Visual Light Communication

By

Matt Martinico, Benjamin Hadra, Sean Foley,
Joe Mulhern, Jensen Rosemond, Bhumika Sood

Table Of Contents

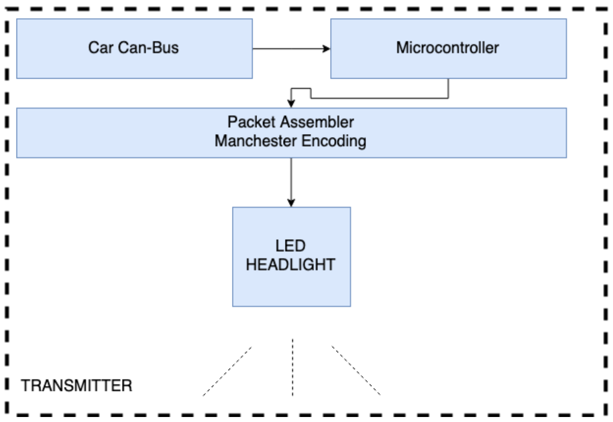
Overview of VLC

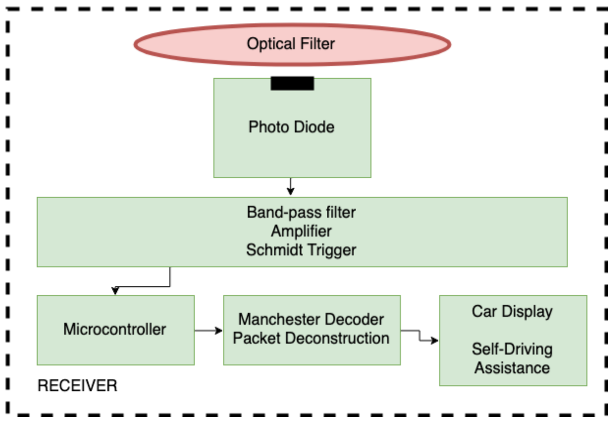
Visible Light Communication (VLC) is a form of wireless communication involving data transmission in the visible light spectrum. Light emitting diodes (LEDs) are generally used as transmitters in VLC through modulation in which data is encoded. VLC has the potential to supersede the limitations of radio frequency (RF) communication, which is the current norm, by allowing for greater capacity for transmission, scalability, and better security. [1]

VLC relies on the visible spectrum to transmit data for which the bandwidth is largely unused. The potential for high data-rate communication through VLC is vast especially compared to traditional radio frequency communication where there is a decline in spectrum bandwidth. The ubiquity of LEDs serves as an advantage to VLC scalability. In VLC, data is transmitted through modulating LED light source at a rate that is imperceptible to the human eye. Through methods of amplification, data transmission at high rates can be achieved in VLC with no impact on the primary function of LEDs which is illumination. [2] VLC requires the receiver to be in the range of the transmitting LED to communicate data. This presents security benefits to data communication as VLC signals would not interfere with one another compared to similar concerns in RF-communication.

Autonomous vehicles have limitations for car-to-car communication for which VLC can be a solution. With little additional technology in vehicles, cars would be able to communicate information with one another through existing architecture via head and tail lights at high data rates passing information that could allow for greater safety on the road.

System Block Diagram





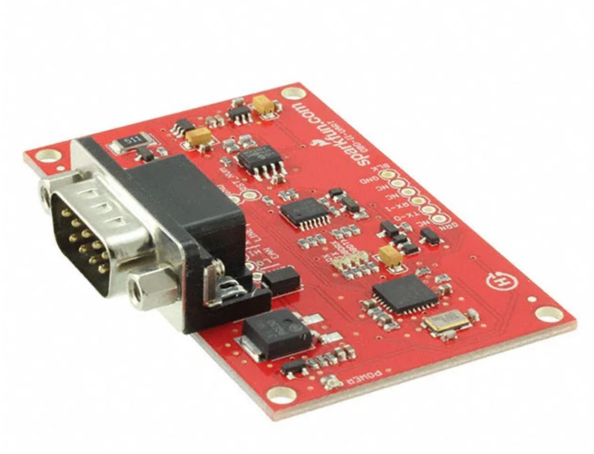
Transmitter

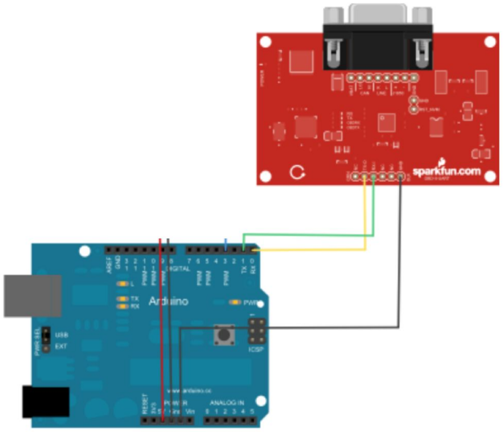
Bit Rate

In order to transmit information using an LED headlight, we needed to modulate the LED at a high rate to prevent the flicker from being visible to the human eye. In addition, we needed a high enough bit rate to send a significant amount of information. After trials with different LED headlights, we found that a bitrate of 2400 Hz gave optimal results. We were able to send all of our data packets with stellar signal integrity at the detector.

OBD II

One of the key challenges of our project was getting data from the car's computer encoded it into the modulation of the headlight. We were able to accomplish this by taking advantage of the on board diagnostic (OBD) port, which is standard on most modern vehicles. At this port one is able to access a plethora of information about the current state of the car, such as speed, RPM, acceleration, etc. Unfortunately there are a variety of different standards of OBD II used. This meant that the serial protocol used by the OBD port was dependent on the maker of the car, making is necessary for us to use the below interpreter board, called the WIG-09555.





This board is able to interpret any of the common OBD II standards, making is easy for us to communicate with any make or model that uses OBD II. It gives us access to a serial interface which uses the ELM327 command set making it easy for us to decode all necessary information. Therefore, by connecting the Rx and Tx pins on the OBD II board to the Rx and Tx pins on our arduino as shown above, we were able to access the car's speed and RPM using our arduino. We could then encode this information into a PWM signal using our arduno in order to modulate the headlight with ODB II data encoded in it.
https://learn.sparkfun.com/tutorials/obd-ii-uart-hookup-guide/connecting-to-an-arduino

LED Headlight Modulation

A typical LED headlight runs on a 12V DC supply from its car battery. The LED headlight ran at about 30W, producing roughly 5000 lumens of light. This is a fairly significant amount of power, so we used components that are safe at high currents. Due to the relatively high power of the LED we needed a method of modulating these high currents. In order to accomplish this, we used a power MOSFET, capable of handling up to 12A, giving us the ability to modulate the headlight at full brightness. This was possible with the simple circuit shown below.

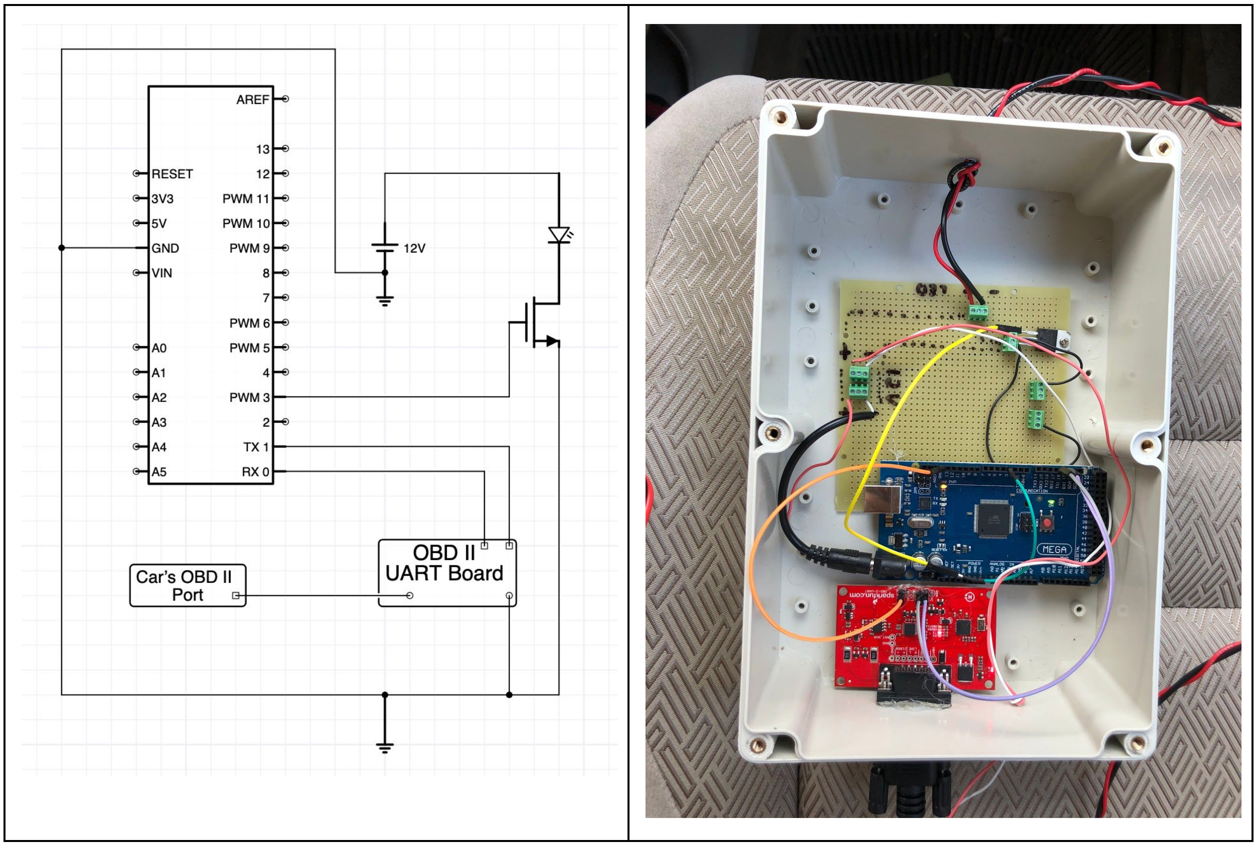


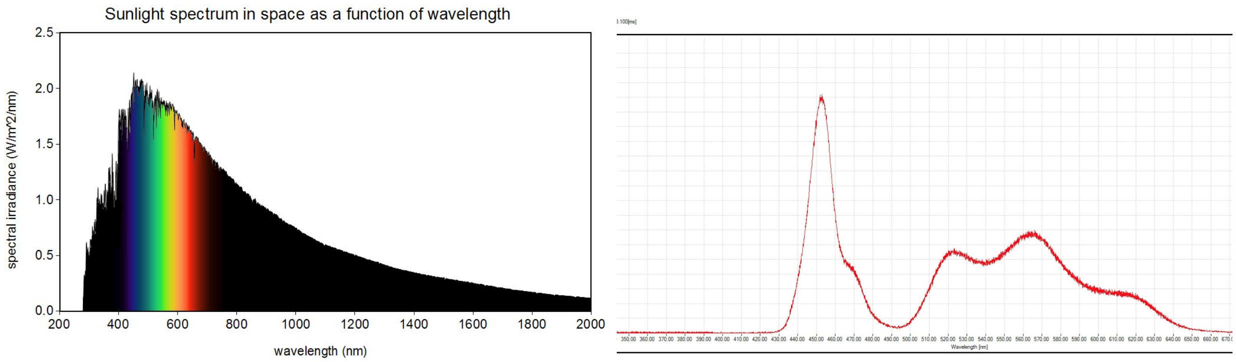
Figure 1: Transmitter circuit and implementation

By connecting the gate of the MOSFET to one of the PWM pins on arduino we were successfully able to modulate the headlight according to the PWM signal. This would therefore send OBD II data from the sending car through the headlights to the receiving car.

Receiver

Optical Filtering

A potential problem associated with VLC is the interference of other lights going into any given receiver. For example, at any time on the road, there are a multitude of light sources, and if the receiver is only intended to receive light from the headlights and taillights of another vehicle, there needs to be some sort of optical filtering. Interference sources could include sunlight, street lights, bike lights, camera flashes, reflections coming from any surface, moving visual billboards, etc. Based on initial testing and results, we initially used a physical filter isolating the 450nm spectrum. We determined this spectrum based on spectrometer readings from the lab. With white LEDs such as ours, often it is really a blue LED shining with a yellow phosphor coating. Therefore if we are sending information with a white LED, we can isolate a band around 450 nm because the bulk of the light energy is coming from blue colors. Representing white light as bands can be seen in the figures below. It is also important to look at the spectrum for sunlight, which will likely be the main source of interference. Unfortunately, adding this filter removed too much irradiance for us to get a clean signal, so it was removed in the final revision.



Figures 2 & 3: Spectrum of sunlight and Spectrum of our LED Headlight [3]

Light Detection - Photodiode

There are a variety of possible light detection devices available for the receiver. The main four which could be convenient for this application are a photoresistor, phototransistor, photodiode, or a video camera. A photoresistor typically has a rise and fall time in the range of 2-50ms, much too slow to work with the transmitter’s switching speed. A standard video camera runs from about 30-60 frames per second, which is again much too slow. A photodiode however, typically has a rise and fall time in the nano-second range. This makes the photodiode the perfect device to detect the light coming from the car’s headlights. When the device is connected in reverse bias, it blocks current from flowing just like a normal diode. However when the device is exposed to light in reverse bias, it allows current to flow proportional to the intensity of the light. Even with no light, however, there is some “Dark current” that will flow through the reverse-biased device. Figure four shows some example curves. The device we settled on is the OPT 101

Trans-impedance Amplifier (TIA). This photodiode comes in a DIP package and allows us to change feedback externally.

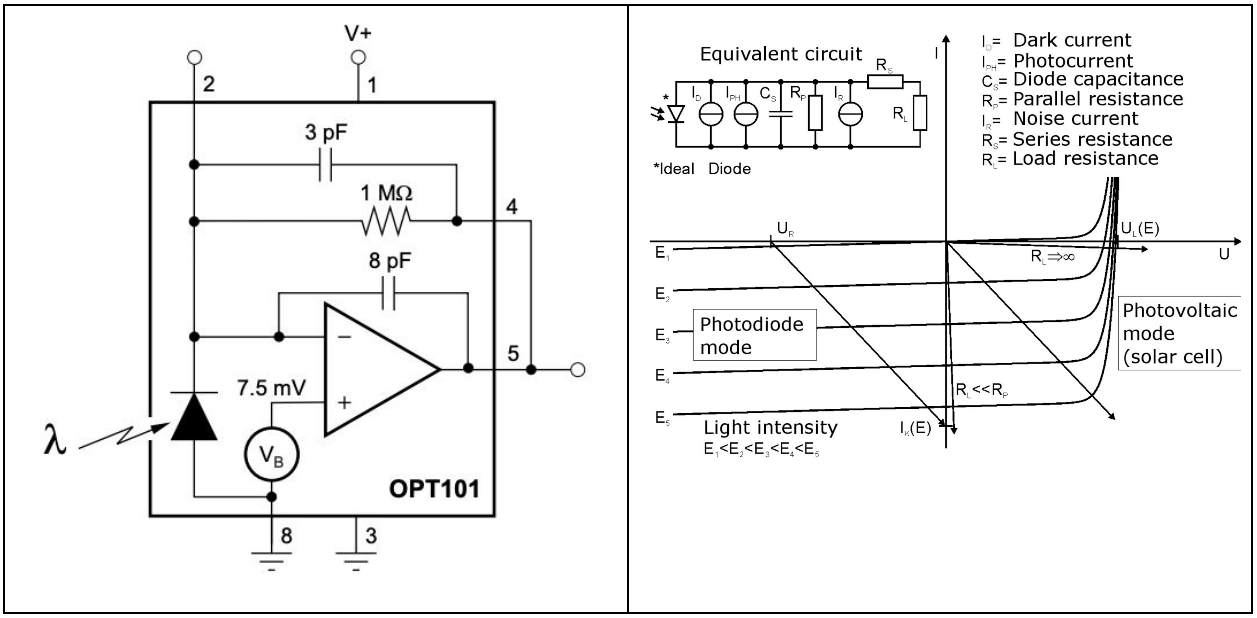


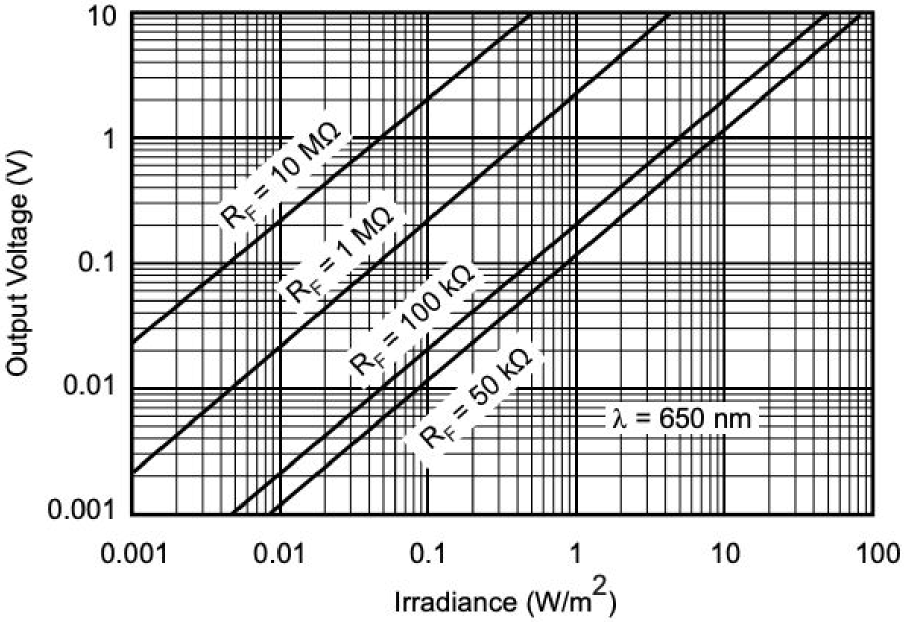
Figure 4 - Circuit diagram for our OPT 101 circuit, with a sample I-V curve of typical photodiodes [4].

From Figure four we see that as the luminance increases, the current through the diode increases proportionally. We also observe the two different load lines indicated. Using a simple KVL loop, we can derive the relationship Vout = $I_F R_F + V_D$, where F denotes the feedback resistor and D denotes the diode. With a large feedback resistor, the load line is very shallow, and the diode risks going into photovoltaic mode, which would cause issues with the fidelity of the filtered signal.

Amplification and Filtering

The relationship between Optical power and current is given by
$i = P_{Optical} * e^{-} * \eta / (h * v)$
Where eta is the optical efficiency of converting photons into electrons, in Amps per Watt. For simplicity it is assumed that our efficiency makes it so we get roughly one mA of reverse current through the photodiode with one Watt of optical power.

As previously explained, the current produced by the photodiode is relatively small, making an amplification step necessary. Fortunately the OPT-101 came with a transimpedance amplifier built in allowing us to either use the built in 1MΩ of feedback, or connect our own feedback, as shown in figure 4. Unfortunately, whether or not to use this built in feedback was a surprisingly difficult question to answer, as larger feedback values would allow the photodiode to produce a current with less light, but also cause it to saturate in lower light levels. The graph shown below shows the relationship between irradiance and output voltage with different values for feedback.



As can be seen from the graph there is a tradeoff with using a lower value for feedback as the receiver would be less sensitive to external light, but would also decrease the range of our system. We ultimately decided to use 50kΩ of feedback as it would make our system reasonably effective even in daylight, which was not possible with the full 1MΩ. However, due to the less amplification we received using lower the feedback, we had to include a second amplification step after filtering out DC. Below shows a high pass filter to remove all direct current, and our second amplification step separated by a buffer.

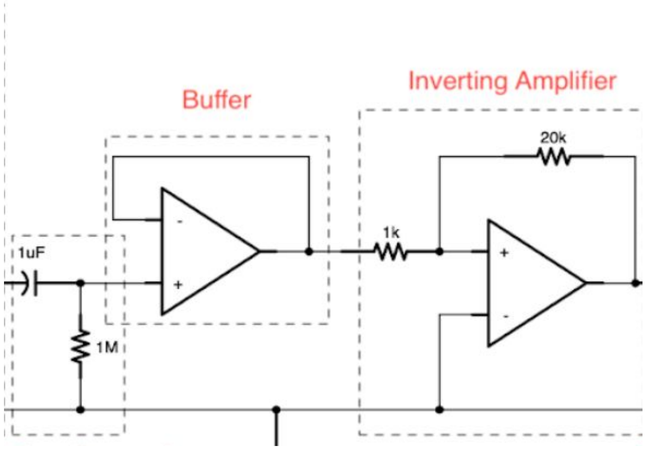


Figure 5: Amplification of signal from the TIA. A buffer is added before it so no loading occurs [5].

This post amp improved our range, while also allowing the system to work with some external light incident on the detector.

Signal Acquisition

The main drawback of using visual light to communicate information is that there will be a significant amount of noise received from other light sources, such as the sunlight. For this reason we needed a way to distinguish noise from valuable signals. To accomplish this we used a Schmitt trigger with a hysteresis of 0.75 mV. This threshold was determined to be the best tradeoff between blocking noise and passing signal. This comparator action converts our data from noisy analog data to crisp digital data for the Arduino.

Next, we needed to make sure that the Arduino we are using to decode our data is given Voltages in the range from around 0V-5V. To do this, A diode was added at the end of our circuit to bleed away negative voltages from the comparator.

hysteresis gives us noise immunity. The derivation for the 0.75 mV hysteresis is given by the following equations.

$$H=V_{HT}-V_{LT}$$

$$\therefore H= R_1/R_f\,(+V_{sat})-[-R_1/R_f\,(+V_{sat})]$$

$$\therefore H=2[(R_1/R_f)(V_{sat})]$$

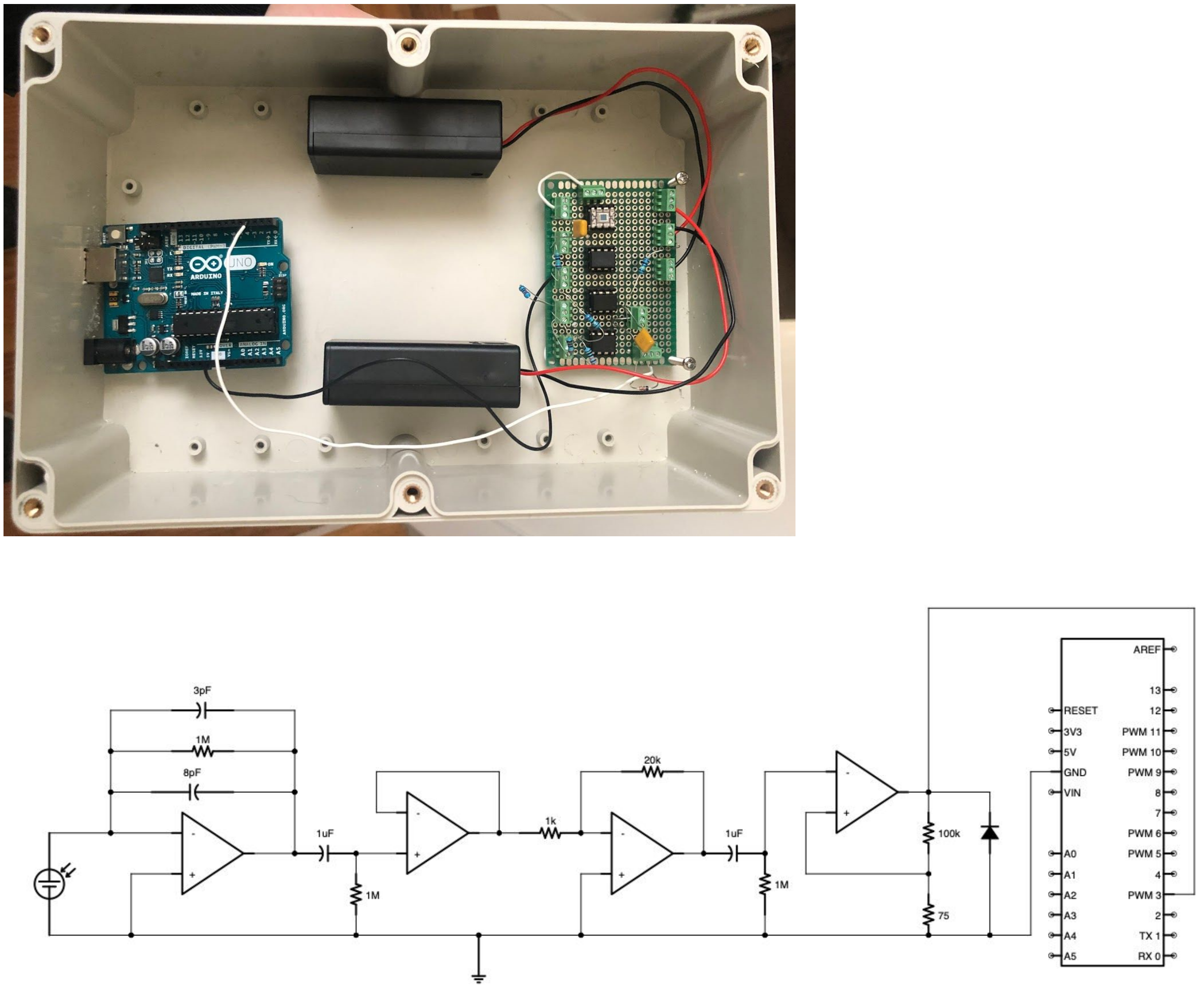
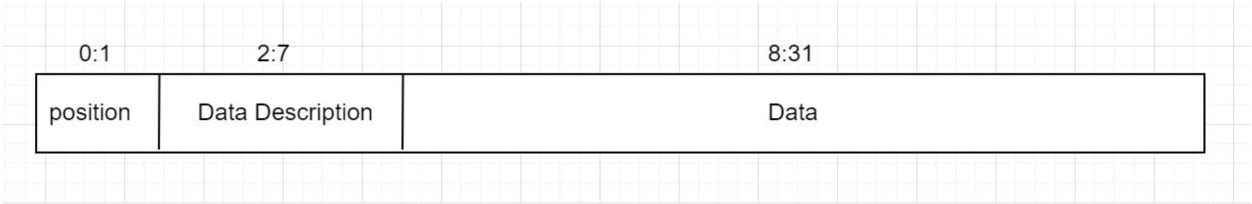


Figure 6: Schmitt trigger and diode blocks shown in the full receiver schematic, along with its physical implementation [5].

Communication

The basis of the project is the ability to send data between cars using headlights and taillights. This requires certain communication protocols and hardware for raw data to be distinguished on either end. The break down of how the communication will be transmitted is as follows: Car requests transmission, data is organized into a transmission of packets with a custom packet structure, the data is then encoded using Manchester Encoding, finally the data is sent, through the modulation of the lights. The receiving of the data is very similar, the manchester encoded message is decoded, and the important data is parsed from the packets based on the same structure. All of this is handled by two seperate Arduinos and then relayed to a main computer.

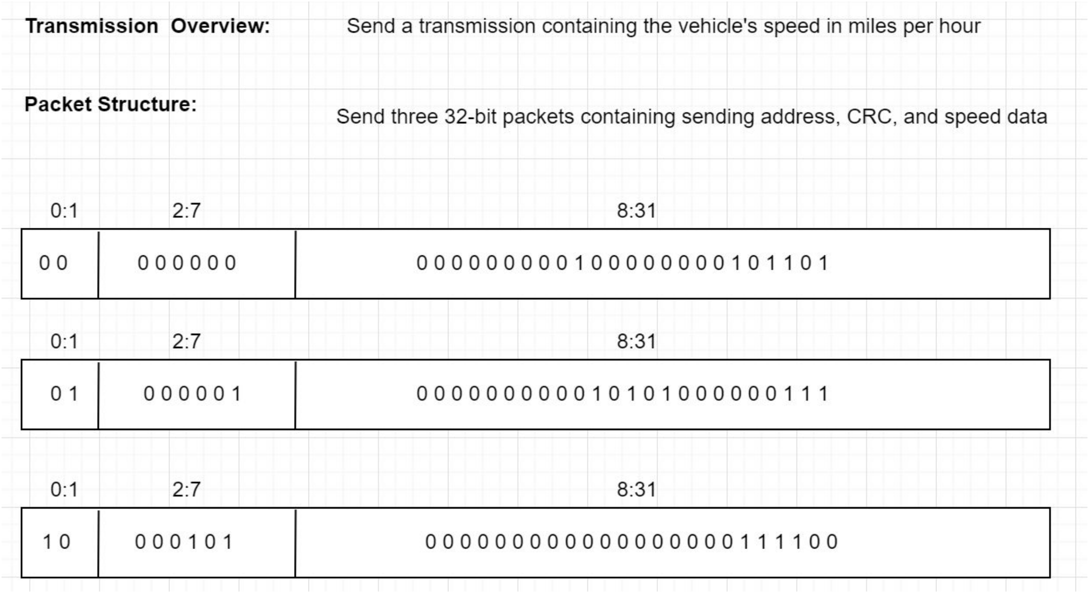
Transmission and Packet Structure



The structure of the packet describes what each bit represents and how it should be used. Any given transmission should have at the very least three packets containing address, CRC value, and message (speed, RPM…). The packet structure can be seen above. The first two bits represent the position of the particular packet within a given transmission of n packets. For example 00 is the first packet, 01 is for all packets 2 to n-1, and finally 10 is the last packet. The next 6 bits contain the data description. These bits explain what the data itself represents. The table below provides examples of the types of data that can be sent and what their corresponding description bits value is. Finally the last 24 bits contain the packet data.

Decimal Value	**Binary Value**	**Description**
0	000000	The sending car’s address
1	000001	The CRC value for the transmission
2	000010	The relayed vehicle’s address
3	000011	The vehicle’s braking value
4	000100	The vehicle’s directional signaling
5	000101	The vehicle’s speed in MPH
6	000110	The vehicle’s speed in KPH
7	000111	The vehicles distance to car in front
8	001000	The vehicles distance to rear car
9..63	~	RESERVED

When building a transmission three packets must be included, the address of the vehicle sending the packet, the CRC value for error checking, and a message. Below you can see an example of a vehicle sending its speed in miles per hour. When sending a transmission there is no limit to how many different messages one can send as long as the address and CRC value comes before all of them.



Transmission Overview: Send a transmission containing the vehicle's speed in miles per hour

Packet Structure: Send three 32-bit packets containing sending address, CRC, and speed data

0:1	2:7	8:31
0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1

0:1	2:7	8:31
0 1	0 0 0 0 0 1	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1

0:1	2:7	8:31
1 0	0 0 0 1 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0

CRC - Cyclic Redundancy Check

Cyclic Redundancy Checking is a form of error detection. Using a specified polynomial and initial value data can be assigned a check value. This check value is based on the remainder of a polynomial division of the data and the initial value (which changes as bytes are fed into the calculation). We will be using CRC-16/CCITT with a polynomial of $x^{16} + x^{12} + x^5 +1$ or 0x1021 and an initial value of 0xFFFF. CRC will be calculated sequentially on each byte in the transmission except the 2 bytes containing the CRC value which we are calculating. The CRC value is in the 16 least significant bits of the data portion of a CRC packet. This means bytes 8-15 are always 0 in a CRC packet.

Manchester Encoding

Manchester encoding doubles the bandwidth needed for messages because it requires 2 bits for every 1 bit of unencoded data. Even though there is this drawback, there are benefits that make it the obvious choice for visual light communication. Manchester encoding provides two main advantages it eliminates the problem of AC-Coupling with a DC blocking capacitor and provides a data-plus-synchronization signal [6] . The AC-Coupling problem is eliminated because every bit requires a transition and therefore the data signal will never remain at a logic low or high state for an extended period of time [6] . The data-plus-synchronization signal means a synchronization of transmitter and receiver on separate vehicles can occur. Specifically, the transitions provided by Manchester provide information about when the signal should be sampled. This means that no separate clock signal is needed [6] . Manchester encoding also gives us a 50% duty cycle. This is significant because it allows us to match our non sending duty cycle to our sending duty cycle.

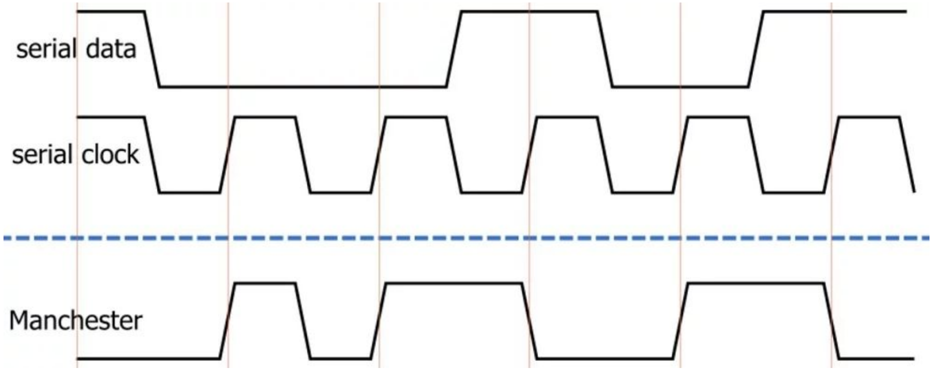


Figure 7. Serial data and clock represented as a Manchester encoded signal [6]

In the diagram above the serial data is converted to manchester data by representing a 1 as a high to low transition. A 0 is therefore represented by a low to high transition.

Complete transmission sequence

Transmission Overview: Send a transmission containing the vehicle's speed in miles per hour

Packet Structure: Send three 32-bit packets containing sending address, CRC, and speed data

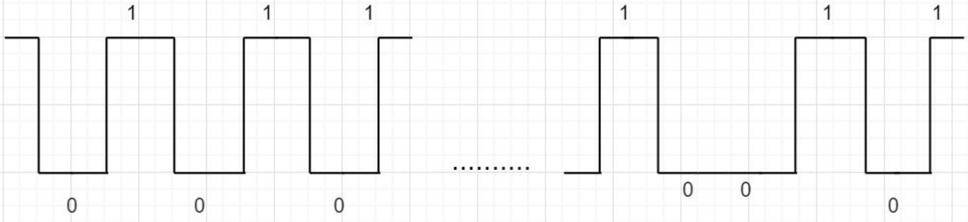
0:1	2:7	8:31
0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1
0 1	0 0 0 0 0 1	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1
1 0	0 0 0 1 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0

Manchester Encoded Message:

```
0101010101010101010101010101010101011001010101010101011001101001100110010101010110010101010
101010101011001100110010101010101101010100101010110011001010101010101010101010101010101
01101010100101
```

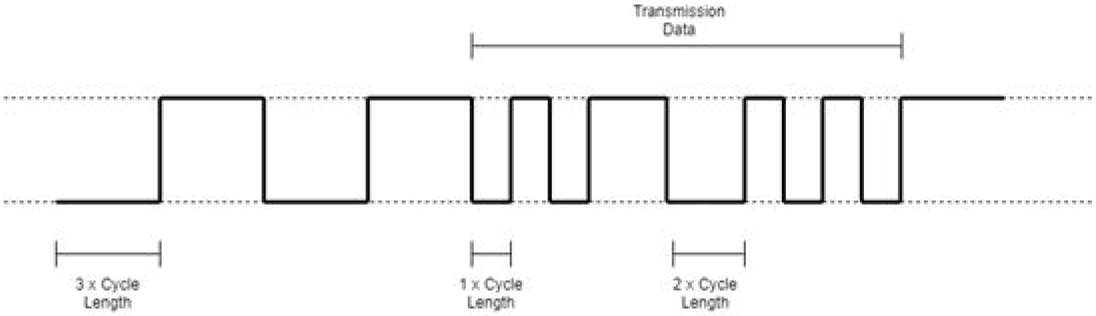
First 6 of Man

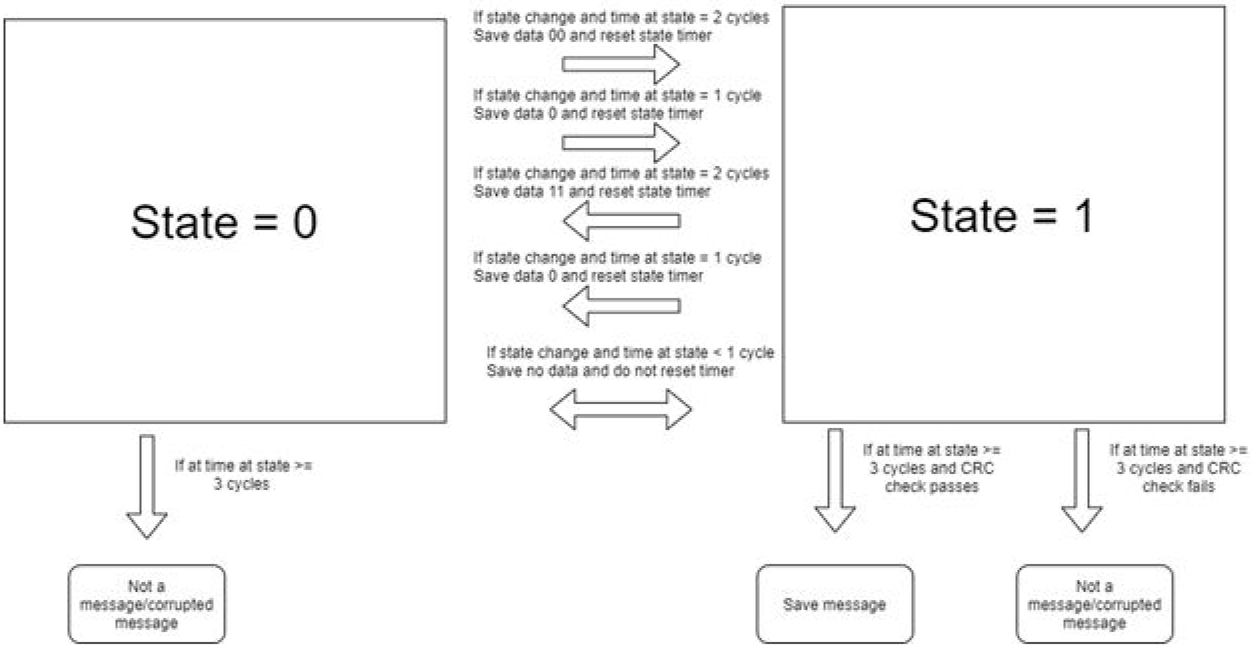
Last 6 of Man



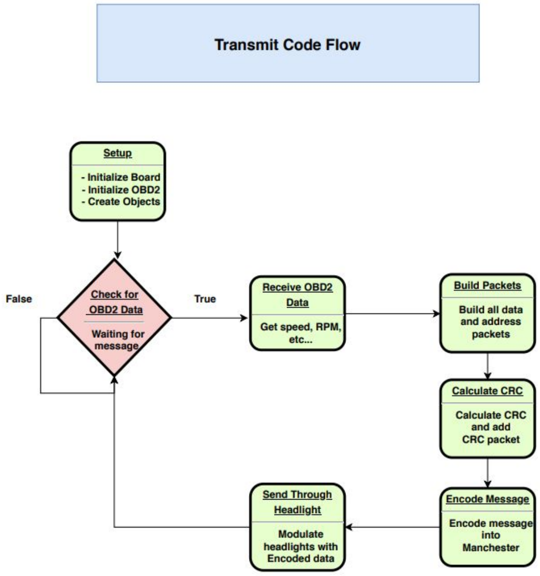
Data Acquisition

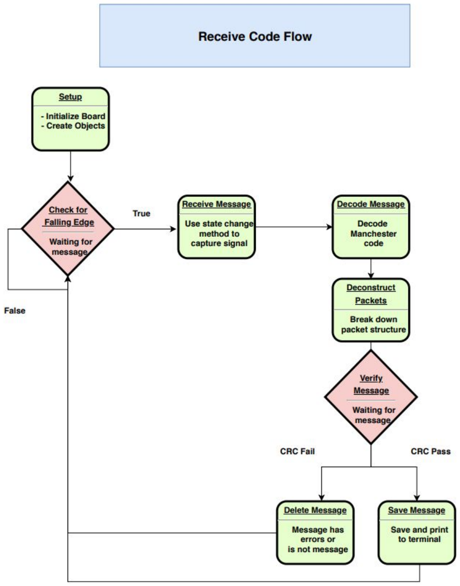
The original methodology for obtaining the signal in software was to sample the incoming signal from the receiving circuit once per cycle (every 1 / bit rate, seconds). We found this approach to be unsuccessful. Due to the low threshold of the Shmitt Trigger the signal “pulses” were irregularly shaped. High values would be wider than low values. This lead to constant misreads and double reads. Our solution is shown in the figure below. The idea is to constantly sample and look at state changes (1 to 0 or 0 to 1) and the time between those state changes. Based on this we are able to decipher the encoded data.





Simplified Software Flowcharts

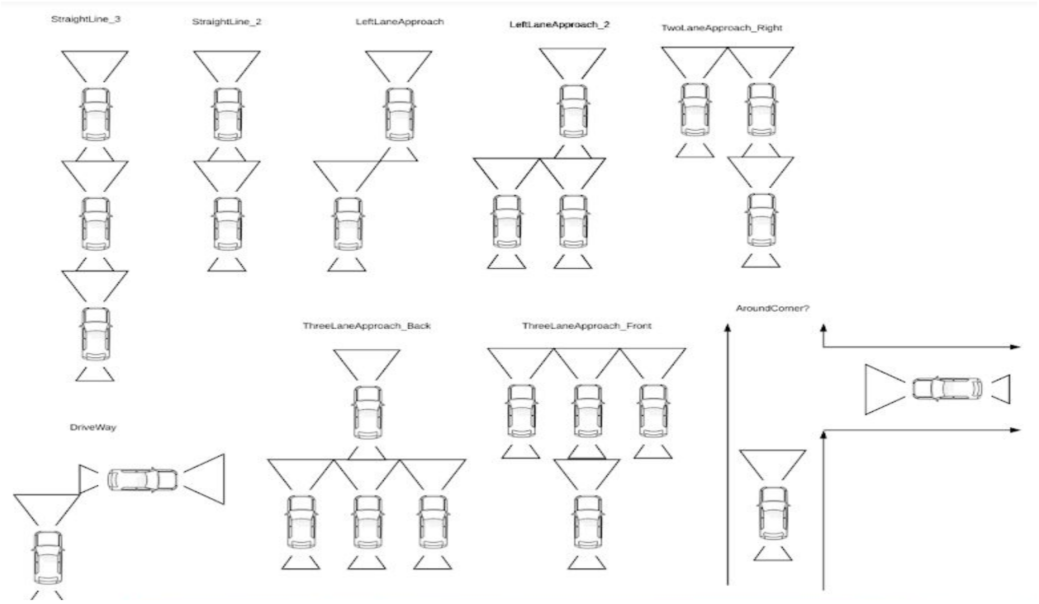




Applications

Visual Light Communication (VLC) is a technology that has potentially limitless applications and benefits. However, due to the large scope, this project aims to focus on a few of those with discussions on what may be possible in the future. As stated earlier on in the proposal, VLC is simply the communication of vehicles using the light emitted from the headlights and taillights. And through the use of the Manchester Encoding technique, we can send information through the light itself. At the completion of this project, we intend to be able to relay general car information such as speed, heading, and relative location. With this information, vehicular safety can be improved upon as long as that info is interpreted correctly. Ideally, ‘Car A’ would send its information to ‘Car B’ which would result in ‘Car B’ performing some action that ultimately decreases the chance of an accident. Upon completion of this segment, some future advancements and improvements would include applications such as the involvement of traffic lights, detecting light around corners/obstacles, and a network of intercommunicating self-driving cars.

In this project’s most basic form, being able to communicate between vehicles is what the goal is. In the diagram shown below, potential scenarios of importance are shown.



In the first scenario, ‘StraightLine_3’, some information that might be communicated would include the first car telling the second car that there’s a red light and that it should slow down. And then the second car would tell the third car to also slow down. Or something like the ‘TwoLaneApproach_Right’, the two forward cars would tell the back car that both lanes are being occupied. Should the back car want to pass, it could also communicate that such that the car ahead can move into the other lane. For the ‘ThreeLaneApproach_Back’, the back middle car could receive information that there are cars on either side and also in front. This would let the middle car know that it can’t change lanes at the moment. Something that is also important to our proposal is having the ability to receive and transmit around corners or obstacles. There are blind zones everywhere such as driveways or tight corners. In these instances, light can sometimes be visible before the actual vehicle is. If that visible light can be received and processed, this could further any safety measures. These are just a few examples of the type of information that could be communicated, but it’s very important to determine which information holds priority in determining how decrease the chances of a dangerous situation.

Moving on to future advancements and improvements, the addition of traffic lights or maybe even traffic signs with added transmitters could be used to improve safety. For example, in the event that a driver doesn’t slow down as their vehicle proceeds towards a red light, the traffic light could indicate to the car itself that it must slow down in order to avoid driving into a busy intersection. Or someone who’s stopped at a green light might need the opposite. The light would indicate to the stopped car that it must drive. In the case that there’s a vehicle approaching from behind the stopped car, this would prevent a rear end accident. Essentially, vehicles having the ability to receive data from traffic lights or signs could help improve safety as there are many incompetent and distracted drivers on the road.

Finally, a potential application of this technology is it being applied to self driving cars. Currently, self driving cars use a combination of sensors and cameras to simply detect its surrounding environment. While for the most part this has served to be satisfactory, there is always room for improvement. With the addition of VLC, these self driving cars have another safety measure. Should other vehicles be outside of range, maybe light transmissions are still visible, which effectively increases the overall range. Peace of mind is essential with self driving cars as humans are ultimately giving a machine control over the complicated and dangerous action of operating a motor vehicle. It’s also possible that with VLC, these self driving cars can form an extensive network as they travel. Being able to transmit and receive data among these autonomous vehicles has major benefits. Such data could improve traffic conditions, safety in conditions where other sensors may fail, and etc. It’s hard to know for sure as the possibilities are plentiful, but this technology has the opportunity to improve safety on the road in a huge way.

Expenses

Part	Quantity	Price	Purpose
Power MOSFET	5	$2.27	Modulate headlight
LED headlights	3	$100	Transmit light messages
LED Housing	1	$25.00	Enclosure for LED. Allowed us to focus the direction of light
PhotoTransistors + Photodiodes	~	$200	Receive Light Messages
Arduino	4	$71.48	Interpret messages received and control of headlight modulation
Thorlabs bandpass optical filters	~	$300	Filter out sunlight and other excess noise
Draper Labs Car	1	Borrow	Easily obtain and send messages from car such as a person stepping on the brakes
Mounting Kits	2	$100	Used to mount transmit and receive circuits to vehicles

Total: $798.75
Remaining: $201.25

The remaining budget will be used to package our transmitters and receivers into form factors that are friendly to test with. This includes buying DC-DC converters to reduce the amount of voltage inputs we have coming to our circuit. It also includes supplies for protoboards and headers and such.

Calendar

Milestones	**Objectives**
End of September - Milestone 1	Completed circuitry for photodiode biasing and signal filtering. Confirmed that the photodiode circuit is capable of accepting data at a speed of at least 2.4kb/s. Also used spectrometer to determine our headlights' light spectrum. After having completed these steps we ordered an optical filter and a more expensive photodiodes.
End of October - Milestone 2	Completed software implementation of communication framework outlined in this report. Ensured that a computer can send and receive understandable messages. Ensured that transmit distance is at least 10 feet.
Mid November - Milestone 3	Applied project to real world applications to obtain videos and supporting documentation to show capstone success. Completed a series of tests: Stationary car communication with light and receiver in car, moving car communication with light and receiver in car, stationary car communication with light and receiver outside of the car, moving car communication between two vehicles inside the garage, moving car communication between two vehicles on real road.
December - Milestone 4	Finished all necessary documentation and placed 3rd in Capstone competition.

References

[1] S. Vijay and K. Geetha, “A Survey on visible light communication appliances used in inter-vehicular and indoor communication ,” International Journal of Applied Engineering Research, vol. 11, no. 7, 2016.

[2] A. Jovicic, J. Li, and T. Richardson, “Visible light communication: opportunities, challenges and the path to market,” *IEEE Communications Magazine*, vol. 51, no. 12, pp. 26–32, 2013.

[3] C. Baird, “What is the Color of the Sun?,” Science Questions with Surprising Answers, 03-Jul-2013. [Online]. Available: https://wtamu.edu/~cbaird/sq/2013/07/03/what-is-the-color-of-the-sun/. [Accessed: 14-Aug-2019].

[4] “Photodiode,” Wikipedia, 21-Jul-2019. [Online]. Available: https://en.wikipedia.org/wiki/Photodiode. [Accessed: 14-Aug-2019].

[5] “Non-inverting Schmitt Trigger,” *Basic Electronics Tutorials*. [Online]. Available: https://www.electronics-tutorial.net/analog-integrated-circuits/schmitt-trigger/non-inverting-schmitt-trigger/. [Accessed: 14-Aug-2019].

[6] R. Keim, “Manchester Encoding: What Is It, and Why Use It?,” *All About Circuits*, 20-Sep-2018. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/manchester-encoding-what-is-it-and-why-use-it/. [Accessed: 14-Aug-2019].