files that are written in C (for compatibility reasons).

Once the header files were complete and valid for our board and our sensors, we were able to move on to making getter functions for the data for each sensor. These getter functions were written in C/C++ but, since they are directly acting on the hardware, the easier approach for these was to use non-object-oriented software design so that the hardware can function procedurally. All of these functions are very specific to our board and each and every sensor and light block. Each LiDAR sensor is programmed with a specific address, and each RADAR sensor is tied to a GPIO pin with a HWI assigned to it.

We then created setter functions to control lights and also set configurations on our external sensors. The setter functions are used on device startup so that each device can be automatically configured to the optimal settings for the device’s environment. This lets the program have multiple configurations for different physical setups. This gives our software added robustness because we can change the configurations of the device during runtime as long as it fits into the overall state-machine.

With those in place, higher level methods can call these functions to then gather and interpret data from the sensors with an object-oriented approach. This was done through interfaces and abstraction, as discussed, to guarantee that our main software component only needs to receive distances, speeds, and accelerations. This makes the design of our main software component more manageable since it is also abstracted away from the hardware to a certain degree.

LiDAR specifically reports distance data to the object that it detects, so our design is focused around being able to get that data (accurately) within a quarter of a millisecond to guarantee enough time for the microcontroller to interpret the data involved. Since the LiDAR sensor is only part of the system, we made comparisons at the lowest level possible to guarantee that the data is consistent and safe to use. This was done in tandem with the lower level value smoothing that is used to get average values. This, though, increases our latency, so we used ad-hoc strategy to make sure that the timings and distances we are calculating are safe. This had to be done in a way so that it is accurate because even half of a meter of error could cost someone their life in a situation like this.

The low-level abstractions manage the connection interfaces, signal controls, and GPIO pins. Those functions completely handle those interactions so that they can be modified without affecting the higher-level software as long as the lower level functions still return the correct values.

5.3.3.3 Object Detection (OD)

The object detection module is separate from the rest of our system. This is because we are using the mmWave evaluation modules and they are programmable. So, what we had to do was program them like we would program a custom RADAR module. Using our own libraries, we interface with the RADAR sensor and do preprocessing to ensure that we can send only the data we would require for the software located on the main board. This includes a distance and velocity.

We do most of the calculations on the main board since it is more power and time efficient to do so. This process could be made easier by using a small CPU instead of a MCU but the power efficiency of this device would be degraded in that case and may not run strictly on solar power.

Object detection is a relatively simple function. Since RADAR collects a range of data in the form of a large array, we use linear algebra to calculate the gradients of the RADAR data. Those gradients are then used to calculate the average velocity of an oncoming vehicle which can then be pre-processed and used to classify each vehicle. For us, vehicle detection starts at 25+ meters, so at a range of 25 meters our device should be able to detect a new moving vehicle and begin its classification to determine whether it is safe or unsafe. Object detection itself is primarily performed by the RADAR module, as it is the first to observe approaching vehicles and acts as the first line of defense. LiDAR is used in a second stage of Object Detection when a vehicle stops at the intersection.

Since the majority of object detection functionality is being ran on the RADAR submodule (in this prototype the mmWave evaluation module) there was little need for extra software complexity. Because of this, the mmWave boards were programmed using procedural C code that manages the entire mmWave system. For the mmWave, we consistently check to see what vehicles are approaching the intersection. Then each vehicle that we find we then classify as safe or unsafe. Once that is done the system sends logged data back through UART that may then be used to visualize the data on a computer.

The software onboard the mmWave evaluation board is independent of our other system, but it is designed so that we are able to analyze and post-processes data inline with collecting the data we need for the system.

5.3.3.4 Object Classification (OC)

Object Classification performs the majority of data analysis in the stop signal software and may be divided into two modes of operation: one normal and one for emergency circumstances. In normal operation, Object Classification takes place for a specified direction or side of the stop signal. It is entered after an object is detected in Object Detection and classifies and tracks the object until it is stopped. In emergency operation, Object Classification is entered after an object is classified as a threat. In this mode, it takes a round-robin approach and polls each side of the stop signal until it determines that a threat no longer exists. An overview of this operation is described by Figure 38 below.

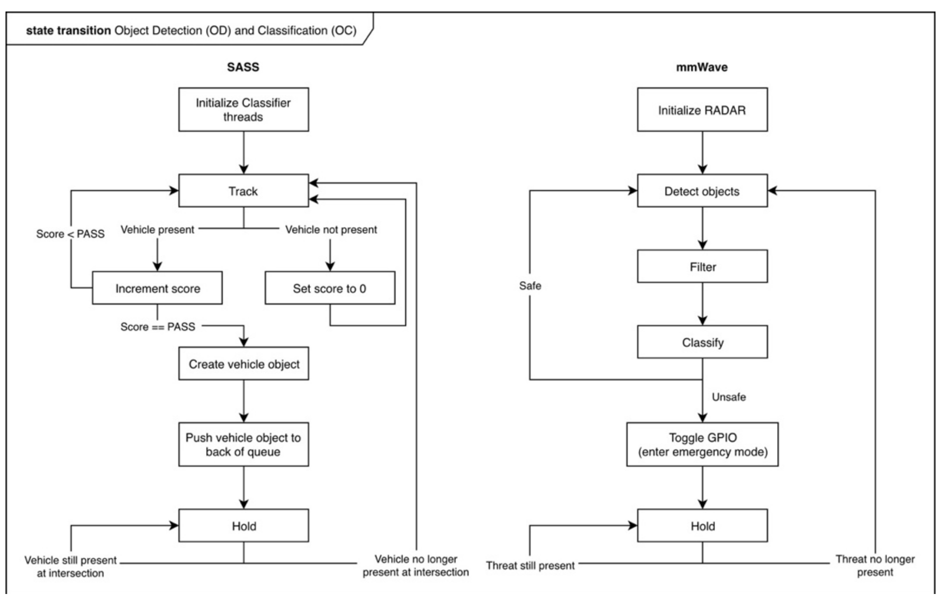


Figure 38: Object Detection and Classification State Transition Diagram

As depicted in the state transition diagram, the normal operation of Object Classification involves the instantiation of a Classifier object. A constructor that receives an enumeration argument is used to specify the direction in which classification is to take place. Use of this constructor calls the main track() method that handles normal operation. When this was first designed, initial classification was intended to determine and return the type of object that exists. That object was then supposed to be passed to a method that was to use methods built into the object to track its velocity, acceleration, and distance. This design change in the course of the implementation of the stop signal software. When it was determined the LiDAR sensors would only be able to be used to monitor stopped vehicles, Object Classification on the stop signal side was limited to supporting only this kind of tracking. Classification of moving vehicles was left to the mmWave RADAR software, an undertaking of its own.

In the event that an object, or vehicle, is found to be a threat during its tracking in Object Classification, a hardware interrupt is thrown in order to stop all traffic and similar tracking routines are used to monitor each side of the intersection. Indeed, the entering into this emergency mode represents a different kind of classification approach, one no longer focused on only one side of the intersection but all sides. In this mode of operation, sensors on each side are polled and a safety counter is used to ensure that a threat is no longer present in any direction. One point is given for each safe side, with the counter being incremented until a safe score is met. The safety counter is reset to zero and sensors continue to be polled until this condition is met.

Both the normal and emergency modes of operation were designed to make use of the same code to enable the monitoring of vehicle presence at the intersection. This reuse, particularly preserved by the consistent operation of the mmWave tracking software, allows for efficient programming and leads to the concentrated development of the classification and tracking algorithms. Those written for the mmWave sensors are largely based on kinematics and utilize values for velocity, acceleration, and distance to determine whether or not a vehicle is coming to a safe stop. These values are used to define a safe stopping distance for each vehicle from the intersection which serves as the mark by which that determination is made.

5.3.3.5 Traffic and Light Control (TLC)

In Traffic and Light Control, a simple algorithm is followed to facilitate the scheduling of traffic at the intersection. All of this is embodied in the Scheduler class and the FIFO queue that have previously been referenced. When a vehicle is found to be stopped during the previous stages, a vehicle object is then added to the queue managed by the Scheduler thread. The queue is emptied on a loop, with the front of the queue being removed only when it has waited the predetermined hold time. As this condition is met, the head object is removed from the queue and the lights are set to give the favored direction the yellow go-ahead and all others red. After a short duration of time, all lights are turned red and the queue is again checked for the processing of the next vehicle in line. If the emergency hardware interrupt is raised from a threat being detected at the intersection, or if the queue is empty, the lights will remain red. This operation is depicted by Figure 39.

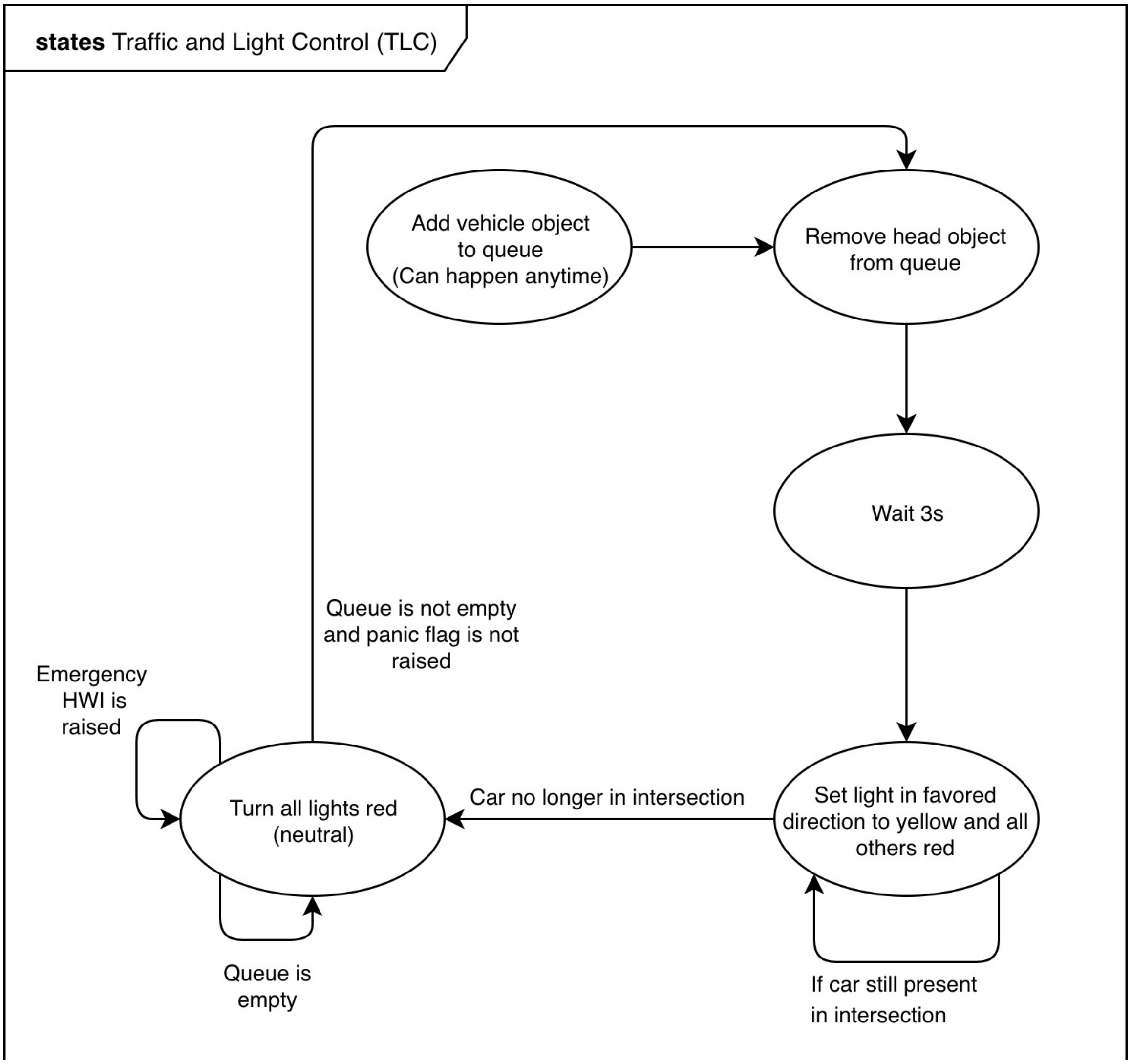


Figure 39: Traffic and Light Control State Transition Diagram

The simple behavior of Traffic and Light Control is largely due to our use of the FIFO queue as the data structure by which traffic is scheduled. The FIFO queue was chosen because the prioritization it defines is most resemblant of the rules and regulations of existing traffic law. Current traffic law states that the first one to arrive at the intersection should be the first to cross. This is what most often occurs at small intersections, the exception being when two vehicles stop at same time. When this occurs, current traffic law states that the driver on the left should yield to the driver on the right. Use of the FIFO queue lends itself to the behavior first described and essentially eliminates the need to reproduce the second. Because all vehicles that are added to the queue will inherently. Have an ordering since the first object in the queue will have arrived first, even if by nanoseconds.

5.4 Mechanical Design

This product is designed to be outside indefinitely in the high heat of Florida, which can at times be over 100°F in the summer. Batteries, microcontrollers, and many sensors do not operate at their peak performance in high heat conditions. Florida is constantly experiencing heavy rainstorms in the summer season, and most electronic components are be permanently damaged upon being exposed to water or excess humidity. The mechanical design of this project allows this device to be constantly outside in the harsh heat and weather conditions that Florida experiences. The design of this project follows the blueprints outlined in Figure 40.

5.4.1 Location

This device is going to be located at a central location in an intersection. This cut down on costs of production and eliminate the need for communication between multiple units. Similar technology could be implemented by having one light in front of each lane; however, this can create unnecessary delay between communicating devices. To mount our device for effective and quick prototyping, the device is be mounted on a post in the center of the intersection. Because we are using visual sensors, it is important for our device to be stationary in the middle of the intersection. The environment this device is being built around an urban environment with building surrounding the device. If this device was implemented in an area less building coverage, a more concrete support may be more ideal. An engineering schematic of the integrated Situation-Aware Stop Signal system is provided in Figure 40 below.

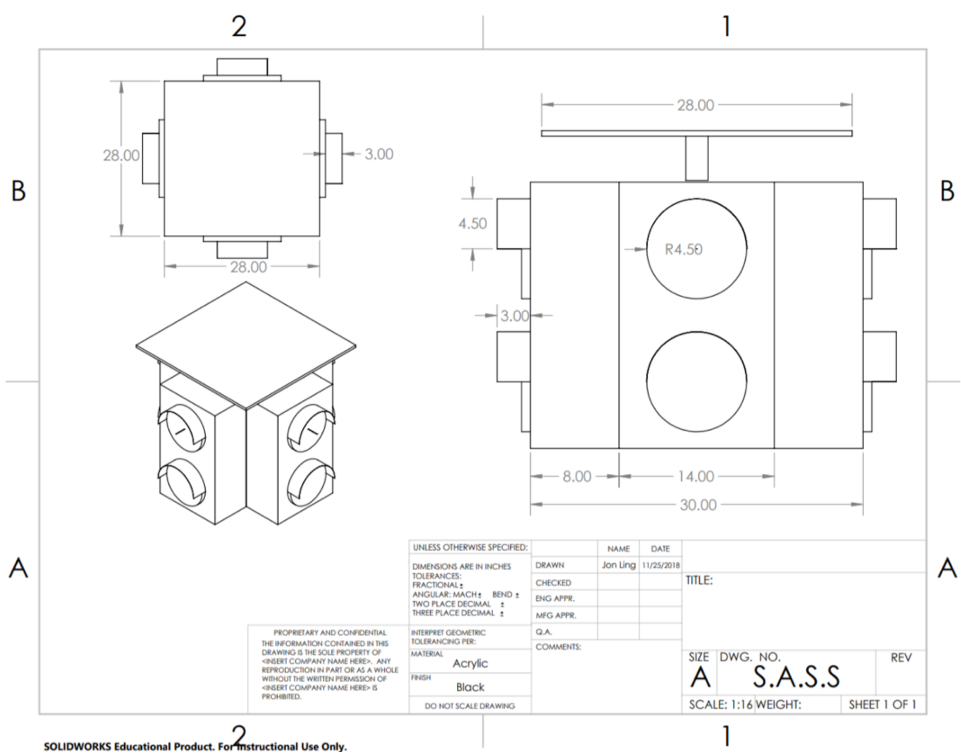


Figure 40: 2D Drawing of S.A.S.S.

5.4.2 Building Material

The prototype for this device is constructed out of laser cut underlayment plywood. In a future version of this, our device could be completely assembled out of steel or aluminum. Due to the lack of access to large machine equipment, this is not an achievable goal given our time constraint. Instead, with access to the Texas Instruments Innovation Lab at UCF, we used the laser cutter to cut out pieces of acrylic plastic. Acrylic plastic is cheap and laser cuts easily without cracking, melting, or releasing toxic fumes. A comparison of the materials considered is given in Table 15 below.

Wooden supports that are 2x2 inches are be used for the frame of the device to add structural rigidity. This support wood is cheap, heavy, and easy to work with compared to other metal building material. Wood is available in local hardware stores which makes the prototype assembly obtainable. Because 2x2 support wood is heavy, it was only be used for the framing of the device and other panels were be laser-cut out of plywood.

Table 15: Comparison of Building Materials

Material	Wood	PLA plastic	ABS plastic	Acrylic plastic	Steel
Able to be laser cut	Yes	No	No	Yes	No
Able to be 3D printed	No	Yes	Yes	No	No
Waterproof	No	Yes	Yes	Yes	Must be coated
Melting Point	3500°C	150°C	105°C	160°C	1370°C

Other materials considered were acrylic, PLA, and ABS. Using sheets of wood would allow for an easy assembly and result in a strong building material. Because the device we are constructing is placed outside, the durability to rain and moisture was a concern. Acrylic was the desired choice; however, because of the price difference between plywood and acrylic, this option was discarded. ABS and PLA are commonly used in 3D printers and were highly considered for the physical design on the project. 3D printing would allow for a variety of shapes and bends in our design; however, keeping a square prototyping device allowed us to avoid 3D printing. 3D printing is time consuming and could take days to complete large prints. Our device is several feet in length, width, and height. Because our device is able to be assembled using flat sheets of plastic, 3D printing would not be necessary. Minor physical components, however, were made from 3D-printed PLA plastic. This gave us the flexibility to create designs in SolidWorks and print out the designs through a 3D printer.

The components inside of our stop signal amount to less than 5lbs. Our building material and design were considered when putting our device under the load of the components. Most of this weight can be attributed to the lithium ion battery and the solar panel charge controller. To support this weight, we constructed a frame out of wooden supports. These were screwed together and reinforced with wood glue. The frame was then be covered in laser cut, wooden panels. Wood can be laser cut, which allows for quick prototyping that can be easily and accurately replicated. The device is square and symmetrical, so having the ability to cut the same design over several sheets of wood allows for a presentable physical design that does not require excessive time, labor, or cost.

Our product is outside in Florida's tropic weather, which can bring large rainstorms. These storms can be damaging towards the electrical components inside. The waterproofing of our device was a concern; however, because our focus was to demonstrate the electrical and computer engineering aspects of the project, waterproofing was not a large concern. Figure 41 exhibits a rendering of what our device looked like in its final stages of design.

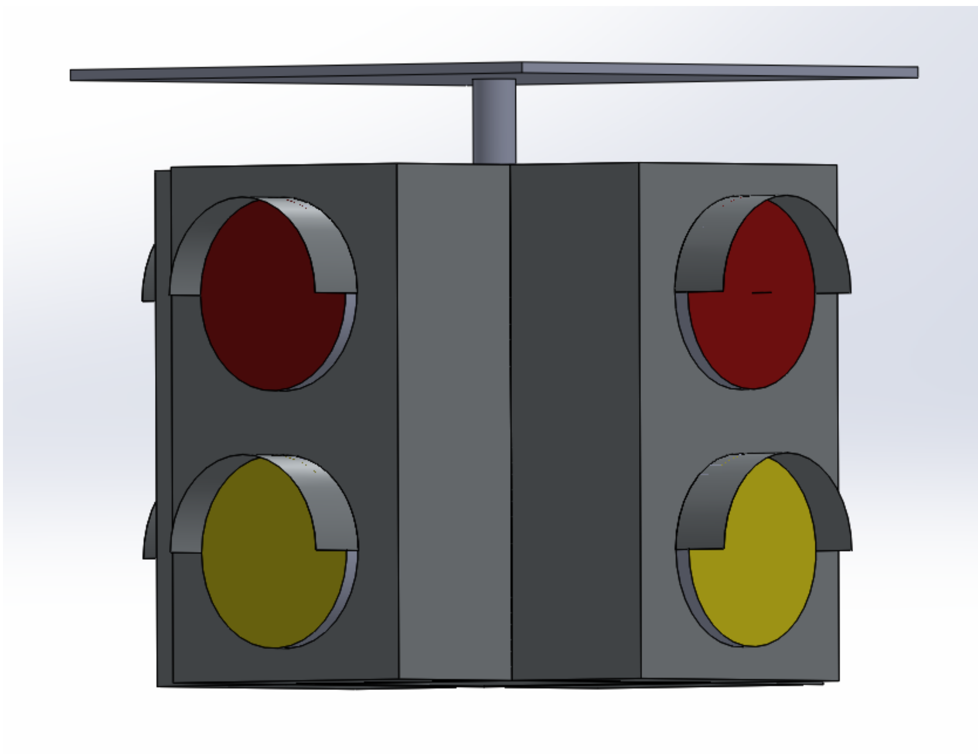


Figure 41: Rendering of S.A.S.S.

5.4.3 Modular Design

The physical design of our device is modular to allow for easy adaptation to any intersection. A central unit holds the PCB, charge controller, solar panel, and battery. From here, the unit sends out signals and processes to each subunit to gather data and output traffic signals accordingly. Each subunit contains a mmWave RADAR module, a LiDAR module, a red traffic signal LED, and a yellow traffic signal LED. These subunits communicate with the central unit. The centralized unit has support for two to four subunits to optimize the device for the specified intersection. Dimensions for all units to be integrated in the Situation-Aware Stop Signal are detailed in Table 16 below.

Table 16: Physical dimensions of S.A.S.S.

Unit	Length (in)	Width (in)	Height (in)
Central Control Unit	14	14	24
Traffic Sub Unit	14	8	24
Solar Panel	28	28	4.5
Total	30	30	28.5

The overall physical dimensions of the device were modeled after existing traffic lights. The central unit is rectangular and extends the length of the assembled device. The length and width of this object is square, extending 14 inches in both directions. This allows for an ample amount of space to store our battery, charge controller, and PCB while creating a symmetrical device. The central unit only has brackets, allowing for the traffic subunits to be attached with connectors which include necessary inputs and outputs of the subunits.

The central control unit for our device is able to accommodate four subunits. Each subunit includes a mmWave sensor, a LiDAR sensor, and the traffic light LEDs. This was the ultimate solution to cut down on unnecessary costs. The modularity of the device allows for a lower budget execution in smaller 2 or 3-way intersections. Each subunit covers the length and height of the device and have a width of only eight inches.

5.4.4 Solar Panel Mounting

The solar panel which is powering the device is mounted at the top to have the best exposure to the sun. The solar cells are between a sheet of glass and a laser cut wooden frame which is both waterproof and transparent. Overall, the solar panel is the length and width of the device.

Connecting the solar panel to the device is a 3D printed adjustable u-joint like hinge. This allows the solar panel to be aimed in the direction of the sun for optimum solar power generation. In future renditions of this project, the panel would be configured to automatically move towards the sun by gathering data from photodiodes and moving a motor accordingly.

6 Testing and Operation

In order to create a workable and practical device that can be used on intersections in the future, it is important to test both the hardware and software in order to ensure the reliability and accuracy of this device. Since this device involves traffic management, it is crucial that each part and the entire system functions without a hitch to ensure the safety of both drivers and pedestrians on the road. As a result, each major component of the PCB and other hardware was tested multiple times and inspected for faults. Because this device is being created specifically for outdoor environments in Florida, tests will also be performed to test the components under intense heat and humidity.

6.1 Hardware Testing

The best way to test a system of electronics is by partitioning them or separating them into different modules. Here, testing related parts, unrelated parts, and how they interact becomes simpler. When partitioning, it is also important to test the communication between different modules. Another way to improve system testing is adding test functionality and designing the system for optimal testing. This helps the designer easily test the entire system as well as the software and hardware implementation. Partitioning also allows the designer to perform different types of testing with more ease. In this project, the two main types of testing being considered are usability tests and critical components tests. Usability tests help assess the usability and functionality of the components in a specific system. Critical components tests are usually done prior to the completion of a prototype. Some examples of this type of testing include: high temperature or humidity operation, UV exposure, RF emissions, and electrostatic discharge.

Many of the hardware components for the Situation-Aware Stop Signal have been acquired during the design phase to accelerate testing of the system. These components are shown in Figure 42.

Figure 42: Acquired S.A.S.S. Hardware Components

Amongst the parts purchased and delivered during design and prototyping are two TI IWR1642 mmWave RADAR sensors, one Garmin LIDAR Lite V3HP LiDAR sensor, two MSP432 LaunchPad development kits (one illustrated), one PowMr 30A MPPT solar charge controller, 28 SunPower C60 solar cells, and one Wei De Mei red LED traffic light. Figure 43 illustrates the use of the MSP432 LaunchPad and red LED traffic light for basic light control testing.

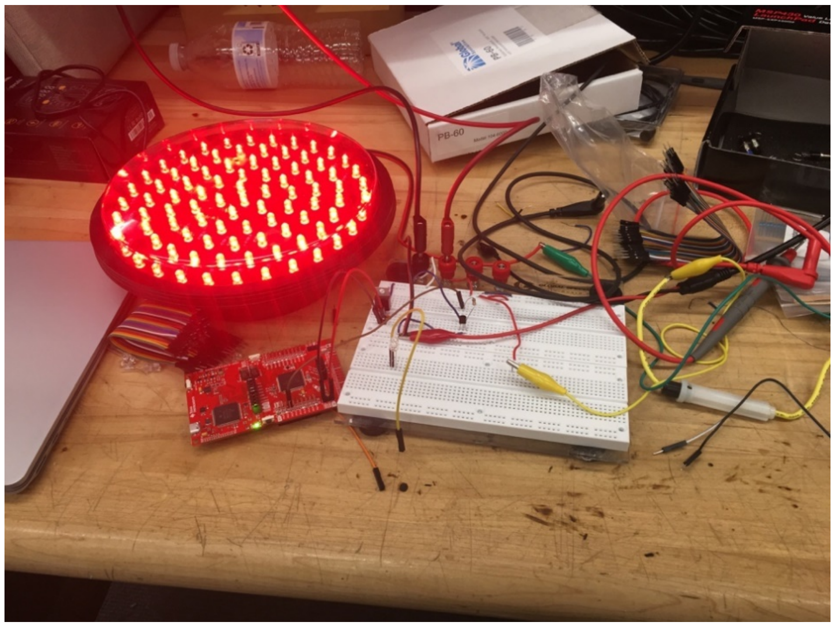


Figure 43: MSP432 MCU Light Control Testing

6.1.1 LiDAR Testing

Because this project relies on both LiDAR and mmWave sensors for traffic management, making sure the sensors are working as accurately as possible is crucial to the success of the design. The most important thing to test in both mmWave and LiDAR sensors is the shape of the signal they produce. If the output produces an unexpected shape or two competing signals, it is apparent that there are issues in the sensor. It is also important to determine the bandwidth, sampling rate, and response time of the sensors. For example, if the sensor does not have a big enough bandwidth, the signal may be distorted for larger distances. These measurements will be done using an oscilloscope. While an oscilloscope will be used to test the sensor's reaction to different stimulus, it is important to note that the sampling rate of the oscilloscope tends to be quicker than that of many sensors. To overcome this issue, oversampling can be used to get a more accurate measurement of the sensor's output. While these sensors require the same basic testing using an oscilloscope to determine the output of the electrical signal each sensor will produce, there are more specific tests required for each type of sensor. A preliminary test of communication with the LiDAR sensor is illustrated below in Figure 44.

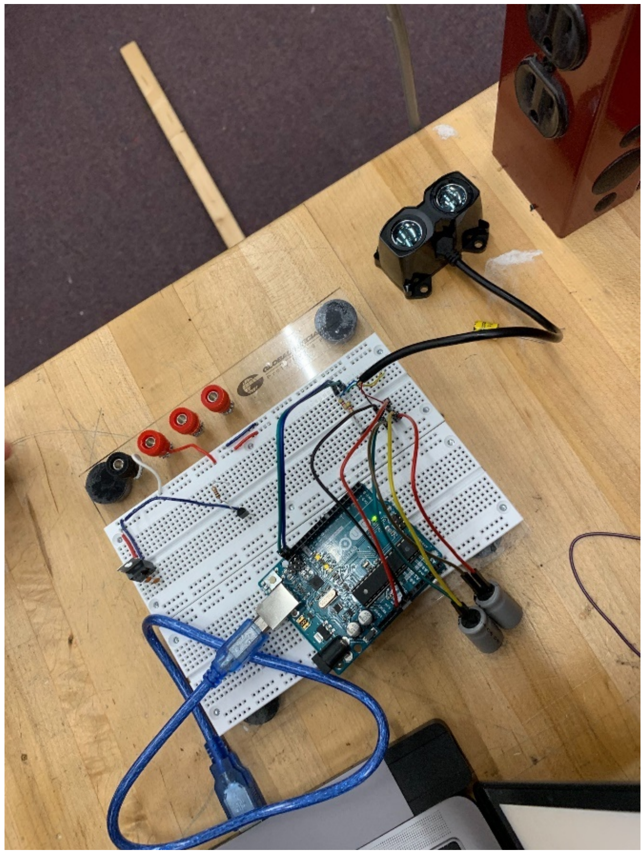


Figure 44: LiDAR Breadboard Testing

6.1.2 RADAR Testing

When testing RADAR sensors, target generation is both a common and valuable test to ensure the sensor is functioning properly. Depending on the specific type of RADAR sensor, (in the case of this project, a mmWave sensor was used) the placement and type of target object generation varies. In target generation testing, the goal was to present the sensor with a target to verify that the sensor sees the target properly. Because this project requires the sensor to detect large cars moving between 15 and 30 miles per hour, it is important to present the sensor with a similar target to a car or motorcycle in both size, speed, and shape. This target generation tests the wave transmission and reception chains of the sensor, which determine the functionality of the sensor. Because this project's only source of power is from the solar panel, the actual power required, and transmitted power must be determined for this sensor in testing. In order to determine the power required for this sensor, the antenna gain should be known as well as the area of the sensor itself. It is important to note that the transmitted power is proportional to the received power and the area of the antenna. Determining this received power can also help determine the signal noise ratio. Since one of the major requirements of this project is the accuracy of the Situational Aware Stop Signal, understanding how much noise and where it is coming from will help to understand how to limit this noise. In RADAR sensors, most noise comes from background radio frequency and the electronic system, for example, the PCB.

6.1.3 PCB Testing

When testing the PCB, there are many common tests that were performed to test each of the components, the stress on the system as a whole, and design testing. When testing the individual components, both resistance and capacitance tests were required. In resistance testing, the resistance of the system or net resistance is tested. It is important to know, when designing a circuit, that longer wires will have more resistance. The change in resistance may affect a large-scale circuit by adding more resistance than required to the system. The goal of a capacitance test is to look for shorts on the board. This test involves charging a network of connected components on the board and measuring the induced capacitance in each network of components. While both these resistance and capacitance tests are common, both may produce inaccurate results due to the variety of styles in PCB board production.

Another important PCB test is comparative testing, this type of testing uses a “master” board or previously tested standard board to compare the current board to. This “master” board can also be used to test other hardware components, such as sensors, to make sure that they work before testing the PCB board. In the case of this project, this testing method was beneficial for both PCB design and understanding how other similar boards work.

Continuity testing ensures that there is the right amount of resistance between every point. Usually, test points are chosen, and their resistance is verified using a multimeter. Many multimeters have a continuity setting and are able to check if there is a proper connection between two points. The multimeter can also be used to check resistances throughout the circuit. This test made sure that there is a continuous path between all necessary points and that the resistance between those points doesn’t exceed the maximum resistance limit. This type of testing also requires both short and open circuit tests, which can also be performed using a multimeter.

Adjacency testing is another method of testing for shorts in the circuit. This test checked for isolation between two conductors. This test uses a software program that checks that certain networks on the board meet the tolerances specified by the user. Two common types of adjacency testing are proximity testing (testing conductors in close proximity) or line of site adjacency (testing conductors in a line from corner to corner).

Run-in or burn-in testing is a type of stress test performed by most manufacturers. This test stresses circuits before they are put in the field and can help manufacturers find faulty hardware. Even though this test is usually done before the board is delivered, there are many open source options of different run-in programs that could be used to double check the tests performed by the PCB manufacturers. This software ran a series of adjustable wrapper scripts that allowed us to build our own tests.

While many PCB manufacturers perform their own tests, as stated above. It is always better to be on the safe size and perform similar tests to ensure they have tested the board or find an error if the board was not tested. Testing an already developed PCB board can be tricky;

however, since the PCB board was designed for easy testing before it is developed, testing was be much easier and more efficient.

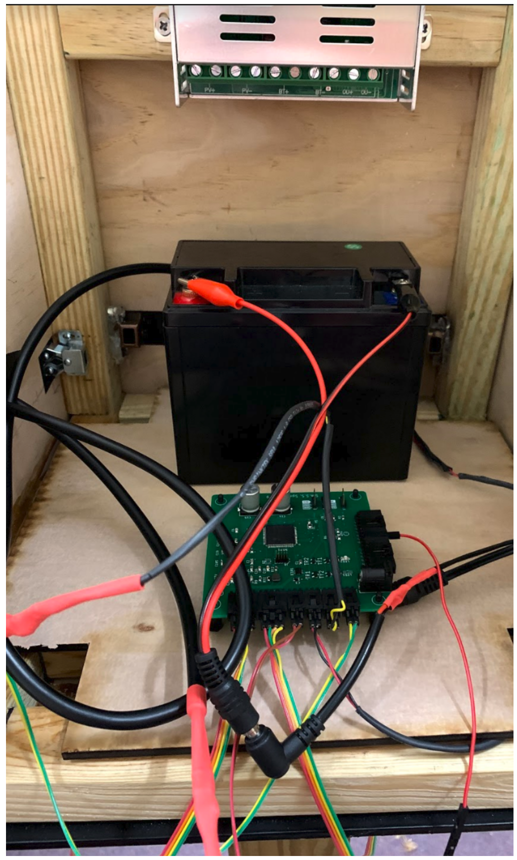


Figure 45: PCB Testing

6.1.4 Solar Panel Testing

Since this project relies solely on the light energy gathered from the solar panel, these panels are an important part of the project and must be tested to ensure that they are used to their fullest potential. Solar panels should be tested outdoors under a significant amount of sunlight to ensure light is reaching the solar cells, as is depicted in Figure 46 below. Because power depends solely on current and voltage, these are the two factors to determine when testing a solar panel. Both the short circuit and open circuit test will also be used to determine the output of the solar panel.

When performing both the short circuit test, it is important to set the multimeter to approximately 10 Amps. When the solar panel is under direct sunlight, using MC4 connectors, connect the solar panel only to the multimeter. This will produce a short circuit current. The same can be done to find the open circuit voltage. These readings should correspond to those in the datasheet. If they do not, the junction box can be tested using the same test and a multimeter. To find the power, just use the equation below and compare that value to the datasheet as well to make sure the readings are accurate.

Another test that must be performed on the solar panel is finding the operating current. This current will connect the charge controller to the solar panel and battery and must be within an acceptable range to prevent overheating or shorting out of the circuit. In order to test this current, the panel must be properly connected to the regular and the battery. The panel also must be under direct sunlight. Using a multimeter, break the connection between the controller and the battery, and measure the current that the panel and controller pass through the battery. It is important to note that if a low reading is found from the current, the battery may be fully charged and therefore will not accept as much or any current from the controller.

Figure 46: Solar Panel Testing

6.1.5 Charge Controller Testing

In addition to the solar panel itself, as mentioned above there will be a charge controller used in this project to limit and regulate the current coming from the solar panel. In addition to the operating current testing, testing the charge controller provides latency if there is too much current coming from the solar panel. To test the controller, the battery will be connected to the solar panel and the current from the panel to the battery will be tested with and without the charge controller. The resulting current should be more manageable but should not be zero. If the controller produces no current, the charge controller could be faulty or there could be a lose connection between the components. It is vital that the connections are made properly, the panel is tested in visible sunlight and that the battery connected to these components is not fully charged.

6.1.6 Battery Testing

In this project, a battery will be used to hold the energy gathered from that panel that is not being used immediately to power the electronics. As a result, the battery must also be tested for its ability to hold energy. There are many rudimentary tests that will sometimes be overlooked by both manufacturers or consumers that can result in battery or system failure. For example, before testing the battery, it must be visually inspected to make sure it is not damaged or leaking. Damaged batteries can harm the system. It is also important to check the voltage of the battery, verify the voltage is at maximum 0.5V-0.7V above the voltage advertised. While different batteries have different voltages and tolerances, it is important to confirm with the appropriate datasheet in order to know the allowable tolerance for the battery to perform optimally. The capacity of the battery is its ability to store energy. It is important to test this capacity in the battery. A new battery should be delivering 100% capacity, if the battery does not, it must either be refilled or replaced depending on the capacity. It is suggested that any battery under 80% capacity should be replaced. Testing the amperage can also determine if the battery is working properly. Since this measurement depends on internal resistance, battery temperature, and voltage, it is a good confirmation test that helps ensure that the other tests were conducted correctly and can provide confidence that the battery is working properly.

6.1.7 Microcontroller Testing

To properly test the control unit, a development board was used. This provided examples of uses for this MCU, requirements that this MCU needs to meet, as well as multiple ways of testing this controller with similar components already on the development board. Additionally, the development board came with tests and parameters with specific conditions to meet and test. For example, the development board's datasheet provided a specific schematic for testing the DC-to-DC converter on the development board. The board also included certain pins and schematics for testing. This microcontroller must pass both software and hardware tests.

The two test cases for software testing were random program generation (instructions and data are randomly generated) and custom program generation (instructions are generated from a test pattern from CPU testing). The random number generation test generates an op code as all data received would have. Internally, a look map table is provided to look up each opcode and the instruction's total length. It will also look up the data structure of the remaining operands. This type of testing is commonly used for software platforms.

The custom program generation test was set off of scripts from various OS platforms that used parts of previous instructions to create hex addresses. These were sent to the MCU simulator or physical MCU through an interface that controls how they are outputted. This type of testing can outperform the random program generation because faulty behavior is only exhibited from previously generated actions.

There is a different method used for testing both low pin-count microcontrollers and high pin-count microcontrollers. When testing lower pin-count MCUs, anywhere from 32 to 128 different pin sites may be tested in parallel. This required multiple power supplies with moderate power output. When testing the more complex and higher pin-count MCUs, four to eight different pin sites are tested in parallel. This test requires over ten different power supplies to allow different networks and unrelated elements to be tested individually.

Hardware was tested as well in the case of the microcontroller. In addition to the software testing previously mentioned, the hardware testing is equally as important. If the software is fully functional, the device will not work without the hardware being tested also.

The most common and useful tests for the hardware in the MCU are error tests along with mathematical tests. The error tests include the full-scale error test, the gain error test, and the offset error test. The graph in Figure 47 below shows where the errors should be present in reference to the voltage.

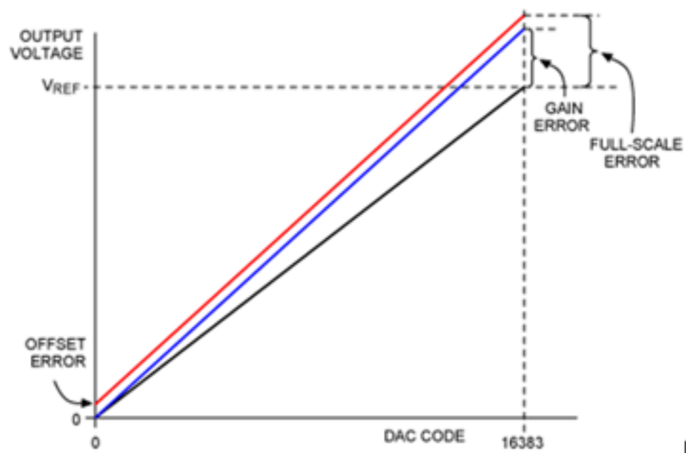


Figure 47: Error Testing Expected Results (Public Domain)

The full-scale error test takes the difference between the actual transition voltage minus the ideal transition voltage. This error is expressed using any measurement made within the limits of the MCU hardware requirements. As shown in the figure, the ideal the ideal full-scale error is before the rail voltage. At this point, on the evaluation board, the digital converter has reached it largest digital value before hitting the maximum input voltage.

The offset error is the first transition voltage minus the ideal first transition voltage. The offset error can have either a negative or positive value because it is usually defined as a straight line. As shown in the graph above, this error should be small when compared to the full-scale error.

Finally, the gain error is the difference between the full-scale error and the offset error. Because the full-scale error encompasses the entire system, it must be the result of the offset error and the gain error which measure both first and last transition voltages. The gain error represents the difference in slope of the actual transfer function (red and blue) and the ideal transfer function (black).

The mathematical or code density tests are performed using a histogram test. During this test, a dynamic sine wave input signal is applied to the analog to digital converter on the evaluation board. This will generate a distribution of digital outputs from the converter. If there is any digital output that does not correspond to those outputs, errors are generated. Even though these mathematical error tests are mostly used to test the converters that communicate with the MCU, without the proper input from this converter, the MCU will not function properly.

6.2 Software Testing

Testing the software involved in our project is an important step in the design process and must be done correctly. Software testing generally consists of what are called "unit tests." In these, a testing framework is designed to individually test each aspect of the code. Every class and every function is tested individually to show the correct behavior. Then, after each component of the code passes each individual test, the next step in testing is to test combined functionality using the same unit tests and then overall test functions. Using this testing method allows the designers to not only test lower-level functionality but also the integration of each piece. This isolates components and their possible bugs in the program before all of the components are fully integrated together. Another software testing method is to use test procedures that a user or designer would read and then with which they would manually test the product to make sure that the appropriate actions are taken depending on the given inputs.

In relation to these types of testing is the testing methodology and how it effects to the software development cycle. There are different software development strategies; the most common in modern software engineering is called "Agile." This development strategy, illustrated in Figure 48 below, follows alongside of iterative development and test-driven development and is done in short bursts called sprints. Using this methodology forces the development team to reevaluate and recheck code on a weekly or bi-weekly basis as a team. This gives flexibility to the coding schedule and allows the team to set priorities on the software design as necessary.

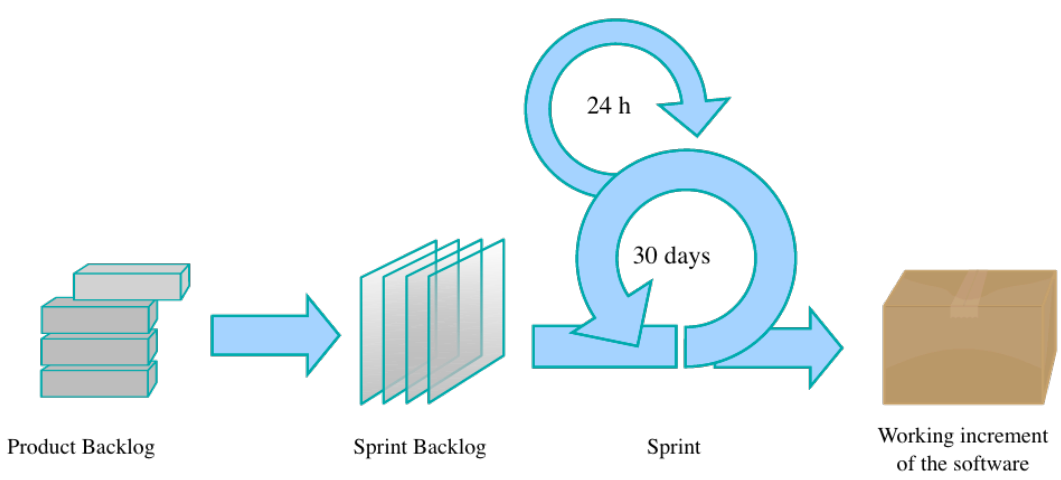


Figure 48: Agile Development Cycle (Public Domain)

Another software development strategy is known as "Waterfall," depicted in Figure 49. Testing in the waterfall methodology takes place only after the software design is complete and the code has been written. Then, in that methodology, the code is tested. If the program is not working, then the entire structure from requirements through design and implementation may need to be completely redone instead of just modifying what is needed iteratively.

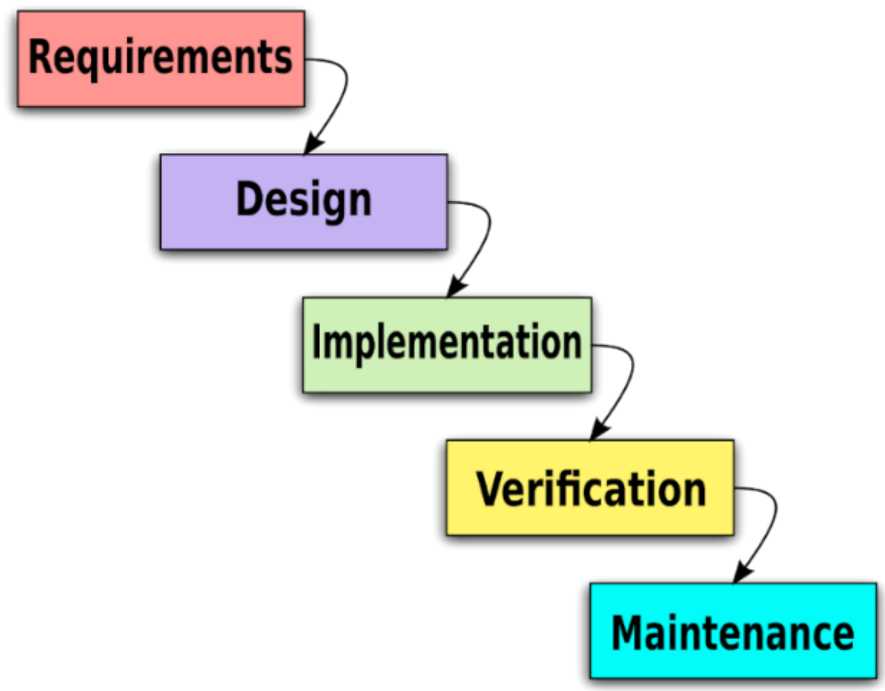


Figure 49: Waterfall Development Cycle (Public Domain)

In most real-life cases, both methods are done and at the end of the development cycle a full regression test is performed to ensure that the entire system, and every component, works as expected for every single situation presentable. In most cases, this is a full check making sure the system fails where it is designed to fail and pass when it is designed to pass. For a real-time system, typically tests include checking tolerances and making sure that systems return within the proper time frame to guarantee safety.

6.3 Product Operation and User Manual

As a guideline for our testing we want to make sure the device can eventually be operated as indicated by this operation guide and user manual. Since we are so focused on staying within department of transportation standards, those dictate the location of our device, and thus how it is monitored and changed. If the user setup guide is not adhered to it is possible that the device will not function as intended and could cause major accidents.

6.3.1 Product Setup

The Situation-Aware Stop Signal first needs to be configured for the roadway it is to be used on. Due to state government and national government regulations on traffic signals, this must be done for any new intersection. For demonstration purposes we will be using timings that are configured for 25 miles per hour. Then, each lighting submodule is to be attached to the main device as shown in Figure 50. Since each intersection can be in roadways with different speed limits, this device must be programmed (or manually configured) on an individual intersection basis to ensure that the device adheres to the MUTCD standards.

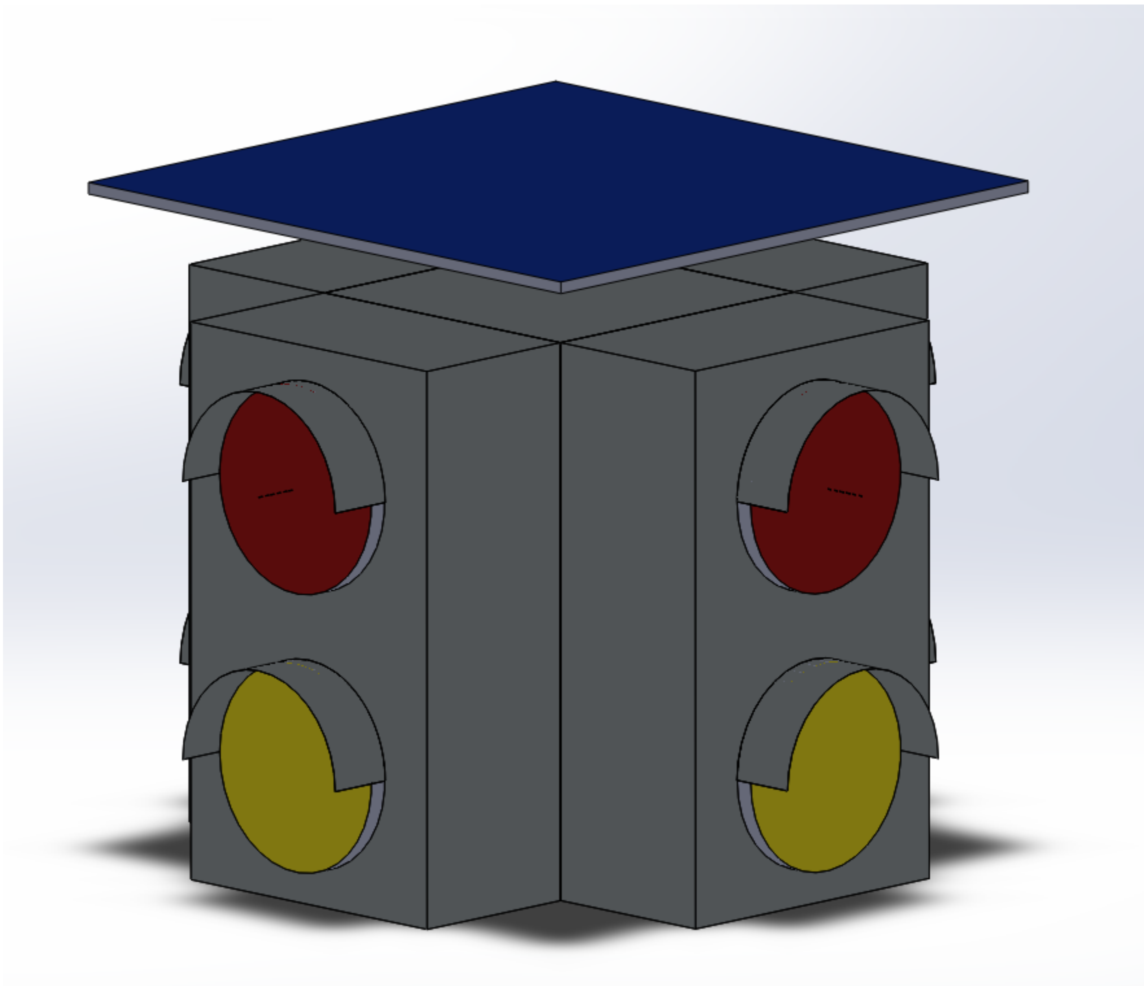


Figure 50: Detailed 3D Rendering

The device must be placed in the intersection with each sensor module facing the roadway that it will be monitoring, as illustrated in Figure 51. When placing the device in the intersection please refer to the MUTCD to ensure that the device adheres to the placement standard so that optimal safety standards are maintained. Failure to do this may result in incorrect or inconsistent operation which will put driver's life's in danger.

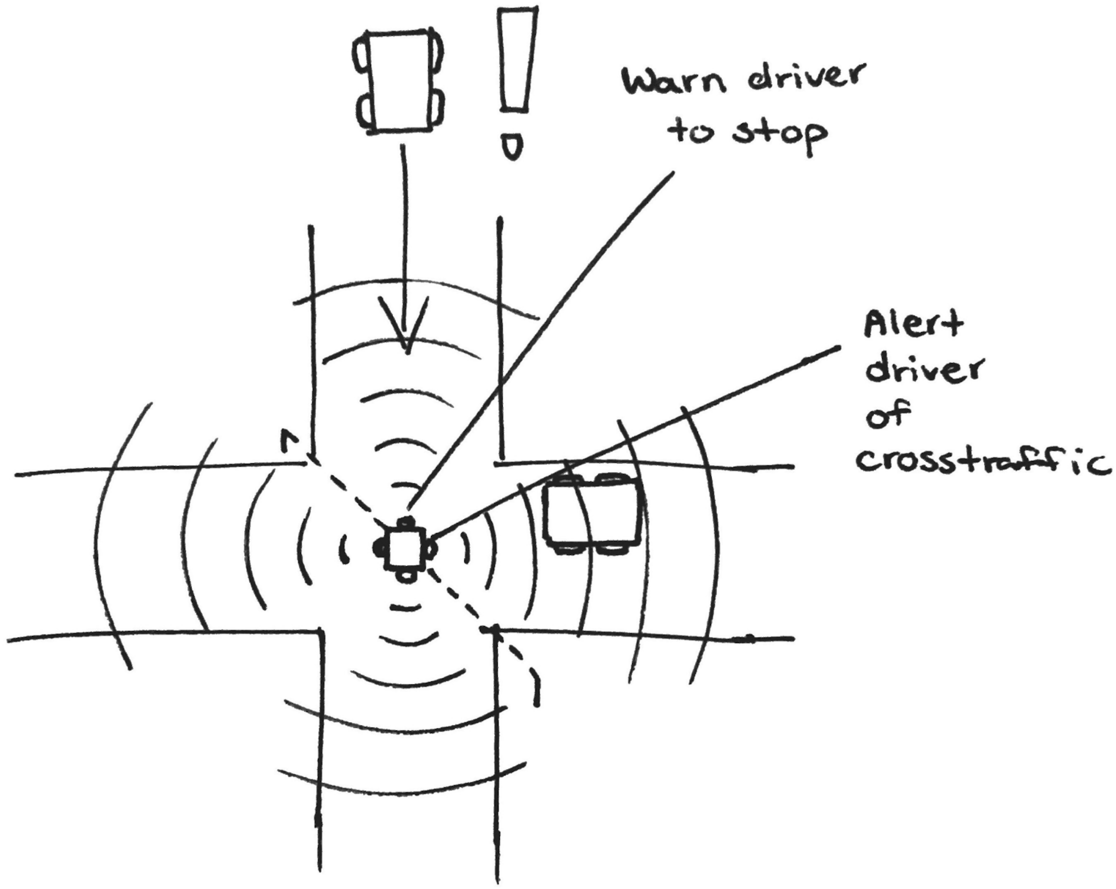


Figure 51: Placement in a 4-way intersection

The last two steps are to first set the solar panel to the optimal angle and secondly to angle the RADAR and LiDAR at their proper angles. The RADAR should be angled ~15 degrees down, and the LiDAR should be pointed at the stop bar. Once that has all been done, the device is properly setup and no additional user input should be needed.

7 Administration

Work pertaining to the Situation-Aware Stop Signal project has been planned with consideration given to schedule and budget. Schedule involved an analysis of the scope of the work required for the development of the stop signal and included the division of that work into manageable phases that collectively built upon one another until project completion. Budget sets a financial limit to the project and requires that all work and material not exceed that which can be afforded by the project. This chapter of the design document will provide further explanation of project schedule and budget given the constraints of the Senior Design curriculum.

7.1 Schedule

The schedule of work for the Situation-Aware Stop Signal project is adapted from the V-model of systems engineering. It suggests an extensive process of requirements analysis and design before implementation. Once implementation is complete, the project enters a prolonged period of integration and test. It is during and after this period that the system is verified and validated as to be performing its proper and desired function. An overview of all project phases, their durations, and associated milestones is given in the Gantt chart found in Figure 52 below on the next page.

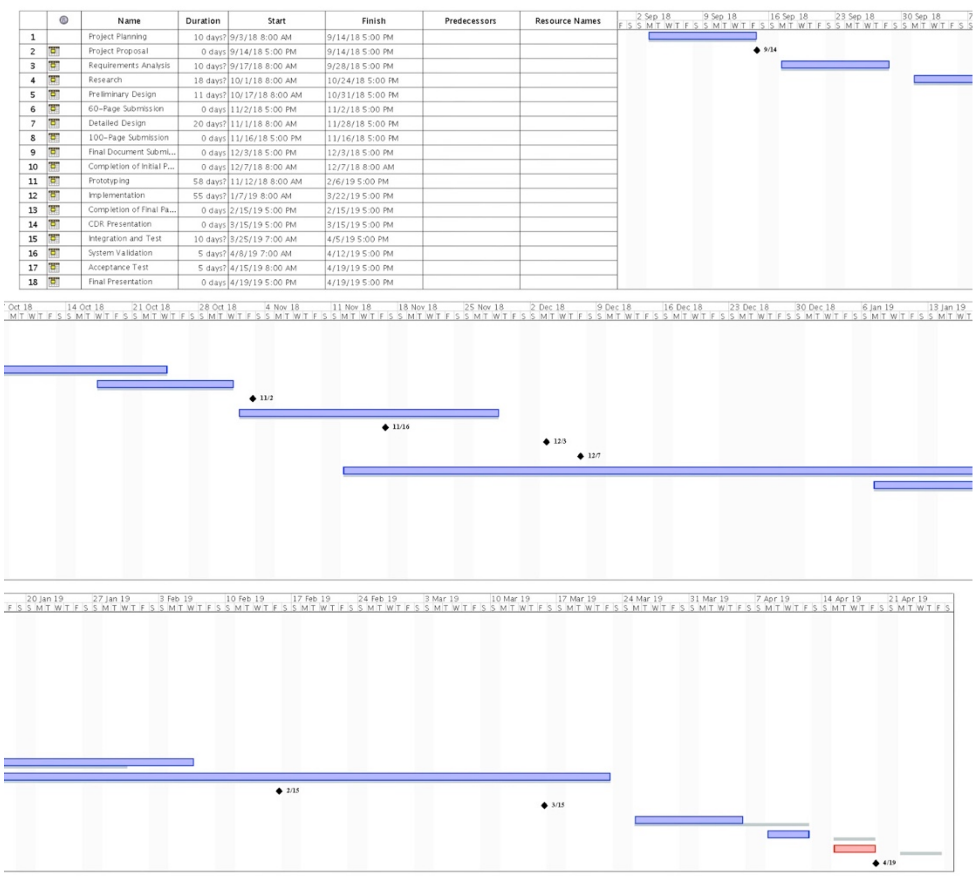


Figure 52: Gantt Chart

Project milestones are indicated as diamonds in the Gantt chart and are spread throughout the length of the project. The first milestones shown are related to project proposal and design, most particularly in the form of typed deliverables, or submissions of design documentation. Later milestones include the completion of part orders and presentations to be given during the second semester of Senior Design. All milestones are listed in Table 17 with their fixed and tentative dates.

Table 17: Project Milestones

Milestone	Date
Project Proposal	September 14, 2018
60-Page Submission	November 2, 2018
100-Page Submission	November 16, 2018
SD1 Final Document Submission	December 3, 2018
Completion of Initial Parts Order	December 7, 2018
CDR Presentation	February 1, 2019
Completion of Final Parts Order	February 24, 2019
Midterm Demo	March 31, 2019
Final Presentation	April 16, 2019
SD2 Final Documentation Due	April 22, 2019

7.2 Budget

The Situation-Aware Stop Signal was designed to be a low-cost product that could replace stop signs. Because the device was run from solar power, installation was intended to be quick and low cost. Stop signs are sold for less than $100, which is unparalleled for any traffic light device; however, for this project we aim to stay within a $1000 budget. This budget is representative of a two-sided stop signal, determined to be sufficient for demonstration purposes. While the project does not have any formal sponsors and is self-funded, mmWave sensors were provided by Texas Instruments. With this generosity considered, we plan to purchase all remaining components for around $900. Market costs for all major components in a two-sided stop signal system are shown in the Bill of Materials given in Table 18 below. While we did slightly go over budget, this was just to buy replacement components such as regulators and connectors. The projected bill of materials remained relatively the same.

Table 18: Bill of Materials

Product	Subsystem	Quantity	Unit Cost	Total Cost
Solar Cell	Power	28	$4.13	$115.64
Lithium Ion Battery	Power	1	$87	$87.00
MPPT Solar Charge Controller	Power	1	$69.45	$69.45

LED Traffic Light	Hardware	4	$42.75	$171.00
TI mmWave Evaluation Board	Hardware	2	$300	$600.00
Garmin LIDAR-Lite V3HP	Hardware	2	$149.99	$299.98
MCU	Hardware	1	$10	$10.00
PCB	Hardware	1	$10	$10.00
Physical Building Material	Mechanical	1	$100	$100.00
Misc.	Misc.	1	$50	$50.00
Total				**$1513.07**

7.3 Requirements Traceability

The requirements determined at the conception of the Situation-Aware Stop Signal project have been considered throughout the design process and may be traced to sections in the design document as indicated in Table 19.

Table 19: Requirements Traceability

Requirement	Design Document Section(s)
The signal shall be one centralized unit.	5.4.1
The device shall be visible during the day.	5.2.4
The device shall be visible at night time.	5.2.4
The signal shall be IP52 weather resistant.	5.4.2
The signal shall maintain operability between 0°C and 60°C.	5.2
The signal shall detect an oncoming vehicle within 20 meters.	5.2.3.1
The signal shall be operable 24/7.	5.1.2.2
The device shall signal drivers to stop safely before the intersection.	2.2.1
The device shall signal all drivers to stop when a threat is detected approaching the intersection.	2.2.1
The device shall prevent conflicts when vehicles arrive at the same time at the intersection.	2.2.1
The device shall normally run autonomously.	6.3
The device shall have a manual override mode.	**Error! Reference source not found.**
Solar charge controller shall output 12V.	5.1.3
Solar panel shall output greater 6V.	5.1.1
Battery shall hold enough power for a minimum of two days of operation.	5.1.2
Solar panel shall output greater than 40W.	5.1.1

Solar panel shall charge the battery while device is in operation.	5.1.1
The signal shall abide by road sign laws specified in the Manual for Uniform Traffic Control Devices (MUTCD).	3.1

8 Conclusion

The goal of this project was to develop a smarter and more efficient replacement for the current stop sign. To do this, many design decisions and concerns had to be resolved. This device had to abide by all traffic standards and laws put in place by the Department of Transportation. The last, but most important goal to achieve with this project is that the device must run completely off solar power. The success of this design lies with the success of each individual subsystem and designing each aspect with maximum power efficiency.

The main requirement of the Situation-Aware Stop Signal is to keep drivers safe and prevent car accidents. With this in mind, the design challenge became how to orient this device so that it does not distract or confuse drivers. The solution to this challenge was to design the device so it resembled a traffic light. The difference being that our device does not have a green light. This decision was made as a result of the decision not to fully control traffic with this device, but to monitor it. Rather than giving the drivers the "go" signal, this device tells the drivers to "proceed with caution." In addition to the redundancy created by using both a LiDAR and RADAR sensor to collect information on the car's location and speed, this minimizes the risk of drivers blindly following the device. Also, making this device modular allowed for both additional power saving as well as the ability of this device to be placed in both 4-way and 2-way intersections.

In addition to safety, in order for the Situational Aware Stop Signal to be implemented, it must follow the traffic standards in place. These standards created limits and constraints for the way the device could be designed. Originally, traffic warning lights were placed on each side of the device that would indicate which direction traffic was approaching from. Because of the light orientation standards found in the MUTCD, this design was reconsidered after finding that this orientation was not to the standard. This design decision was just one of many that was chosen based upon the current traffic standards in place.

Speed was another major requirement for this device. Delay in the changing of the LEDs could create more safety concerns for both drivers and pedestrians. The way this issue was resolved was by optimizing the software as well as the hardware. Originally, the sensors were designed to be cascaded from the designed PCB board. In this case, the PCB would communicate with the RADAR board, which communicate with the LiDAR board, all using the same line and form of communication. While it is easier to use the same form of communication, this design was not efficient and would create large delays. To solve this issue, a more paralleled design was developed where each sensor used a different method of communication and each sensor was directly connected to the PCB board. This design used different communication types which allows the MCU to perform multiple processes at the same time instead of delaying the less important processes.

Resolving these issues has allowed the Situational-Aware Stop Signal to evolve into an efficient device that can detect where cars will be and if these cars will stop or not. Additionally, this device is self-sufficient and able to easily be integrated into society because of its stop-light-like appearance. We hope this project improves traffic on low volume roads while reducing both traffic accidents and waiting time for cars.

Appendix A - References

"Advanced Circuits." *Printed Circuit Board Manufacturer & PCB Assembly | Advanced Circuits*, www.4pcb.com/pcb-electrical-testing.html.

Araujo, Keith. "Battery Cell Comparison." *Epec Engineered Technologies - Build to Print Electronics*, www.epectec.com/batteries/cell-comparison.html.

"Battery Anodes." *The Energy Materials Center at Cornell Site*, www.emc2.cornell.edu/content/view/battery-anodes.html.

Diaz, Juan Carlos Fernandez. "Lidar Remote Sensing." *Springer Link*, 2013, link-springer-com.ezproxy.net.ucf.edu/referenceworkentry/10.1007/978-1-4419-7671-0_44.

EETimes, and Embedded. "2017 Embedded Markets Study Integrating IoT and Advanced Technology Designs, Application Development & Processing Environments." *M.EET*, ASPENCORE, Apr. 2017, m.eet.com/media/1246048/2017-embedded-market-study.pdf.

"Electromagnetic Fields and Public Health: Radars and Human Health." *World Health Organization*, World Health Organization, 4 Aug. 2016, www.who.int/peh-emf/publications/facts/fs226/en/.

Larmour, Jonathan. "Https://Www.ecoscentric.com/Docs/ew06-Paper-Larmour-Ecos.pdf." *ECosCentric*, Cambridge, www.ecoscentric.com/docs/ew06-paper-larmour-ecos.pdf.

Federal Highway Administration. "Manual on Uniform Traffic Control Devices." *FHWA*, Department of Transportation, 2009, mutcd.fhwa.dot.gov/pdfs/2009/mutcd2009edition.pdf.

"Florida Traffic School | Defensive Driving Courses." *Parking Laws in Florida | Drivers Handbook*, www.123driving.com/dmv/drivers-handbook-road-signs.

Garcia, Keegan, et al. *Robust Traffic and Intersection Monitoring Using Millimeter Wave Sensors*. Texas Instruments, May 2018, www.ti.com/lit/wp/spyy002b/spyy002b.pdf.

Garmin. *LIDAR Lite V3HP Operation Manual and Technical Specifications*. Sparkfun, cdn.sparkfun.com/assets/9/a/6/a/d/LIDAR_Lite_v3HP_Operation_Manual_and_Technical_Specifications.pdf.

Goebel, John. "Hardware Testing and System Qualification: Procedures to Evaluate Commodity Hardware and Production Cluster." *SLAC Stanford*, Stanford, www.slac.stanford.edu/cgi-wrap/getdoc/slac-pub-9761.pdf.

Google. “Google Style Guide.” *GitHub*, github.com/google/styleguide/blob/gh-pages/pyguide.md.

“Google C Style Guide.” *Google Java Style Guide*, google.github.io/styleguide/cppguide.html.

Gotzig, Heinrich, and Georg Geduld. “Automotive LIDAR.” *Handbook of Driver Assistance Systems*, Aug. 2015, pp. 405–430., doi:10.1007/978-3-319-12352-3_18.

“History.” *Counting Semaphores for Resource Management and Event Counting in FreeRTOS Real Time Embedded Software Applications*, www.freertos.org/RTOS.html.

“How Batteries Work.” *HowStuffWorks*, HowStuffWorks, 1 Apr. 2000, electronics.howstuffworks.com/everyday-tech/battery5.htm.

“How To Check Your Solar Panel & Regulator/Controller.” *Select Solar*, www.selectsolar.co.uk/cat/171/testing-your-solar-panel-regulator.

INTERNATIONAL DARK-SKY ASSOCIATION. *THE PROMISE AND CHALLENGE OF LED LIGHTING: A PRACTICAL GUIDE*. Dark Sky, www.darksky.org/wp-content/uploads/bsk-pdf-manager/IDA_LED_handout_48.pdf.

“Introducing Zephyr.” *CC2650 SensorTag - Zephyr Project Documentation*, docs.zephyrproject.org/latest/introduction/introducing_zephyr.html.

Iovescu, Cesar, and Sandeep Rao. “The Fundamentals of Millimeter Wave Sensors.” *Texas Intsruments*, www.ti.com/lit/wp/spyy005/spyy005.pdf.

“Its All about the Sensors.” *Second Robotics*, Second Robotics, 10 Apr. 2016, secondrobotics.com/its-all-about-the-sensors/.

JJS Manufacturing. “Design Guidelines for in-Circuit Testability.” *cdn2*, JJS Manufacturing, cdn2.hubspot.net/hubfs/353296/ebook_pdfs/Design_guidelines_for_in-circuit_testability.pdf.

Jo, Jun. “A Likelihood-Based Data Fusion Model for the Integration of Multiple Sensor Data: A Case Study with Vision and Lidar Sensors.” *Springer Link*, July 2016, link-springer-com.ezproxy.net.ucf.edu/chapter/10.1007/978-3-319-31293-4_39.

Knight, David. “Practical PCB Layout Tips Every Designer Needs to Know.” *All About Circuits*, 13 Nov. 2015, www.allaboutcircuits.com/technical-articles/practical-pcb-layout-tips/.

Marsh, Jacob. “Why Are Solar Panels Blue vs Black? | EnergySage.” *Solar News*, EnergySage, 29 Aug. 2018, news.energysage.com/why-are-solar-panels-blue/.

"MMWave Sensor Overview." *LM741 Operational Amplifier | TI.com*, www.ti.com/sensors/mmwave/overview.html.

Miller, Jessica. "NEWS." *Virtualization and the Internet of Things*, WIND, www.windriver.com/news/press/pr.html?ID=12641.

"MSP430™ Ultra-Low-Power Sensing & Measurement MCUs." *LM741 Operational Amplifier | TI.com*, www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html.

"Noisecom - Characterizing RADAR Interference Immunity." *A Leading Manufacturer of RF and Microwave Noise Generation*, www.noisecom.com/resource-library/articles/characterizing-radar.

"PCB Design Tips | Sierra Circuits Blog." *Sierra Circuits*, www.protoexpress.com/blog/pcb-design-tips/.

"PCB Electrical Testing." *MCL*, 2 Jan. 2018, www.mclpcb.com/pcb-capabilities/pcb-electrical-testing/.

"PCB Test - Printed Circuit Boards." *Bay Area Circuits*, 24 July 2017, bayareacircuits.com/bare-printed-circuit-board-electrical-test/.

"Peak Sun Hours for Solar Panels in Florida." *TurbineGenerator*, www.turbinegenerator.org/solar/florida/.

"Photovoltaic Cell." *Internal Combustion Engine - Energy Education*, energyeducation.ca/encyclopedia/Photovoltaic_cell.

Ramasubramanian, Karthik, et al. *Moving from Legacy 24 GHz to State-of-the-Art 77 GHz Radar*. Texas Instruments, Oct. 2017, www.ti.com/lit/wp/spry312/spry312.pdf.

"RF Wireless World." *Modulation Index in AM and FM | What Is Modulation Index*, www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-millimeter-wave-radar.html.

Robledo, Edwin. "Ten Best Practices of PCB Design." *EDN*, www.edn.com/electronics-blogs/all-aboard-/4429390/Ten-best-practices-of-PCB-design.

Spiess, Andreas. "#203 Best LIDAR Sensors for Makers (Comparison and Test)." *YouTube*, YouTube, 9 June 2018, youtu.be/ddxguAzzzJE.

Staff, Stouch Lighting. "Everything You Need To Know About Outdoor LED Lighting." *LED Lighting Distributor and Consulting Company*, www.stouchlighting.com/blog/exterior-and-outdoor-led-lighting.

"Standard for Sensor Performance Parameter Definitions." *An Introduction to Biometric Recognition - IEEE Journals & Magazine*, Wiley-IEEE Press, ieeexplore.ieee.org/document/6880296.

"Statutes." *Statutes & Constitution :View Statutes : Online Sunshine*, 28 Nov. 2018, www.leg.state.fl.us/Statutes/index.cfm?App_mode=Display_Statute&URL=0300-0399/0316/Sections/0316.075.html.

"STOP Short Yellow Lights." *Yellow Light Time Standards | Stop Short Yellow Lights*, www.shortyellowlights.com/standards/.

Texas Instruments. *MSP432P401R, MSP432P401M SimpleLink™ Mixed-Signal Microcontrollers*. Texas Instruments, Sept. 2017, www.ti.com/lit/ds/symlink/msp432p401r.pdf.

Texas Instruments. "IWR1642 Single-Chip 76- to 81-GHz MmWave Sensor." *Datasheet*, Texas Instruments, Apr. 2018, www.ti.com/lit/ds/symlink/iwr1642.pdf.

Texas Instruments. *MSP432P401R SimpleLink™ Microcontroller LaunchPad™ Development Kit (MSP-EXP432P401R)*. Texas Instruments, Mar. 2018, www.ti.com/lit/ug/slau597f/slau597f.pdf.

"The International Dark-Sky Association (IDA)." *International Dark-Sky Association*, 24 Sept. 2018, www.darksky.org/our-work/lighting/lighting-for-industry/fsa/fsa-products/.

TI-RTOS Kernel (SYS/BIOS) User's Guide. Texas Instruments, Feb. 2018, www.ti.com/lit/ug/spruex3u/spruex3u.pdf.

"Types of Batteries." *PRBA Types of Batteries Comments*, www.prba.org/battery-safety-market-info/types-of-batteries/.

"U.S. Energy Information Administration - EIA - Independent Statistics and Analysis." *Factors Affecting Gasoline Prices - Energy Explained, Your Guide To Understanding Energy - Energy Information Administration*, www.eia.gov/totalenergy/data/annual/showtext.php?t=ptb0802a.

"Whats the Difference between Nickel Cadmium (Nicad), Nickel-Metal Hydride (NiMH), and Lithium Ion (Li-ion)?" *Battery Universe Online Battery and Accessories Sales since 1999*, www.batteryuniverse.com/help/battery-chemistries.

Windy Nation. "Testing Your Solar System." *Windy Nation Inc.*, Windy Nation, www.windynation.com/cm/Testing a Solar System_R1.pdf.

Yurtoğlu, Nadir. "History Studies International Journal of History." *History Studies International Journal of History*, vol. 10, no. 7, 2018, pp. 241–264., doi:10.9737/hist.2018.658.

Zhang, Chunjia, et al. “Real-Time Road Detection Using Lidar Data.” *Advances in Intelligent Systems and Computing Knowledge Engineering and Management*, 2013, pp. 575–585., doi:10.1007/978-3-642-37832-4_52.