In-Road Vehicle Speeding Monitor

ECE 445 Design Document
Arnav Das, Darius Haery, Noah Salk
Team 25
TA: Dhruv Mathur
2/23/2020

1 Introduction

1.1 Objective

Speed limits exist because authorities have determined that abiding by these limits makes driving safer for everyone. The most common method of enforcing speed limits utilizes police officers with radar guns and is problematic for multiple reasons:

> 1) Police officers are greatly outnumbered by the number of drivers on the road. This results in drivers breaking the speed limits when no police car is in sight and slowing down when one is spotted. The low risk of getting caught is a chance that most drivers are willing to take.
>
> 2) Police officers are trained to handle dangerous situations and keep civility. Having them sit in a car for hours on end waiting for someone to speed is a great under utilization of their abilities.

Consequently, our objective is to build a small packaged, speed measurement system that can be embedded in the middle of the lane so that cars can pass over it. The measurement system will include a Lidar proximity sensor to capture the speed of a vehicle and a small camera that can photograph the rear license plate. Computer vision software will be used to extract the license plate number to send a bill to the driver. Since our system will be able to be placed discreetly and in several locations, drivers will be more likely to adhere to the imposed speed limits. Furthermore, the license plate recognition software and billing system will not require any human intervention so police officers can focus on other important tasks.

1.2 Background

Speed trap cameras are in use worldwide and have been proven to help reduce speed-related car accidents, reducing fatal and serious injury crashes by 11- 44% [1]. However, the camera systems currently in use are fixed and overt, so it is easy for drivers to slow down only when approaching speed traps. We hope to solve this issue by making our system as discreet as possible.

Furthermore, current camera systems are typically placed on the side of the road, rendering them ineffective in multilane highways. Since we propose a system that can be embedded in-road, our approach is far more versatile. We hope that a covert and more portable speed trap can reduce speed related accidents even further.

1.3 High-Level Requirements

- The sensor and camera must be placed discreetly, so that the speed trap cannot be avoided by incoming drivers.
- The object sensor should be able to compute the average speed of a driver along a short distance with high precision.
- The camera should be able to snap a clear, well-lit image of the car's rear license plate, so that a computer vision system will be able to accurately recognize and record the number.

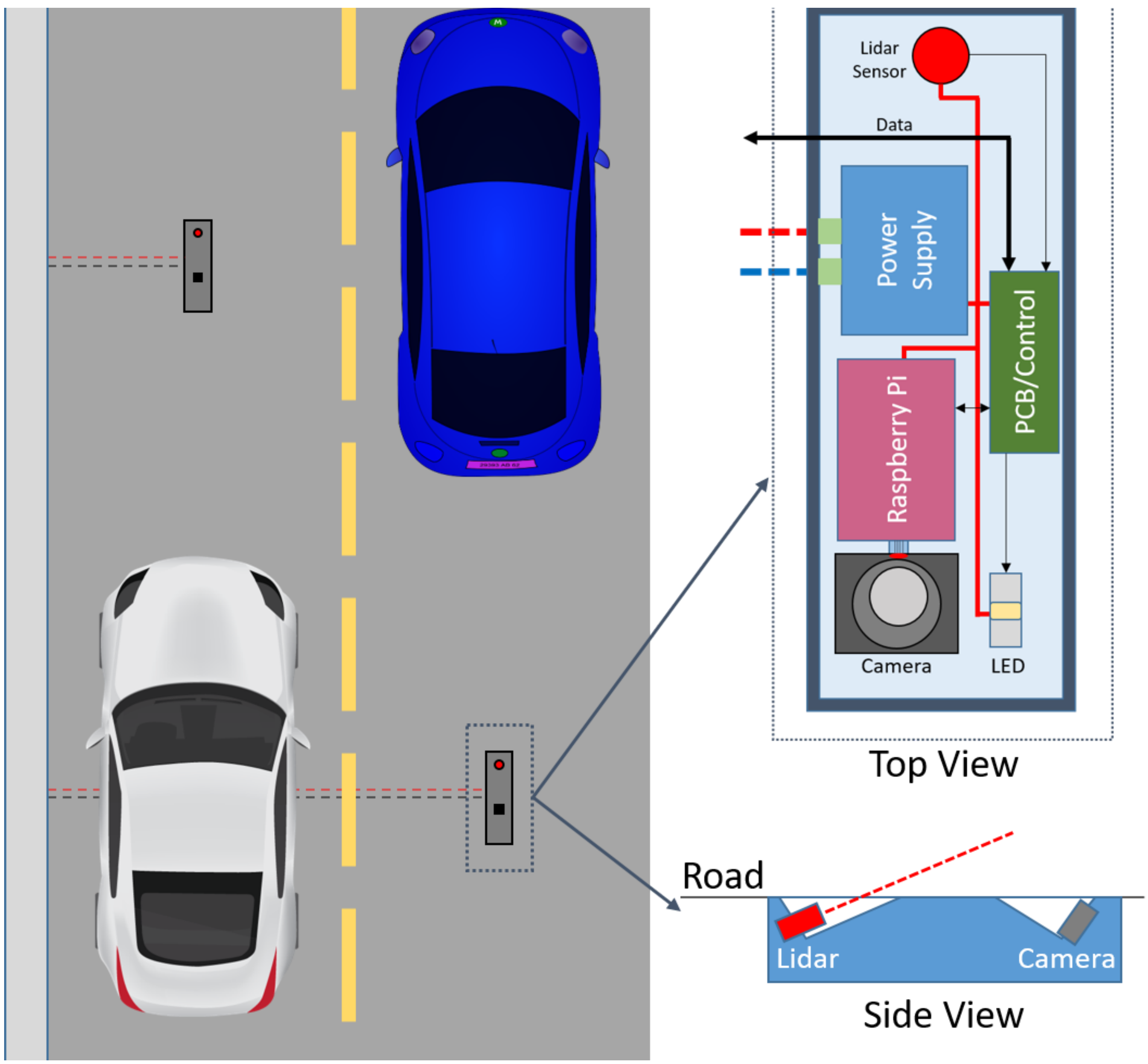

Fig. 1 - High-level graphical depiction of the design concept

2 Design

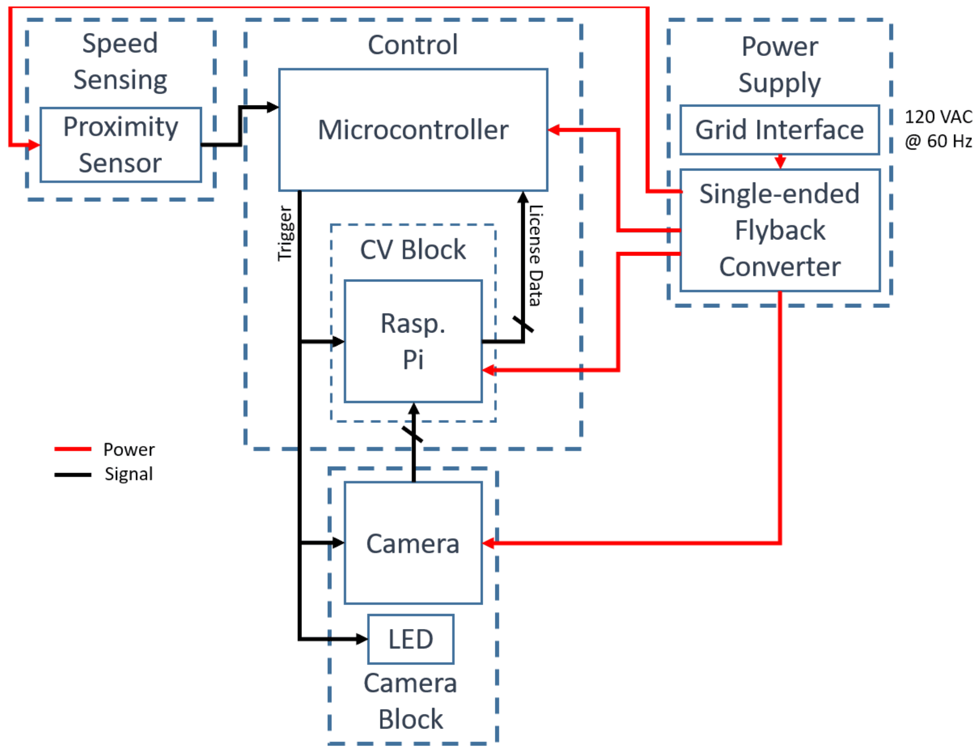

Fig. 2 - Block Diagram of Device

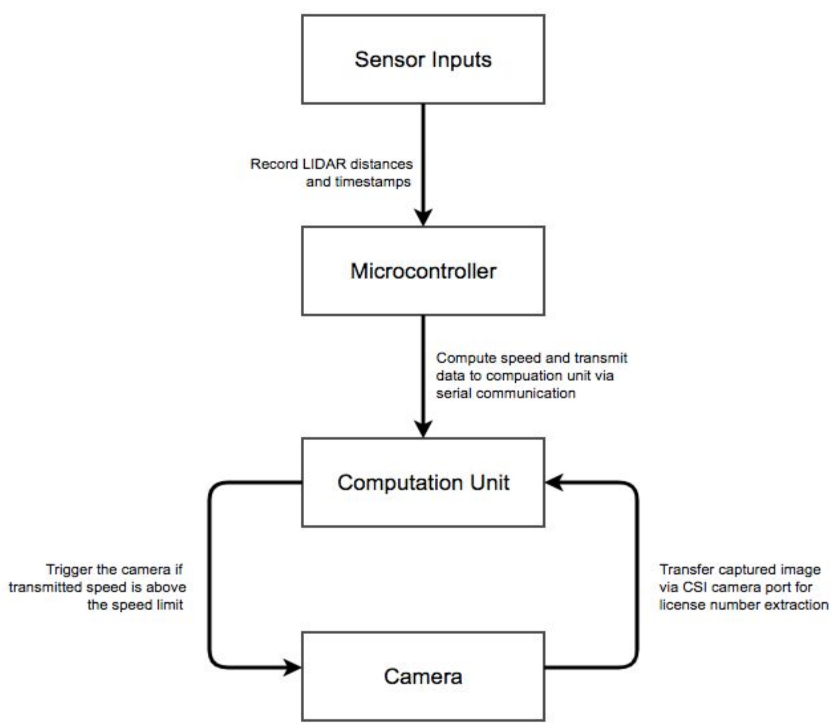

Fig. 3 - High Level Software Pipeline

2.1 Power Supply

The power supply is an interface between the power grid and our device's electronics. Similar to a "wall-wart" or phone charger, this power supply must condition grid power into a form that can be consumed by smaller electronics (i.e. a microcontroller). In our design, this grid power comes from the same lines that service street lamps and will therefore be 120 VRMS at 60 Hz. This AC must be safely rectified into 5 VDC for the Raspberry Pi, ATmega328p microprocessor, LED, and ultrasonic sensors to consume within the minimum ripple specification of all devices. To power all components in the design with the specifications listed in Table I, it's determined that the power supply should be rated for 25 W at 5 VDC with maximum 5% ripple.

Component	Maximum Current Rating (mA)	Voltage Ratings (V)
Raspberry Pi	1000	4.75 – 5.25
Raspberry Pi Camera Module	750	4.75 – 5.25
LED (CREE XLAMP XM-L2)	2000	2.75 – 3.3
ATmega328p Microprocessor	10	2.7 – 5.5
Garmin Lidar Lite V3	135	4.5 – 5.5

Table I - Component specifications

2.1.1 Topology

For islanding and user safety reasons, electrical isolation is required from input AC mains to the DC device-side output. For output power levels below 100 W this is typically accomplished with a flyback converter topology. A diagram of a self-sustaining flyback topology as used in AC-DC applications is shown in Fig. 3. The advantage of a flyback converter is two-fold. The "flyback" transformer is placed after a passive bridge rectifier and therefore can be operated at high frequencies, greatly reducing the size of the transformer as a large magnetizing inductance is no longer required. In a flyback topology we also take advantage of the parasitic energy storage element in the magnetizing inductance. Other isolated DC-DC converter topologies, such as a forward converter, require an inductor on the output as they don't utilize the inherent inductance of the transformer. The flyback topology therefore minimizes the number of magnetic circuit components to one (magnetic components are typically the largest part of an isolated converter).

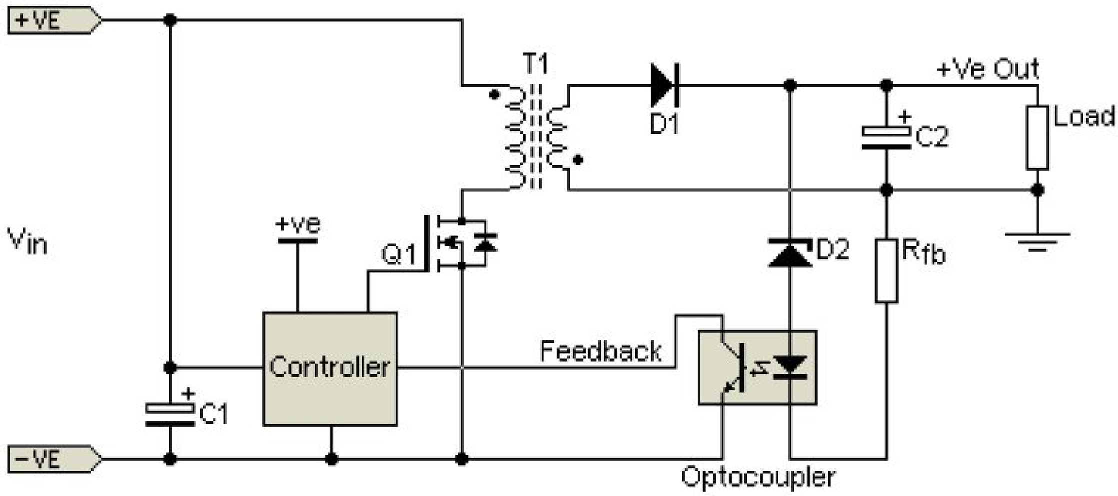

Fig. 4 - Switch-mode Flyback Power Supply, V_{in} is output of a passive bridge rectifier [ ]

2.1.2 Converter Components

For our power supply to be self sustaining, the converter controller needs to feed off of a DC voltage referenced to the same ground as the power MOSFET. This DC voltage is provided by a passive bridge rectifier directly at the input of the AC mains. Texas Instruments conveniently provides an integrated circuit (IC) meant solely for controlling flyback power supplies. The TI UCC28742 provides a typical application schematic shown in Fig. 4 that will be used as a reference for our design. Element value calculation methods are provided in the datasheet of the controller. The exact elements will determine the type specific optocoupler IC to use and therefore will be determined at a later date. The controller insures DCM operation in the flyback transformer and therefore limits the duty ratio to 45%.

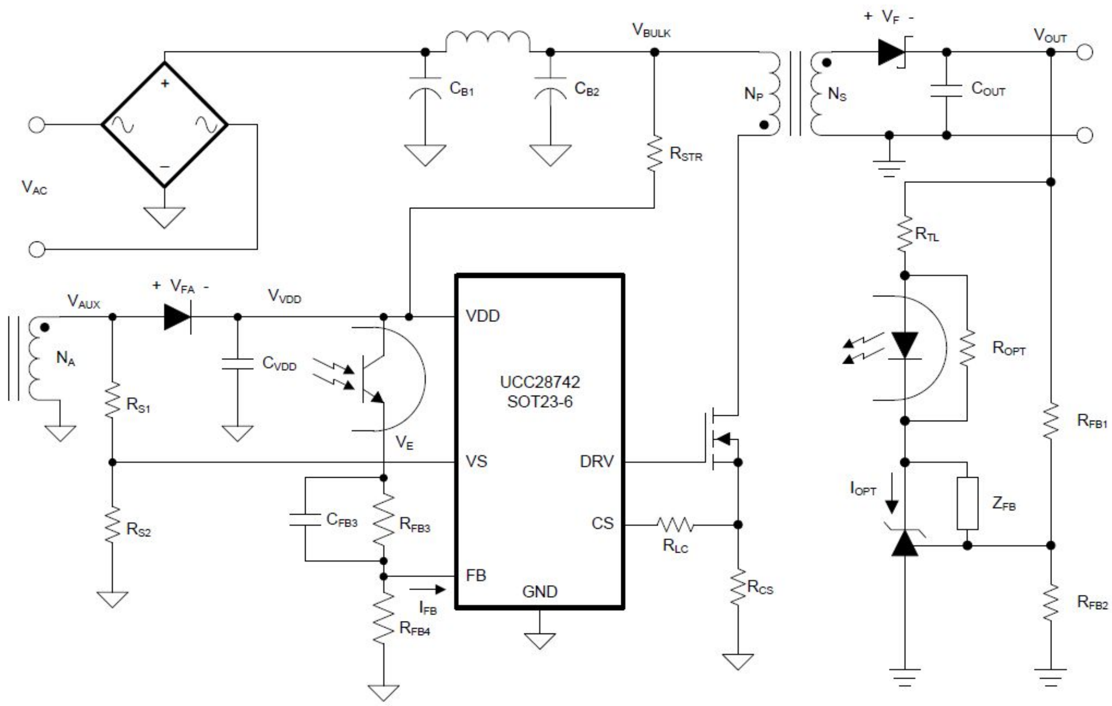

Fig. 5 - Typical application schematic from the TI UCC28742 datasheet [ ]

The flyback transformer has several requirements: it should be gapped to store energy in the air gap, it should have auxiliary windings for VDD input to the controller, and should have a turns ratio from primary to secondary that places results in the desired output voltage while maintaining a duty ratio below 50%. The following equation is used to calculate the ideal turns ratio for a given input voltage, output voltage, and duty ratio:

$$N_{PS} = \frac{V_{in,rms}\sqrt{2}}{V_{out}(1-D)}$$

2.1.3 Verification

To verify the correct operation of the power supply, unit tests will be run. First the optocoupler with driven with a current and the output voltage will be observed to ensure correct operation. If the optocoupler doesn't switch properly, the transformer core will saturate and burn up. To demonstrate the DC-DC operation of the flyback converter a DC voltage supply will be connected to the input of the converter and a 1 ohm resistor will supply rated load to the converters output. Finally, the full system will be tested together including the the passive rectifier with a 5 ohm output resistor at first and then moving to a 1 ohm resistor. Waveforms will be captured on an oscilloscope to validate that the output voltage is within spec.

Requirement	Verification
Optocoupler circuit switches properly.	1. Place 5 V across the resistor divider to get drive current through the input terminals. Make sure output switches as expected.
DC-DC Flyback converter operates with DC-link.	1. Supply 60 VDC and observe output voltage across a 1 Ohm load resistance. Output should be 1.75 V with a tenth of rated power. 2. Test rated condition with 170 VDC link.
Entire system output 5 V at 25 W within 5% voltage ripple ratio.	1. Add passive rectifier and low pass filter input filter to circuit. 2. Place 1 ohm load resistor across the output. 3. Observe voltage waveform across resistor to make sure it's within spec.

Table II - Power Supply R&V

2.2 Speed Sensing

To accomplish accurate speed sensing, a Lidar sensor will be placed at a set angle pointed toward oncoming traffic. Once a vehicle is detected within a range of 1.5 meters horizontally from the Lidar sensor, the horizontal distance will be measured and compared to the next sample. Based on the distance the vehicle traveled in a given sampling period, the vehicles speed will be calculated and recorded. Lidar sensors provide superior range and sampling rate compared to ultrasonic, photoelectric diffuse, and infrared sensors [ ], however, are substantially more expensive than the other sensor types. The limiting factor here is the sampling rate; if the speed monitor is to be used on highways the sensor must have as high of a sampling rate as possible. If the rate is too slow, the vehicle may pass over the device within the sampling period. For this reason, Lidar (which has an update frequency of 150-500 Hz) is the better choice over other sensor types which have 20-50 Hz update frequencies.

2.2.1 Microcontroller

The ATmega328P was determined to be the processor of choice for the control unit. Our system's microcontroller is required to read inputs from both ultrasonic sensors and send an output signal to the CV block, which can be accomplished through any of the ATmega328P's 23 programmable I/O pins.

2.2.2 Lidar Sensor

A candidate Lidar sensor was identified and is shown in Fig. 5. The Garmin Lidar Lite V3 sensor has a beam spread of maximum sample rate of 500 Hz and typical sample rate of 150 Hz and it's datasheet states that it can accurately measure distance up to 40 meters away. This opens up the design space, however, we only need a relatively short measuring distance in order to ensure that the car being measured is the same car we're going to take a picture of. Since the update rate of the Lidar sensor is below the sampling frequency of the ATmega328p's ADC, aliasing may be introduced. However, the distance a vehicle travels between sampling periods of the microprocessor is negligible compared to the distance the vehicle travels between update periods of the Lidar sensor and therefore can be ignored as long as we accurately record elapsed time between measurements. This calculation is shown below.

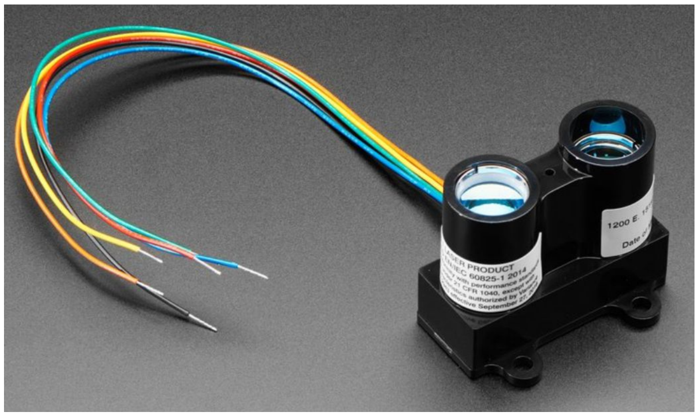

Fig. 6 - Garmin Lidar Lite V3 Sensor [ ]

Using the Lidar's sampling frequency of 150 Hz, we can calculate the distance a vehicle traveling at 120 mph goes with an update period. We ignore distance measurements until the Lidar senses that an object is at least 1.5 meters away. At this distance the light beam from the lidar should be incident with the top of the cars grill at the worst case scenario. This height is estimated as 80 cm and the undercarriage clearance is estimated as 40 cm. We can calculate the correct angle for the sensor to be placed at and can determine tolerances to make sure the light beam continues to be incident on the front of the cars grill throughout the update period of the Lidar. This calculation is shown below:

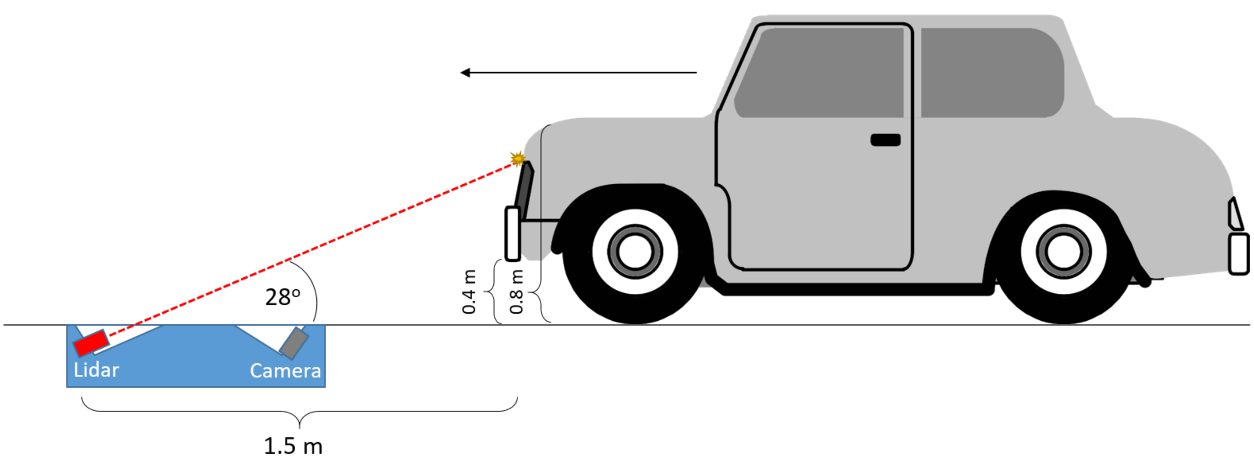

Fig. 7 - Measurement Validation

$$\theta_{lidar} = arctan(\frac{0.8m}{1.5m}) = 28^{\circ}$$

2.3 License Plate Number Extraction

2.3.1 Computation Unit

The computation unit is responsible for triggering the camera, receiving the input image, and running the software to extract the number from the license plate. For this task, we have chosen the Raspberry Pi 3 Model B+ shown in Figure 6.

Fig. 8 - Raspberry Pi 3 Model B+ []

The Raspberry Pi 3 Model B+ utilizes a 64-bit SoC with 1.4GHz and has 1GB of RAM, so the accuracy and speed of any CV software we use will not be subject to any significant hardware constraints. Moreover, this model has a built-in CSI camera port which simplifies the process of interfacing the computation unit with the camera module []. The maximum power consumption of the Raspberry Pi 3 Model B+ is not expected to exceed 980 mA, which is slightly under the limit imposed by the power supply as shown in Table 1 [].

2.3.2 Camera Module

The camera module will capture the image of the license plate and transmit it to the Raspberry Pi via a CSI camera port. We have identified the Raspberry Pi Camera Board, shown in Figure 7) to be the camera module of choice due to its high resolution, small size (0.98" x 0.90" x 0.35"), and compatibility with the Raspberry Pi 3. The camera is capable of taking 2592 x 1944 pixel static images and 640 x 480 p90 videos, both of which are acceptable formats for most algorithms supported by OpenCV [].

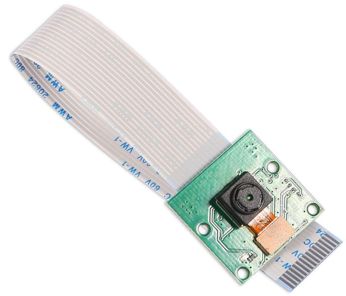

Fig. 9 - Raspberry Pi Camera Board [ ]

2.3.3 Computer Vision Software

All of the computer vision software required to process the image of the license plate will be implemented in the Python version of OpenCV, which is supported by our computational unit. Extracting the license plate number can be broken down into two main components: license plate segmentation and character recognition. The license plate isolation component will crop the license plate from the image, while the character recognition component will extract the number from the cropped image. Both of these steps are described at a high level below.

1) License Plate Segmentation:
    a) The first step is to remove any unnecessary information from the image. Fortunately, our camera will be placed in a stationary position for a long period of time making it easy for us to estimate the background, by intermittently collecting short videos and determining which pixels vary the least through time. By subtracting the background estimate from an image of the rear license plate, our new foreground image will contain only the back of the vehicle.
    b) The next step is to pass the foreground image through an edge detector and break it down into contours. We will loop through all of the larger contours and determine which ones form rectangles, since the license plate is guaranteed to be of this shape. All rectangular contours (there should not be too many of them) will be cropped from the image and passed onto the next step.

2) Character Recognition:
    a) OpenCV offers functions that use pretrained neural networks to both detect and recognize characters in a given image. If the license plate segmentation functioned properly, one of the rectangular segments that we cropped should contain the license plate number. Thus, we simply loop through the cropped images and see if any characters are detected. If no characters are detected in a segment, the cropped image can be discarded. If the image does contain characters, we assume that the image contains the license plate.

b) We pass the characters through a neural network for character recognition and record the output string locally on the Raspberry Pi.

We anticipate that the proposed algorithm should be able to extract license plate numbers of oncoming vehicles with high accuracy. However, it is possible that we may need to adjust the image preprocessing steps after some preliminary testing.

2.4 Tolerance Analysis

We expect that the speed detection will be limited by the LIDAR sensor's 150 Hz sampling frequency. In particular, shifts in the reflection point of the target may result in slight inaccuracies in estimating the exact distance travelled by the oncoming vehicle.

For example, suppose our LIDAR sensor collects two samples of a vehicle moving at 120 miles per hour (~53.6 meters per second) with the configuration shown in Figure 7.

We can compute the distance travelled by the vehicle between the two samples with the equation below:

$$d = \frac{v}{f_s} = \frac{53.6}{150} = 0.357m$$

The height of the reflection point with respect to the system shifts during the second sample since the vehicle is now closer to the sensor. The new reflection point is computed below:

$$h_{new} = (x_{init} - d)tan(\theta_{lidar}) = (1.5m - .357m)tan(28°) = 0.607m$$

Referring to the specifications detailed in Figure 7, we calculate the shift in the reflection point:

$$\Delta h = h_{init} - h_{new} = 0.8m - 0.607m = .193m$$

The reflection point does change significantly, but will remain on the front surface of the oncoming vehicle based on the vehicle dimensions specified in Figure 7. We also note that it is highly unlikely that our system will encounter a vehicle travelling faster than 120 miles per hour on a U.S road, so we use this example to compute an upper bound on the drop.

4 Ethics and Safety

When handling grid voltage (as our device is powered from the grid) safety is the number one priority. Commercial AC/DC converters (i.e. phone chargers) are considered safe as long as they aren't damaged. The property that makes these converters safe is their internal flyback transformer. The transformer provides electrical or galvanic isolation between the grid and the rest of the device. This isolation is a requirement for the AC/DC converter that we design. The converter should be encased so that no metal, liquid, or other components can come in contact with it.

The nature of our project involves the collection of potentially sensitive information about oncoming drivers, which raises some ethical concerns. It is important for us to respect the privacy of individuals and honor the confidentiality of any collected data as outlined by Rules #6 and #7 in the ACM Code of Ethics [ ]. To follow these ethical guidelines, we intend to collect and retain the minimum amount of data required to identify the individuals who are breaking the law and for them to be able to dispute a penalty if necessary. Our system will locally retain the following pieces of information:

1) The vehicle's license plate number
2) A photograph of the vehicle's rear license plate
3) The time and location of the incident

The license plate number is necessary for proper identification of the owner of the vehicle, so the individual who is responsible can be fined. The system will also retain the photograph and the time/location of the incident so that the individual has the ability to dispute the charge in the case of a misidentification. It is the responsibility of the authorities, who utilize and deploy this system, to ensure that the data is kept confidential and dispensed within the appropriate time frame depending on state statutes regarding automated license plate readers [ ].