



Simon Fraser University
8888 University Dr.
Burnaby, BC Canada
Email: lumos@sfu.ca

November06, 2012

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specification for an Automated High Beam System

Dear Dr. Rawicz,

The attached document describes the design specifications for the Automated High Beam System designed by Lumos Technologies. We have developed a system that will automatically adjust a driver's high beam headlights by detecting nearby vehicles through real-time video processing to prevent the bright light from blinding other drivers.

This design specification provides a low-level description of the High Beam System. It specifies how we plan to build and test our system and the technical details required to meet our functional specifications. This document will later serve as a design and verification guideline to ensure all requirements are met in the final product model.

Lumos Technologies consists of five talented and motivated engineering students: Alex Huang, Claire Liu, Linda Zhao, Sujin Lee, and Victor Mateescu. If you have any questions or concerns about our proposal, please do not hesitate to contact me by phone at (778) 233-8586 or by e-mail at lumos-tech@sfu.ca

Sincerely,

A handwritten signature in black ink, appearing to read "Linda Zhao", written in a cursive style.

Linda Zhao
Chief Executive Officer
LumosTechnologies

Enclosure: Design Specification for an Automated High-Beam System



Design Specification

Automated High Beam System





Design Specification for an Automated High Beam System

Linda Zhao

Chief Executive Officer

Sujin Lee

Chief Finance Officer

Victor Mateescu

VP Research & Development

Alex Huang

VP Software

Claire Liu

VP Operation

Contact Person:

Linda Zhao

lza28@sfu.ca

Submitted To:

Dr. Andrew Rawicz– ENSC 440

Dr. Steve Whitmore– ENSC 305

School of Engineering Science

Simon Fraser University

Issued Date:

November 6, 2012

Executive Summary

High Beam headlights were intended to illuminate the driver's range of view. This is done by aiming a long range, high intensity light in an upwards direction, designed to illuminate trees, signs, and any distant obstacles. However, this technology is potentially hazardous if the bright light shines directly into the eyes of an oncoming driver. Proper use of high beams requires careful micromanagement from the driver to prevent such dangerous situations from occurring. Hence, a system that automatically manages the high beams for the driver would allow drivers to devote more attention to the road.

The Automated High Beam System is designed to assist drivers at night by detecting oncoming cars and automatically changing the brightness of the high beam so that they don't blind others on the road. Our system consists of a camcorder to record live traffic, a processor to detect and calculate the distance of the cars from the user, and a hardware circuit that connects the output of the processor to the original car's headlight system. Once a car in front of the driver is detected, the processor will calculate the distance to that car and decide on the appropriate brightness of the high beam. The system is also able to detect the overall brightness of its surroundings and turn off the high beam when it is unnecessary. The purpose of our product is to allow drivers to focus more on the road, especially when the circumstance requires high beam headlights.

The team at Lumos Technologies has members who specialize in video processing, computer and systems engineering, and electrical hardware development. We are committed to delivering a safe and simple solution that will improve the safety of all drivers who share the road.

Table of Contents

EXECUTIVE SUMMARY	I
TABLE OF CONTENTS	II
LIST OF FIGURES	IV
LIST OF TABLES	V
GLOSSARY	VI
1. INTRODUCTION	1
1.1 SCOPE.....	1
1.2 INTENDED AUDIENCE	1
2. SYSTEM OVERVIEW	2
3. HARDWARE	4
3.1 HARDWARE SYSTEM OVERVIEW	4
3.2 VIDEO CAMERA	4
3.3 ALTERA DE2-70.....	5
3.3.1 <i>Hardware System Overview</i>	7
3.3.2 <i>Video-In Decoder</i>	8
3.3.3 <i>DMA Controller</i>	8
3.3.4 <i>Video Image Processing Bridges</i>	9
3.4 AUTOMOBILE RELAYS.....	10
3.5 <i>Headlights</i>	13
3.8 MECHANICAL	15
3.8.1 SWITCH.....	15
3.8.2 CAMERA STAND.....	16
4. SOFTWARE	18
4.1 SOFTWARE SYSTEM OVERVIEW	18
4.2 THRESHOLDING.....	20
4.3 HSV FILTERING.....	23
4.4 OBJECT DETECTION	24
4.5 DISTANCE ESTIMATION.....	27
4.6 DECISION-MAKING AND OVERRIDING	28

5. TEST PLAN	31
5.1 UNIT TEST	31
5.1.1 <i>Automobile Relay</i>	31
5.1.2 <i>Battery</i>	32
5.1.3 <i>Headlights</i>	32
5.1.4 <i>Multifunction Switch</i>	33
5.1.5 <i>Circuit</i>	33
5.1.7 <i>Video Camera</i>	34
5.1.8 <i>Camera Car Stand</i>	34
5.1.9 <i>Altera DE2-70 Board</i>	34
5.1.10 <i>Software System</i>	35
5.2 INTEGRATION AND REQUIREMENTS TEST.....	35
5.4 FAILURE TEST	37
5.5 ENVIRONMENTAL CONSIDERATIONS.....	38
6. CONCLUSION	39
7. REFERENCES	40

List of Figures

FIGURE 1: SYSTEM OVERVIEW	2
FIGURE 2: CANON VIXIA HF M40 HD CAMCORDER [1]	4
FIGURE 3: DE2-70 DEVELOPMENT AND EDUCATION BOARD LAYOUT, SIZE: 153X203 MM [4].....	5
FIGURE 5: VIDEO-IN DECODER CORE'S BLOCK DIAGRAM [5]	8
FIGURE 6: BLOCK DIAGRAM FOR DMA CONTROLLER [5].....	9
FIGURE 7: VIDEO-STREAM BRIDGE: RAW TO VIP CORE'S BLOCK DIAGRAM [5]	9
FIGURE 8: ADDRESS FORMAT BASED ON PIXEL'S DATA WIDTH [5]	10
FIGURE 10: 896H-1CH-D1 SPDT RELAY [6]	11
FIGURE 11: 896H-1CH-D1 RELAY OUTLINE DIMENSION [7].....	11
FIGURE 12: 896H-1CH-D1 RELAY WIRING DIAGRAM [7]	11
FIGURE 13: DUAL BULB HEADLIGHTS [8]	13
FIGURE 16: A DIP SWITCH WITH AN AUTO BUTTON ATTACHED.....	16
FIGURE 17: CAR MOUNT ACTUAL MODEL [10]	16
FIGURE 18: CAR MOUNT OUTLINE DIMENSION [10]	17
FIGURE 19: OVERVIEW OF SOFTWARE SYSTEM.....	19
FIGURE 20: ORIGINAL FRAME (LEFT) AND BINARY MAP OF BRIGHT OBJECTS (RIGHT)	22
FIGURE 21: (A) COLOR THRESHOLD REGION (B) DISTRIBUTION OF TAIL LIGHT COLORS FROM DATABASE OF 300 IMAGES [12]	23
FIGURE 22: ORIGINAL IMAGE (LEFT) BINARY MASK $B(X,Y)$ AFTER THRESHOLDING (RIGHT) HSV FILTER RESULT $B_R(X,Y)$ (MIDDLE) [14]	24
FIGURE 23: INCORRECT CLUSTERING OF HEADLIGHTS SHOWN BY THE RED RECTANGLE	26
FIGURE 24: INCORRECT CLUSTERING OF MINOR FEATURES SHOWN BY THE RED RECTANGLE	27
FIGURE 25: FLOWCHART OF THE DECISION-MAKING PROCEDURE	29
FIGURE 26: RELAY COMPONENT TEST SETUP	31

List of Tables

TABLE 1: ALTERA DE2-70 BOARD INFORMATION [2]	6
TABLE 2: 896H-1CH-D1 SPECIFICATIONS [6] [7]	12
TABLE 3: HSV RED FILTER PARAMETERS	23
TABLE 4: POSSIBLE OUTPUTS OF THE DECISION- MAKING ALGORITHM	28

Glossary

ADC	Analog to Digital Converter
DMA	Direct Memory Access
DSP	Digital Signal Processing
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
GPIO	General Purpose Input/Output
HD	High Definition
HSV	Hue Saturation Value
IP Core	Intellectual Property Core
LED	Light Emitting Diode
NTSC	National Television System Committee
RGB	Red Green Blue
SDRAM	Synchronous Dynamic Random Access Memory
SOPC Builder	System On a Programmable Chip Builder
SPDT	Single Pole, Double Throw
SPST	Single Pole, Single Throw
SSRAM	Synchronous Static Random Access Memory
PAL	Phase Alternating Line
VGA	Video Graphics Array
VIP	Video Image Processing
UP	University Program

1. Introduction

The Automated High Beam System is an automobile integrated system that provides a safer driving environment and prevents vehicle accidents. It is designed to be convenient and reliable to suit the daily needs of people who often drive at nights with minimal lighting conditions. The system will turn off the high beams when there is a vehicle nearby in front of the user and turn on the lights when the road ahead is clear. The design specifications for the proof-of-concept model of the high beam system are outlined in the following document.

1.1 Scope

This document outlines the design specifications to achieve the requirements listed in our document outlining the functional specifications for the Automated High Beam System. Only specifications of first and second priority requirements will be adhered to.

1.2 Intended Audience

This document is intended for use by the engineers at Lumos Technologies and all others who are involved in constructing the Automated High Beam System. It will act as a guideline and provide test cases to ensure all proof-of-concept design requirements are met.

2. System Overview

A high level overview of the system, outlining the major components and their place within the system, is illustrated in Fig. 1. A more detailed explanation of individual hardware and software components is presented in the following sections.

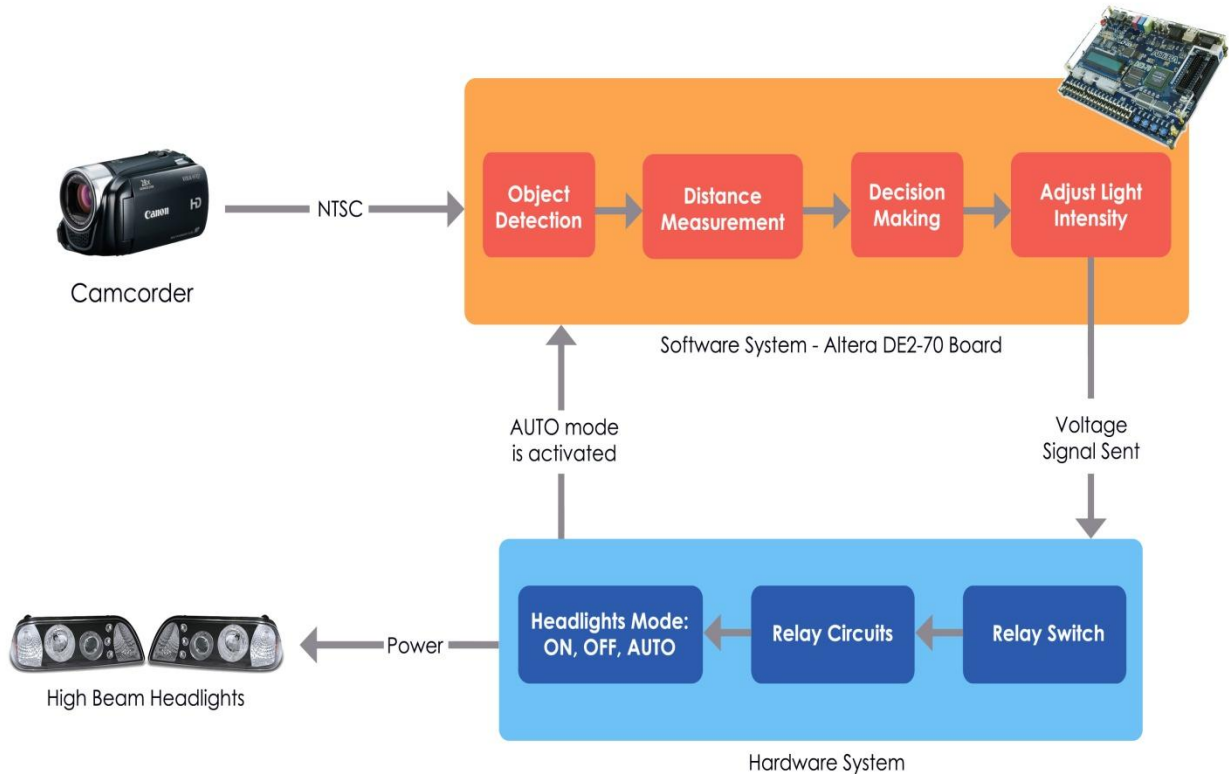


Figure 1: System Overview

The Automated High Beam System concept model comprises of two main components: the software in the FPGA, which detects vehicles and controls the high beam light intensity, and the hardware system, which executes the decision output from the software system.

Decisions to dim or toggle the high beam will be made by the video processing system and will be based on the estimated distance and the light intensity of the surroundings. The system will revert the high beams to their original intensity when the high beam range is clear of other vehicles. The driver will have three options to control high beam headlamps: “on”, “off”, and “auto”. The “on” and “off” states will function as any traditional headlamp controls. This will grant the driver manual control over the headlamps, thus completely overriding our system.

When the system is activated in the “auto” state, the video camera will constantly record the front view of driver’s vehicle and send the information to the processing unit. This information will be analyzed to decide when to open and close the switch. As soon as an oncoming vehicle is detected to be within range, the video processor signals the high beam circuit to reduce the intensity of the headlamps. When the range in front is clear of vehicles, the circuit restores the headlamp intensity.

3. Hardware

3.1 Hardware System Overview

The hardware section includes four main parts: a video camera, DE2-70 board, relays, and high beams. Detailed descriptions and specifications of each part will be given in the upcoming subsections along with pictures and a circuit diagram.

3.2 Video Camera

Any type of video recording device with supported drivers and composite video output is feasible for this system. For this proof-of-concept model, a Canon VIXIA HF M40 HD camcorder, shown in Fig. 2, is used because of its wide range of capabilities. This camcorder provides features including focus assistance, shutter speed and aperture controls, which are particularly useful for filming at night. Critical parameters for our input video such as exposure and sharpness are sufficiently controlled by these features. The high definition capabilities of up to 1920x1080 resolution supported by this camcorder, and the availability of 60 FPS recording rate make it an outstanding option for our system [1].



Figure 2: Canon VIXIA HF M40 HD camcorder [1]

3.3 Altera DE2-70

“The Altera DE2-70 Development and Education board is a modified version of the DE2 board with a larger FPGA and more memory. It is an excellent vehicle for learning about digital logic, computer organization, and FPGAs” [2]. Fig. 3 shows a photo of the DE2-70 board featuring the Altera Cyclone II 2C70 FPGA, with clock speeds of up to 50MHz [2]. The combination of this processor, 64 MB SDRAM, and 2 MB SSRAM on-board memory gradually reduces the runtime for video processing. After modifying the DE2-70 basic computer using the SOPC builder with the addition of video processing hardware from *Altera University Program IP Cores*, this board is capable of resampling and streaming live video as input. Details on *Altera University Program IP Cores* used in our system will be discussed in detail later on. The Altera Monitor Program “provides an easy way to assemble and compile Nios II programs on the DE2-70 Basic Computer that are written in either assembly language or the C programming language” [3]. This makes the development process much easier for engineers at Lumos Technologies. Its DSP capabilities and the publically available IP cores make the DE2-70 board a suitable choice for this proof of concept model.

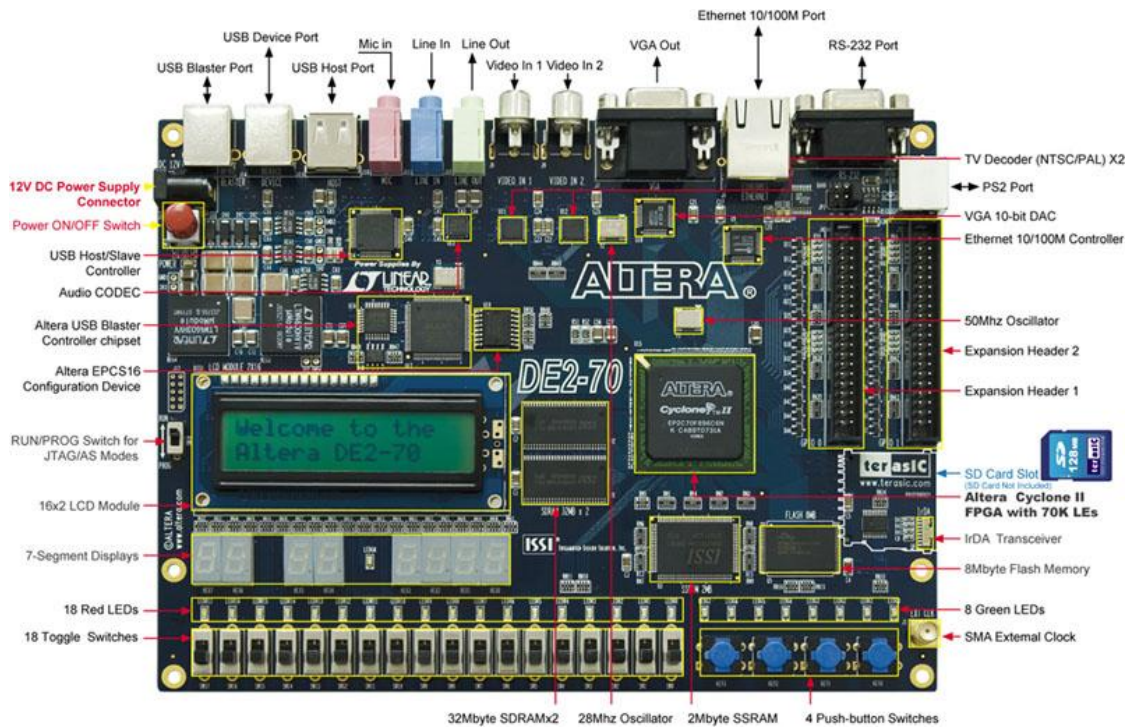


Figure 3: DE2-70 Development and Education Board Layout, size: 153x203 mm [4]

Table below is a summary of Altera DE2-70 development and education board specification and features.

Table 1: Altera DE2-70 Board Information [2]

Feature	Description
FPGA	<ul style="list-style-type: none"> • Cyclone II EP2C70F896C6 with EPCS64 64-Mbit serial configuration device
I/O Interfaces	<ul style="list-style-type: none"> • Built-in USB-Blaster for FPGA configuration • Line In/Out, Microphone In (24-bit Audio CODEC) • Video Out (VGA 10-bit DAC) • Video In (NTSC/PAL/Multi-format) (2 ports) • RS232 • Infrared port • PS/2 mouse or keyboard port • 10/100 Ethernet • USB 2.0 (type A and type B) • Expansion headers (two 40-pin headers)
Memory	<ul style="list-style-type: none"> • 64 MB SDRAM, 2 MB SSRAM, 8 MB Flash • SD memory card slot
Displays	<ul style="list-style-type: none"> • Eight 7-segment displays • 16 x 2 LCD display
Switches and LEDs	<ul style="list-style-type: none"> • 18 toggle switches • 18 red LEDs • 9 green LEDs • Four debounced pushbutton switches
Clocks	<ul style="list-style-type: none"> • 50 MHz clock • 27 MHz clock • External SMA clock input

This Altera DE2-70 board will be primarily used to receive input video, buffer frames, execute video processing, make decisions, and output voltage signals. For this proof of concept model, the camcorder will be utilizing the composite port while the outputs are sent through general purpose input/output (GPIO) port.

3.3.1 Hardware System Overview

The Altera DE2-70 basic computer includes a Nios II processor, memory for data storage, parallel ports connected to switches and lights, a timer module, and a serial port [3]. In addition to these parts, the major cores used for the Automated High-Beam System are Video-In decoder, two direct-memory-access (DMA) blocks, and Video Image Processing (VIP) bridges [5]. A high level overview of the board's structure is shown in Fig. 4. As required by our system, "the Video-In Decoder IP core essentially converts the video data from the ADC into Avalon streaming packets and sends them to the video DMA controller" [5]. Then the video DMA controller writes the video stream to the pixel buffer (SRAM/SSRAM). The integration of selected cores essentially converts the input video into acknowledgeable information for the processor to execute our software. Finally, the output voltage representing the decision will be sent through GPIO pins to the remaining circuitry. Indications for any type of errors and power-on status will be signalled by the LEDs on the board.

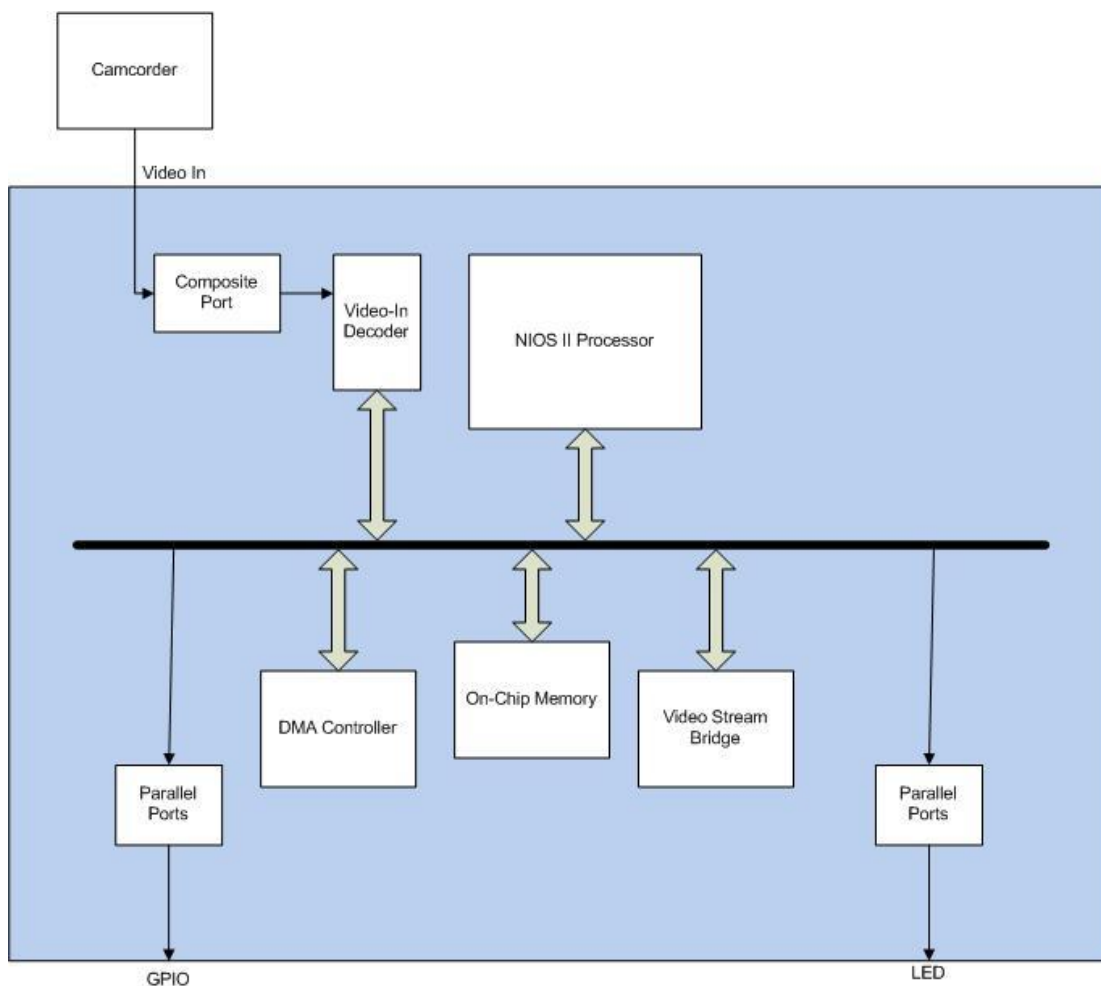


Figure 4: High-level FPGA block diagram

3.3.2 Video-In Decoder

A block diagram of the Video-In decoder is shown in Fig. 5. On the left hand side, the system initially receives video input through a composite video port on DE2-70 board. The video is then decoded, and synchronized from the video source's clock domain to the system's clock domain [5]. Finally, the video is converted into Altera University Program video streaming packets as the output shown on the right hand side. The only disadvantage for this core is that the video packet size for On-Board Video NTSC and PAL are limited to 720x244 and 720x288, respectively [5]. The vehicle detection algorithm implemented in this system may function better at higher resolutions.

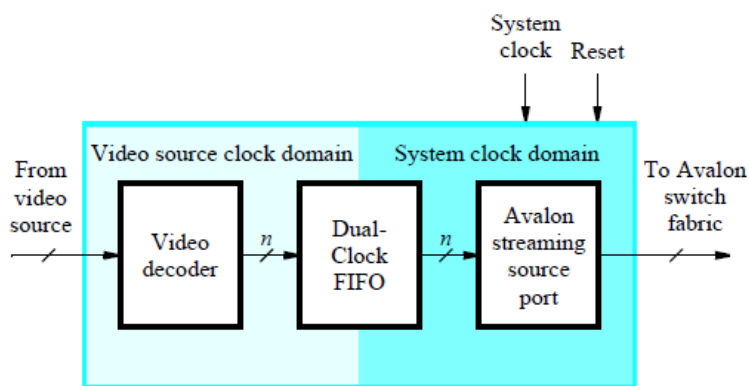


Figure 5: Video-In Decoder Core's Block Diagram [5]

3.3.3 DMA Controller

The DMA Controller IP core stores and retrieves video frames to and from memory and features two modes of operation: "from stream to memory" and "from memory to stream". These two modes use Avalon memory-mapped master interface to send the data to the memory or to read video frames from memory [5]. The block diagram in Fig. 6 shows the basic concept and logic flow of this core.

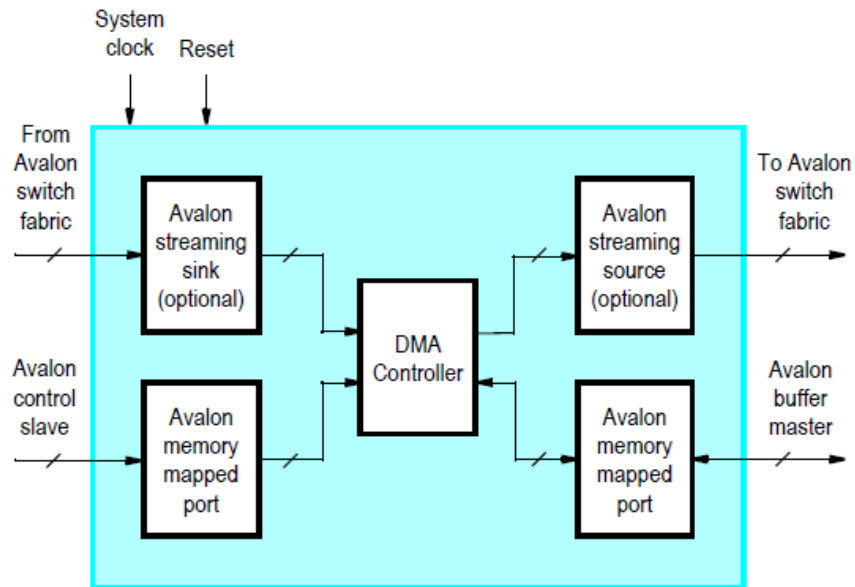


Figure 6: Block diagram for DMA Controller [5]

3.3.4 Video Image Processing Bridges

The VIP Bridges IP core supports bidirectional conversion between the Altera VIP format and UP video stream format, named “Raw to VIP” and “VIP to Raw”. As expected, the Raw to VIP Bridge takes an input stream in the University Program (UP) video stream format and converts it to the VIP format. Then, the VIP-formatted data is outputted via the Avalon streaming source interface as shown in the block diagram in Fig. 7 [5]. Similarly, the VIP to Raw Bridge works through the exact same procedure except in the reverse direction. Its block diagram can also be presented by Fig. 7 with the middle block replaced by “Stream Bridge VIP to Raw”.

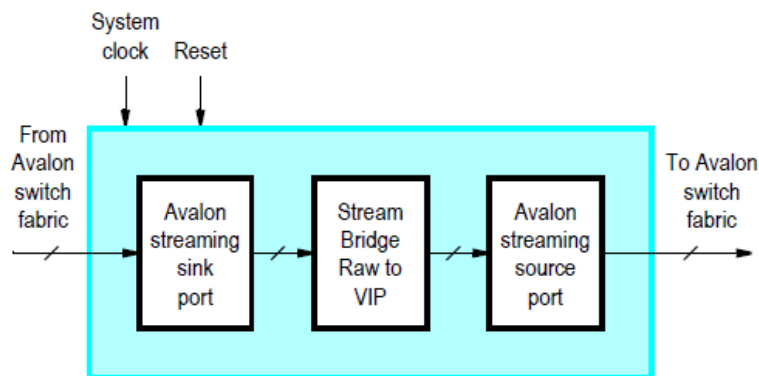


Figure 7: Video-Stream Bridge: Raw to VIP core’s block diagram [5]

The UP video stream format mentioned above is essentially a type of raw video data. The generic addressing for 16-bit and 32-bit color pixel formats is shown in Fig. 8. During the transmission, video frames are sent one pixel at a time in row-major order. “The first pixel – the top left pixel in the frame –

is signalled by the start-of-packet bit in the Avalon Streaming interface. The last pixel, the bottom left pixel in the frame, is signalled by the end-of-packet bit in the Avalon Streaming interface" [5]. Packet representation for streaming video is illustrated in Fig. 9.

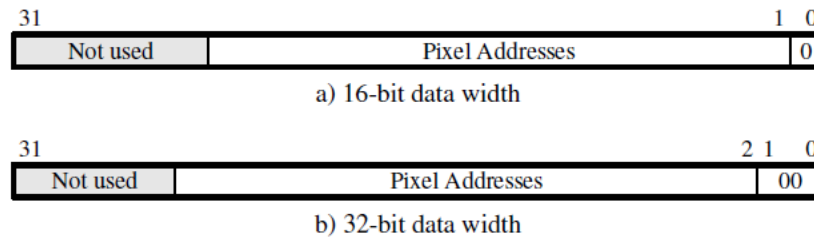


Figure 8: Address format based on pixel's data width [5]

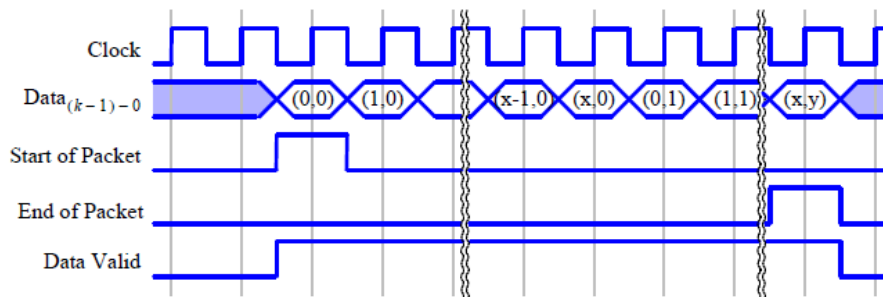


Figure 9: Video frame's streaming packet format [5]

3.4 Automobile Relays

Since the headlight circuit draws high current (normally 5A) for high beams, typical automobile SPDT relays are used to switch between low beam and high beam headlights as shown in figure 10. To minimize the voltage differences over the circuit branches due to coil resistance, the 896H-1CH-D1 SPDT relays are equipped for both SPST and SPDT relay.

The relay has two main functions and each is controlled differently. It is used to turn on/off the headlights through a manual multifunction switch operated by a driver. Similarly, the relays switch between 4 different circuits varying the brightness of the high beam. The brightness control relays are operated by 2-bit signals from FPGA.



Figure 10: 896H-1CH-D1 SPDT Relay [6]

Overall dimension of the relay is 25.8mm × 31.3mm × 46.0mm as shown in Fig. 11. To fit in 6 relays into the vehicle, they can be arranged to 1 by 6 or 2 by 3.

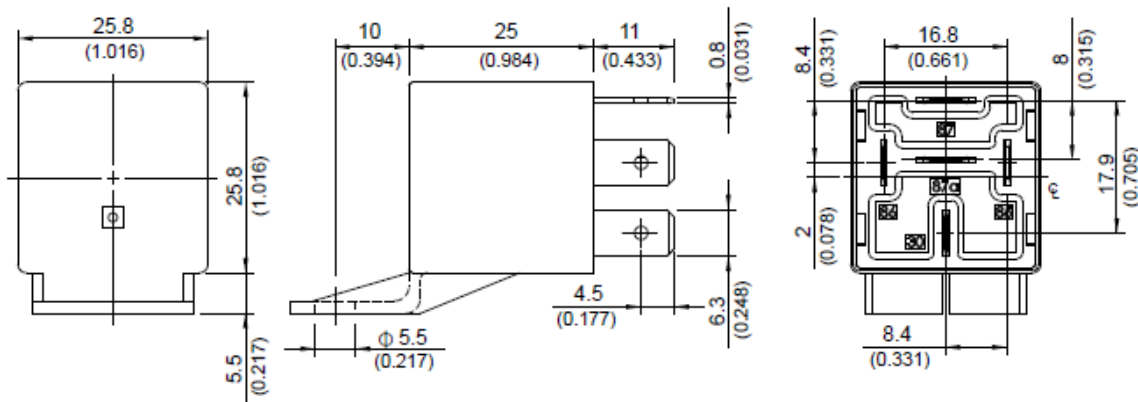


Figure 11: 896H-1CH-D1 Relay Outline Dimension [7]

The relay consists of 5 pins. Pin 85 and 86 are connected to an inductor, which can pull the switch from pin 87 to pin 87a when magnetized.

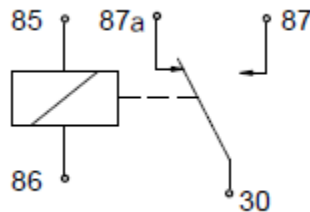


Figure 12: 896H-1CH-D1 Relay Wiring Diagram [7]

The following table is a list of the 896H-1CH-D1 specifications.

Table 2: 896H-1CH-D1 Specifications [6] [7]

Contact material	AgSnO alloy	
Contact voltage drop	Typ. 50mV at 10A	
Insulation resistance	20M Ω min. (DC 500V)	
Operate time	20ms Max.	
Release time	20ms Max.	
Dielectric strength	Between open contact	: AC 500V, 50/60Hz 1 min.
	Between contact and coil	: AC 500V, 50/60Hz 1 min.
Vibration resistance	Operating extremes	10~55Hz, amplitude 2mm
Shock resistance	Operating extremes	10G
	Damage limits	100G
Life expectancy	Mechanical	10,000,000 operations (frequency 18,000 operations/hr)
	Electrical	100,000 operations (frequency 1.200 operation/hr)
Temperature range	Operating	-40~+125 $^{\circ}$ C (no freezing)
Weight	Approx. 40 g	
Contact form	SPDT	
Coil voltage	12 VDC	
Coil resistance	90 Ω	
Coil current	133mA	
Power consumption	1.6W	

3.5 Headlights

Generally, every car comes with 2 headlights that include low beam and high beam functionality. The two beams are either installed as two separate bulbs or as a dual bulb. Single or dual bulbs depend on the vehicle manufacturers, but this could hinder the performance of the system. For the best performance, the final marketable product will come with two 60W halogen bulbs to ensure the safety and quality of our system.



Figure 13: Dual Bulb Headlights [8]

3.6 Battery

Many electronic parts inside a car, including the headlights, are power by the 12V car battery. It is a stable DC source that will be used to power the auto high beam system as well. Again, various car manufacturers will have different car battery requirements. The system hardware system is designed so that as long as the battery is sufficient to power the halogen headlights, it will work for our system.



Figure 14: 12V Car Battery

3.7 Relay Circuit

The relay circuit is the electrical circuit that connects all the hardware components together. This circuit was meant to integrate into the original headlight system. The 12V car battery powers the circuit shown

in Fig. 15. Aside from the original multifunction switch the driver is already familiar with, 4 additional input signals are included. One of these signals is from the auto button that driver uses to turn on the system. This signal will control relay RY3. The other three signals are from the FPGA's 5V GPIO headers. One of the GPIOs is used to provide feedback to the driver after turning on the auto system. The software will output a 1 after it the processor has finished booting up and has checked that the camera is also ready. The user will know that the system is ready to work when this LED turns on. A 250Ω resistor is placed before the LED to limit the current to within 20mA. The other two GPIOs form a two-bit output signal used to control the brightness of the high beam headlights. The most significant bit will switch RY4 to turn the high beam on or off. When the high beam is one, the least significant bit will switch RY5 between two circuits. One will turn the high beam on to maximum intensity. The other will have a resistor to lower the current and therefore reduce the brightness of the headlight. Note that relay drivers are used for RY4 and RY5 to protect the FPGA board. Each driver consists of a diode and a transistor. The diode will protect the FPGA from back-EMF stored in the relay coils by forcing the current flow into the forward direction when the system shuts off. The transistors Q_1 and Q_2 are NTE123AP NPN transistors. These transistors act as switches to be triggered by the GPIOs without needing a large current output. As specified, 896H-1CH-D1 relay models with 90Ω of coil resistance are used; hence 133mA of current will flow through the collector of each transistor. In order to turn on the transistor with a 5V signal, the base current has to be greater than

$$\frac{133mA}{100} = 1.33mA,$$

where 100 is the minimum h_{FE} value at 150mA. To ensure the transistor is in saturation mode when the GPIO is high, the required base current is roughly doubled to 2.66mA. Based on a 5V input and a required 2.66mA current, a 1.88kΩ resistor is placed before the base of each transistor [9].

The schematic in Fig. 15 shows how the original headlight features have not been altered. If the user chooses to turn the headlights off, to turn on low beam or to turn on high beam, they can still do so and the multifunction switch will override the auto system. It is also important to note that the auto high beam system will only function when the low beam headlight has turned on. This is designed to prevent users from unintentionally turning on the system or intentionally turning it on during the day.

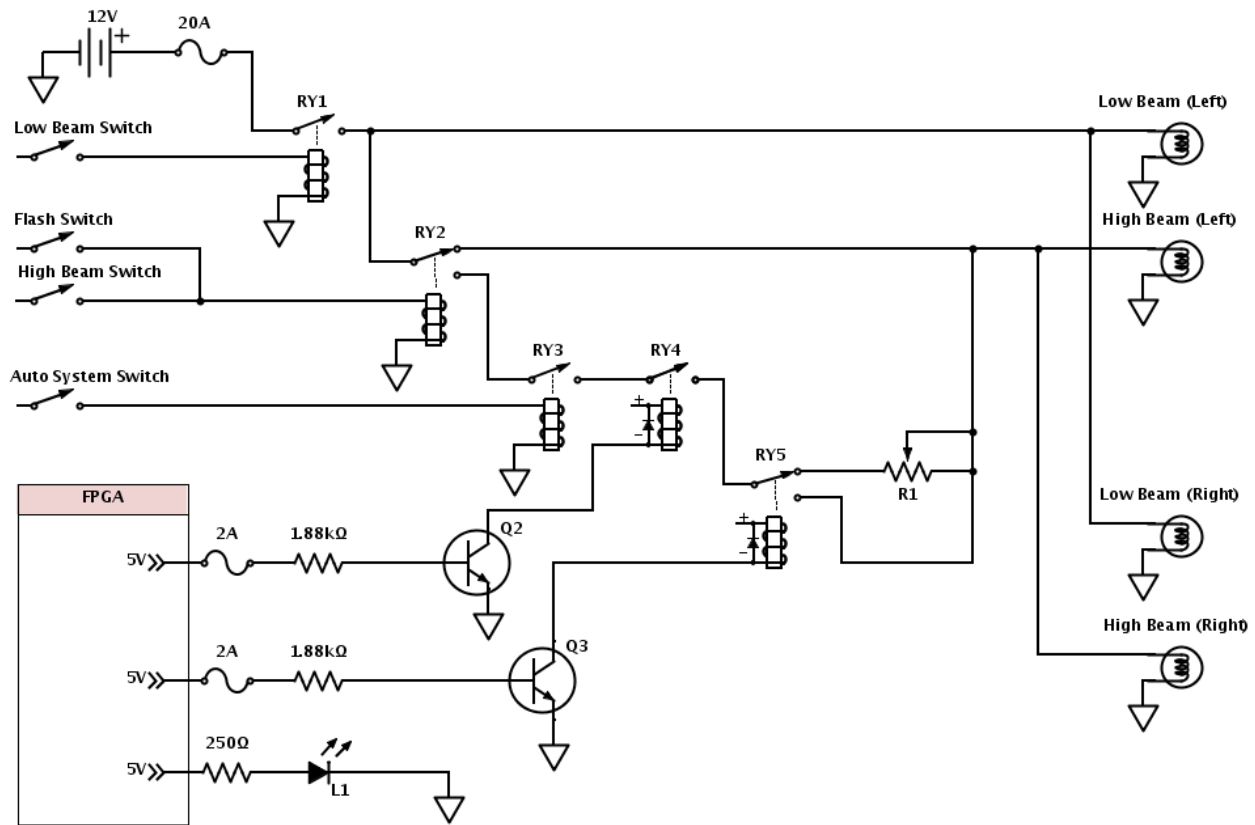


Figure 15: Electrical Circuit Board with Relay

3.8 Mechanical

Auto high beam system consists of 2 mechanical systems: a modified multifunction switch and a camera stand. Mechanisms of each system will be discussed in detail in the following subsections.

3.8.1 Switch

The basic features on a car's multifunction headlight switch are relatively consistent across all models and brands. In addition, installing the auto high beam system would not change nor hinder the original mechanism of the switch. To turn on the Auto High Beam System, a simple on/off auto button placed on the switch as shown in Fig. 15. This button should be placed outside the reach of the driver.

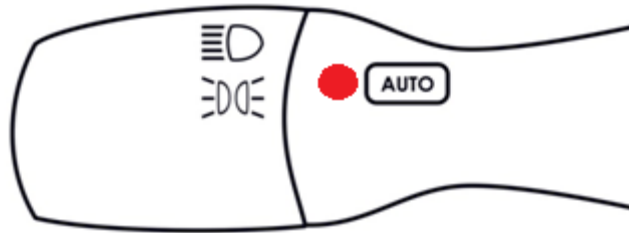


Figure 16: A Dip Switch with an Auto Button Attached

3.8.2 Camera Stand

The camera car stand will be simply composed of car mount and the camera, which will be securely mounted on to the dashboard. For demo purposes, a suction cup stand shown in Fig. 17 will be used. However, the actual product will be secured onto the dashboard with screws to ensure the car stand is mounted securely. The dimensions of the car stand are shown in Fig. 18. The values are in centimetres.



Figure 17: Car Mount Actual Model [10]

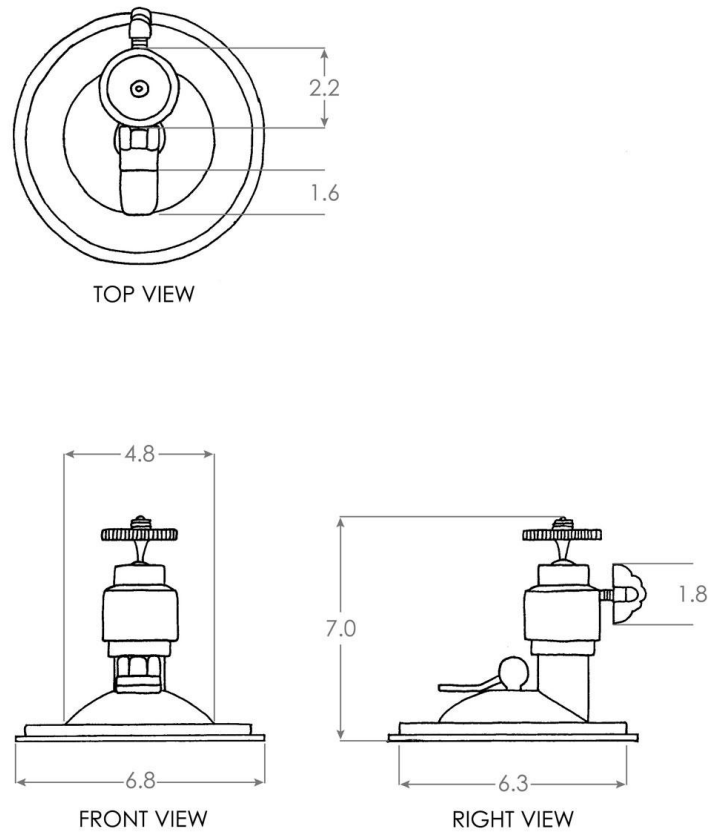


Figure 18: Car Mount Outline Dimension [10]

4. Software

4.1 Software System Overview

An input sequence of 24-bit true color video at a resolution of 320x240 is processed frame-by-frame to detect vehicles and their distances from the camera and decide the setting of the high beams. The algorithm uses the width of a car to determine its distance from the user; a car's apparent width is larger the closer it is to the camera. In order to measure the width, a vehicle must be identified by its headlights or tail-lights. A dynamic thresholding procedure is applied to each frame to identify the bright objects. Pairs of bright objects are grouped and matched as vehicle lights and the horizontal distance between their outside edges determines the width of the car. Since headlights emit more light than tail lights, it is necessary to adjust some of the specifics of the object detection criteria, as well as fine-tune the distance measurement due to the perceived increase in the radii of headlight objects. Thus, a distinction must be made between the two sources early on through color analysis in HSV space. The general procedures that follow will be the same for both light sources; however, the parameters will be customized to match their specific characteristics. Furthermore, an override feature is included that analyzes the video to determine if the surroundings are too bright for high beam use. The vehicle detection software will only execute if high beams are found to be appropriate for the particular situation; otherwise, the software will forego any further video processing and keep the high beams off. The software will be implemented in C and will run on the Nios II processor on the Altera DE2-70 development board. A high-level flowchart of the software system is shown in Fig. 19.

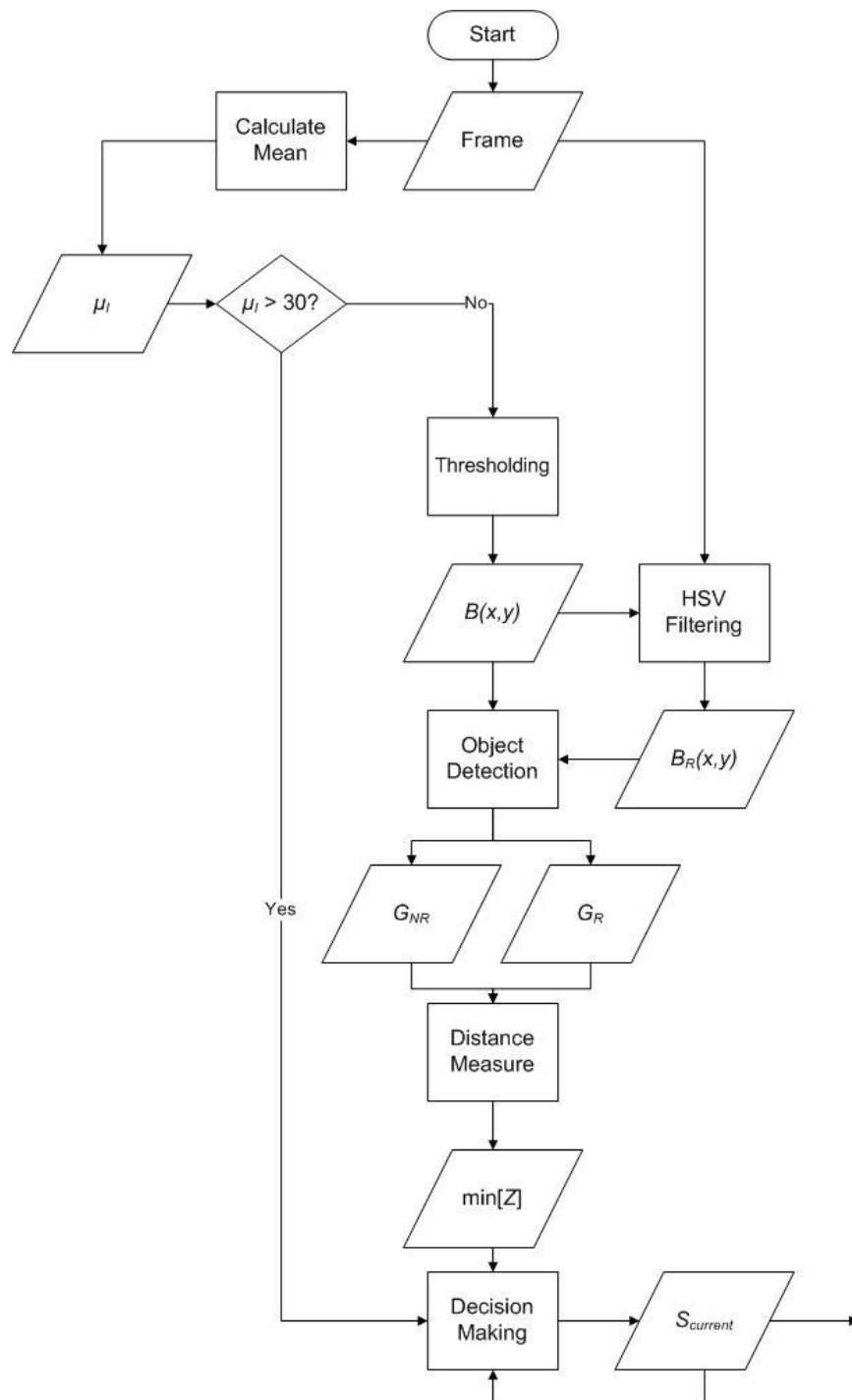


Figure 19: Overview of software system

4.2 Thresholding

The goal of the thresholding procedure is to produce a binary image that holds a “1” in regions containing bright objects and light sources, and “0” elsewhere. Since it is only necessary to look at image intensity to determine whether or not an object is bright, the input RGB frame must be transformed as follows

$$Y = 0.299R + 0.587G + 0.114B. \quad (1)$$

The result is an image $I(x,y)$ with 256 gray levels. The result of thresholding this image will be a binary map $B(x,y)$ such that $B(x,y) = 1$ for all $I(x,y) > T_{opt}$ and $B(x,y) = 0$ for all $I(x,y) \leq T_{opt}$. The next step is to determine the optimal threshold T_{opt} such that light sources will be properly distinguished from other bright objects, such as illuminated street markings and bright, reflected surfaces.

The method used to determine the optimal threshold is from [11]. First, a histogram of the gray values in the grayscale image $I(x,y)$ is formed and is normalized by N , the number of pixel in $I(x,y)$, to yield a probability distribution of gray levels of $I(x,y)$. That is,

$$P_i = \frac{f_i}{N}, \text{ where } P_i > 0, \sum_{i=0}^{255} P_i = 1, \quad (2)$$

where f_i is the number of pixels with gray level i and P_i is the probability of a pixel having a gray level i .

The goal of this procedure to iteratively separate the image into different classes of gray levels, denoted as $D_0 = \{0, 1, \dots, t_1\}, \dots, D_n = \{t_n+1, t_n+2, \dots, t_{n+1}\}, \dots, D_k = \{t_k+1, t_k+2, \dots, 255\}$, which would be obtained by applying a thresholds from a set of k thresholds $\mathbf{T} = \{t_1, \dots, t_w, \dots, t_k\}$. Then, the optimal threshold will be $T_{opt} = \max[\mathbf{T}] = t_k$. In order to optimize this separation among classes of gray levels in $I(x,y)$, it is useful to define some measures of separability. The between class variance is defined as

$$v_{BC}(\mathbf{T}) = \sum_{n=0}^k w_n (\mu_n - \mu_T)^2, \quad (3)$$

the within class variance is defined as

$$v_{WC}(\mathbf{T}) = \sum_{n=0}^k w_n \sigma_n^2, \quad (4)$$

and the total variance and overall mean of the pixels in $I(x,y)$ are

$$v_T = \sum_{i=0}^{255} (i - \mu_n)^2 P_i, \text{ and } \mu_T = \sum_{i=0}^{255} iP_i. \quad (5)$$

The cumulative probability density, represented as w_n , of a class D_n is given by

$$w_n = \sum_{i=t_n+1}^{t_{n+1}} P_i \quad (6)$$

and the mean and standard deviation, represented by μ_n and σ_n , respectively, of a gray level class D_n are defined as

$$\mu_n = \frac{\sum_{i=t_n+1}^{t_{n+1}} iP_i}{w_n}, \text{ and } \sigma_n^2 = \frac{\sum_{i=t_n+1}^{t_{n+1}} P_i (i - \mu_n)^2}{w_n} \quad (7)$$

where $t_0 = 0$. In order to optimally separate pixels into their respective gray level classes, the variance of each class σ_n would need to be minimized. A separability factor can be defined that takes this principle into account

$$SF = \frac{v_{BC}(\mathbf{T})}{v_T} = 1 - \frac{v_{WC}(\mathbf{T})}{v_T}. \quad (8)$$

The value of SF must range between $[0,1]$. As SF approaches 1, within class variance is minimized and all classes of gray levels in $I(x,y)$ are optimally separated. The method of achieving this is to iteratively determine which class has the largest in-class variance (v_{WC}) and then partition that class into two separate classes. The separability factor would then be re-computed to determine if a desired level of separation is achieved. This procedure is outlined in detail below.

Step 1: Compute probability distribution of gray levels in the image $I(x,y)$ along with the initial v_T and μ_T . The variable q will henceforth denote the number of thresholds in \mathbf{T} to separate gray values into $q+1$ classes. The set \mathbf{T} is currently empty since no thresholds have been established, thus $q = 0$.

Step 2: Consider an iteration of the algorithm where q thresholds are established, $\mathbf{T} = \{t_1, \dots, t_w, \dots, t_q\}$, thus yielding the $q+1$ classes of gray-levels $\{D_0, \dots, D_n, \dots, D_q\}$. Compute w_n , μ_n , and σ_n using (6) and (7) for each class D_n .

Step 3: Determine which class to partition further into two separate classes by determining which class has the largest v_{WC} .

Step 4: Given the class $D_p = \{t_{p+1}, t_{p+2}, \dots, t_{p+1}\}$ with maximal v_{WC} from Step 3, divide this class into two new classes D_{p0} and D_{p1} by calculating the optimal threshold $t_s^* \in D_p$. This optimal threshold is

determined by maximizing the between class variance v'_{BC} of D_{p0} and D_{p1} with respect to t_s , as shown below.

$$v'_{BC}(t_s^*) = \max_{t_p < t_s \leq t_{p+1}} v'_{BC}(t_s) \quad (9)$$

$$v'_{BC}(t_s) = w_{p0}(\mu_{p0} - \mu_p)^2 + w_{p1}(\mu_{p1} - \mu_p)^2, \quad (10)$$

$$w_{p0} = \sum_{i=t_p+1}^{t_s} P_i, \quad w_{p1} = \sum_{i=t_s+1}^{t_{p+1}} P_i \quad (11)$$

$$\mu_{p0} = \sum_{i=t_p+1}^{t_s} iP_i / w_{p0}, \quad \mu_{p1} = \sum_{i=t_s+1}^{t_{p+1}} iP_i / w_{p1}, \quad (12)$$

where w_p and μ_p are the class probability and class mean of D_p , respectively. Once all possible values of D_p are considered with the above calculations and a t_s^* is found, place it into set \mathbf{T} and replace class D_p with classes D_{p0} and D_{p1} . Thus, there are now $q+1$ thresholds in \mathbf{T} .

Step 5: Compute the separability factor using (8) and check to see if

$$SF \geq Th_{SF}. \quad (13)$$

If (13) is satisfied, go to Step 6. Otherwise, return to Step 2.

Step 6: Select the largest threshold value in \mathbf{T} as T_{opt} . Apply this threshold to $I(x,y)$ as previously mentioned to obtain a binary map $B(x,y)$ with pixels belonging to the brightest class of objects.

The value of Th_{SF} chosen for our system is 0.92. Anything lower was found to include other unwanted illuminated regions of the image, such as patches of asphalt that were exposed to headlights. Anything higher was found to be unnecessary for extracting vehicle light sources and also significantly increased the runtime of the procedure. The results are illustrated in Fig. 20.



Figure 20: Original frame (left) and binary map of bright objects (right)

4.3 HSV Filtering

Before the bright objects extracted by the thresholding procedure can be grouped together as vehicle light pairs, it is necessary to distinguish between tail lights and headlights. This is due to the size and shape differences between the two light sources, which need to be taken into account when applying the object detection constraints. For example, a headlight appears as a large circular blob whereas a tail light is smaller in height and width and has a distinguishable shape. Distinguishing between the two will also play an important role in the distance calculation.

In a study by O’Malley *et al.* [12], a database of 300 tail light images was analyzed to determine a set of parameters in HSV color space (RGB was found to be ineffective for this purpose) to detect red regions. The HSV color space can be thought of as a cone with hue (tint) as the angle parameter ($0^\circ - 360^\circ$), saturation (shade) as the radius ($0 - 1$), and value (tone) as the height ($0 - 1$). The distribution of tail light color values from the database in [12] along with their thresholding region mapped into HSV space is illustrated in Fig. 21.

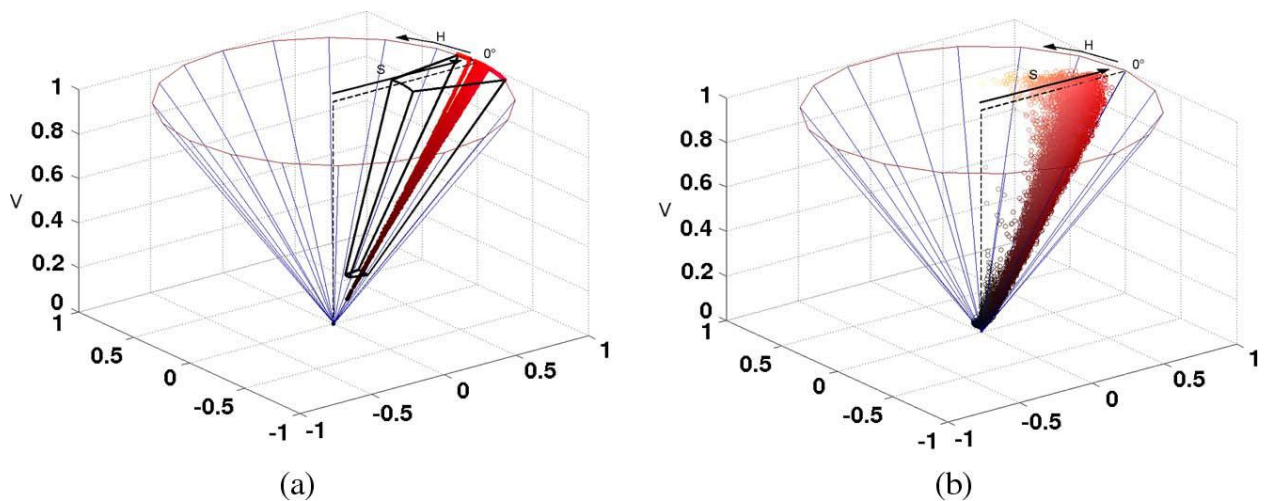


Figure 21: (a) Color threshold region (b) Distribution of tail light colors from database of 300 images [12]

The derived HSV filter threshold limits from [12] are summarized in Table 3.

Table 3: HSV red filter parameters

	Minimum	Maximum
Hue (<i>H</i>)	342°	9°
Saturation (<i>S</i>)	0.4645	1.0
Value (<i>V</i>)	0.2	1.0

A bitwise AND operation is performed between the binary mask $B(x,y)$ and the three individual channels of the original RGB frame. The HSV filter described above is applied to the resulting image. This essentially yields a second binary mask $B_R(x,y)$ such that $\{(x,y) \mid B_R(x,y) = 1\} \subseteq \{(x,y) \mid B(x,y) = 1\}$.

4.4 Object Detection

Connected components (with 8-connectivity) are extracted [13] from the binary map $B(x,y)$. It is important to note that tail lights may not always appear entirely red. Due to saturation in the image, they may often appear as a white center with a bright red surrounding as illustrated below.

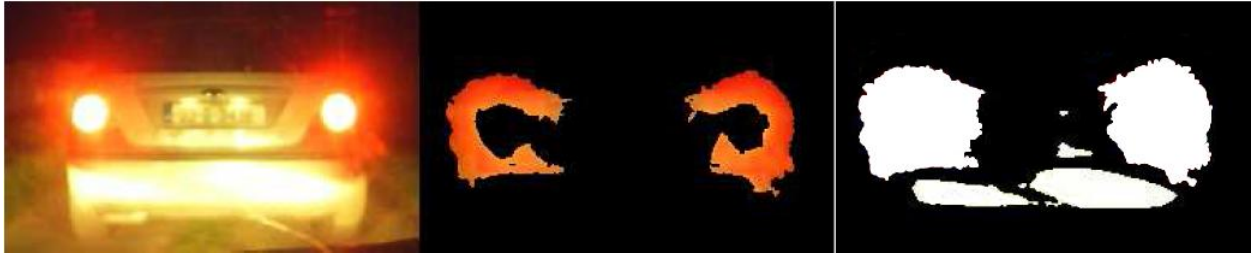


Figure 22: Original image (left) Binary mask $B(x,y)$ after thresholding (middle) [14] HSV filter result $B_R(x,y)$ (right) [14]

The indices of each connected component is compared with the indices $B_R(x,y)$. Due to the issue in Fig. 22, it is likely that only a subset of the indices of a connected component belonging to a tail light would be classified as “red”. Thus, if 30% of the indices (x,y) of that connected component also belong to the set $\{(x,y) \mid B(x,y) = 1\}$, then that connected component can be considered a tail light. Hence, red light sources and non-red light sources are divided into two separate sets of connected components. The object detection algorithm will operate on these two sets individually using the same procedure but slightly modified criteria, which will be discussed in detail. For the purposes of presenting the object detection algorithm, consider a general set of vehicle light candidates.

Vehicle light candidates are the set of connected components, $\mathbf{C} = \{C_1, \dots, C_N\}$, where N denotes the total number of connected components (N would generally vary between red and non-red light candidate sets). The bounding box of a connected component C_i is described by its bottom, top, left, and right coordinates, which will be denoted as $b(C_i)$, $t(C_i)$, $l(C_i)$, and $r(C_i)$, respectively. The width and height of this bounding box are denoted as $W(C_i)$ and $H(C_i)$, respectively and are defined as follows

$$W(C_i) = r(C_i) - l(C_i) \quad (14)$$

$$H(C_i) = b(C_i) - t(C_i) . \quad (15)$$

In this context, a height and width of 0 would mean that the box that bounds the connected component has identical top, bottom, left, and right coordinates, i.e., the connected component is a single pixel.

The horizontal distance between two connected components C_i and C_j is defined as

$$D_h(C_i, C_j) = \max[l(C_i), l(C_j)] - \min[r(C_i), r(C_j)]. \quad (16)$$

Similarly, the vertical distance is defined as,

$$D_v(C_i, C_j) = \max[t(C_i), t(C_j)] - \min[b(C_i), b(C_j)]. \quad (17)$$

Note that when two objects C_i and C_j overlap in the horizontal or vertical directions, D_h or D_v will yield negative values, respectively. Overlapping in the vertical direction will be useful measure since we expect matching vehicle lights to lie across the same y coordinate. Thus, a measure that quantifies vertical overlap [11] is defined as follows

$$P_v(C_i, C_j) = \frac{-D_v(C_i, C_j)}{\min[H(C_i), H(C_j)]}. \quad (18)$$

Note that vertically overlapping objects will have a positive P_v . Furthermore, $P_v \leq 1$, where $P_v = 1$ denotes maximum vertical overlap.

Our goal is to determine which connected components of the set \mathbf{C} belong to vehicles and cluster them accordingly. The set $\mathbf{G} = \{G_1, \dots, G_M\}$ corresponds to groups of vehicle lights for M vehicles. G_j is composed of a group of connected components that belongs to the j th vehicle, $G_j = \{C_i : i \in 1, \dots, N\}$.

To reduce noise, all connected components with $H(C_i) = 0$ or $W(C_i) = 0$ are removed. A comparison among remaining connected components is done such that the operation in (17) yields an upper triangular matrix D_v . Thus, the i th row and j th column of this matrix corresponds to the vertical distance between C_i and C_j . This matrix is then used to compute (18), yielding another upper triangular matrix P_v . Object pairs are considered to be light candidates if there is vertical overlap between them and are identified by finding entries where $P_v > 0$.

Vertically overlapping objects are paired together if the following two conditions are satisfied: they must be horizontally adjacent and they must similar in size. That is, the smaller object must have at least 60% of the number pixels of the larger object (concept taken from [14]). As oncoming cars approach, their right headlight may appear to be larger than the left, as it is closer to the user. Thus, if this constraint is set higher than 60%, the disparity in the apparent sizes of an oncoming car's low beam may prevent detection of the vehicle at close distances. If the pair of objects C_i and C_j satisfy these criteria, cluster them into G_k .

Once this comparison is done for all vehicle light candidate pairs, determine the height and width of each cluster. Clusters of pairs that are too close, or too far from each other are removed. This is determined by the aspect ratio of the box that bounds the pair

$$ASPECT_{lower} \leq \frac{W(G_i)}{H(G_i)} \leq ASPECT_{upper} \quad (19)$$

(concept taken from [14]). The upper thresholds for the aspect ratio will vary between tail light candidates and headlight candidates. Due to differences in shape and overall brightness, tail lights will use $ASPECT_{upper} = 13$ and headlights will use $ASPECT_{upper} = 7$. The value of $ASPECT_{lower} = 2$ for both cases.

The next step is to remove clusters whose bounding boxes overlap with the bounding boxes of two or more other clusters. This situation is illustrated in Fig. 23. An incorrect cluster, with a bounding box shown as a red rectangle, has been formed between the headlights of two different vehicles. Since its bounding box overlaps with both of the two green bounding boxes of the correct pairings, it must be removed.

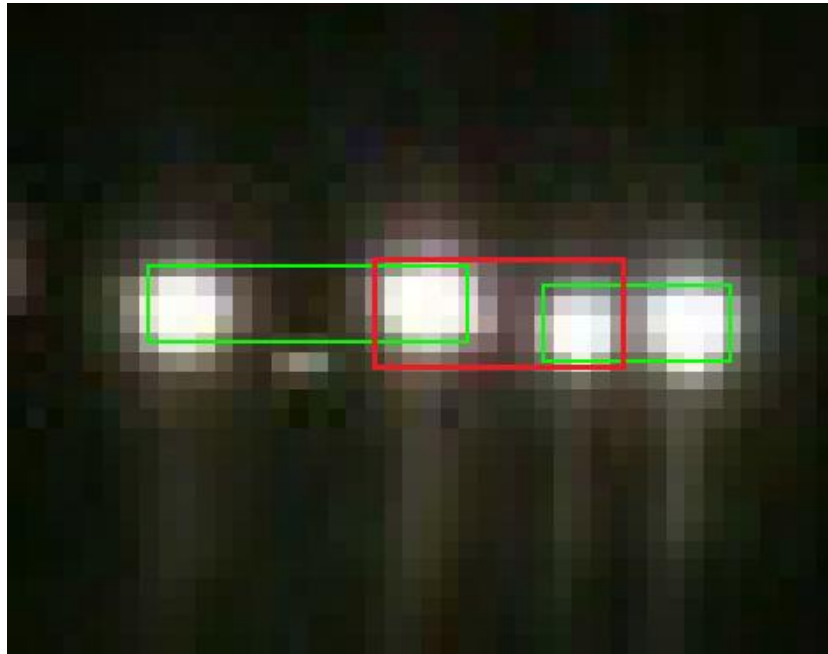


Figure 23: Incorrect clustering of headlights shown by the red rectangle

In order to detect such an overlap, the horizontal and vertical signed distances between bounding boxes of clusters are computed very similar to (16) and (17). Hence, if $D_h(G_i, G_j) < 0$ and $D_v(G_i, G_j) < 0$, concurrently, for any two j such that $j \neq i$, remove G_i .

The final step is to remove any clusters whose bounding box is within the left and right bounds of any other cluster. This is done to remove clusters that consist of minor feature pairs belonging to vehicles that have already been identified by their tail lights or headlights, as illustrated below.



Figure 24: Incorrect clustering of minor features shown by the red rectangle

The situation in Fig. 24 is the result of illumination around the license plate causing two distinct connected components to form after thresholding. Since the vehicle has already been identified by its tail lights, this second pairing is unnecessary and is removed. In general, if $l(G_i) > l(G_j)$ and $r(G_i) < r(G_j)$, remove G_i .

4.5 Distance Estimation

The distances from user to a vehicle measured based on the vehicle's width as given by the width of the box that bounds headlight/tail light pairs. To accomplish this, the system utilizes the equation from [11],

$$Z = k_{hor} \cdot f \cdot (W/w), \quad (20)$$

where f is the camera's focal length in millimeters, w is the width of the vehicle in the frame given by $W(G_i)$, and W is the estimated width of a typical car, which according to the author of [11] is 1.8 meters in Taiwan. The constant k_{hor} is a scaling factor that converts pixels to millimeters. It can be reliably determined through a calibration procedure described below.

Step 1: Measure a distance Z in meters from an object, i.e., a vehicle, and place the camera level at that distance from the object.

Step 2: Measure the real world width of the object as it appears from the camera’s perspective.

Step 3: In the recorded video, measure the width of the object in the frame by pixels.

Step 4: Solve for k_{hor} .

Step 5: To obtain an accurate measure for k_{hor} , repeat this process several more times with varying distances Z while keeping the resolution of the video the same.

It is important to note that due to the differences of sizes between headlights and tail lights, the width of box that bounds them needs to be modified before it is used in (20). Based on our observations, the tail light bounding boxes need to be compensated by 1 pixel on each end. Therefore, $w = W(G_i) + 2$ is used in (20) when G_i is a cluster of tail lights. Since the beam of a headlight is stronger, 1 pixels from either end of the bounding box is subtracted, i.e., $w = W(G_i) - 2$ when G_i is a cluster of headlights.

4.6 Decision-Making and Overriding

The software will output a decision based on the smallest distance in the frame measured with (20). In order to prevent sporadic behavior, the system must employ hysteresis when making its decisions. Thus, it is necessary to save the system’s decision from the previous frame. The possible decisions of the software are summarized in Table 4.

Table 4: Possible outputs of the decision- making algorithm

Decision	Effect
$S_0 = "00"$	High beams ON at 100% of max intensity
$S_1 = "01"$	High beams ON at 50% of max intensity
$S_2 = "10"$	High beams OFF

Given the previous decision and the value of $\min[Z]$ from the current frame, the new decision is made according to the flowchart in Fig. 25. The decision is ultimately represented as a binary 2-bit output to the GPIO of the DE2-70 board.

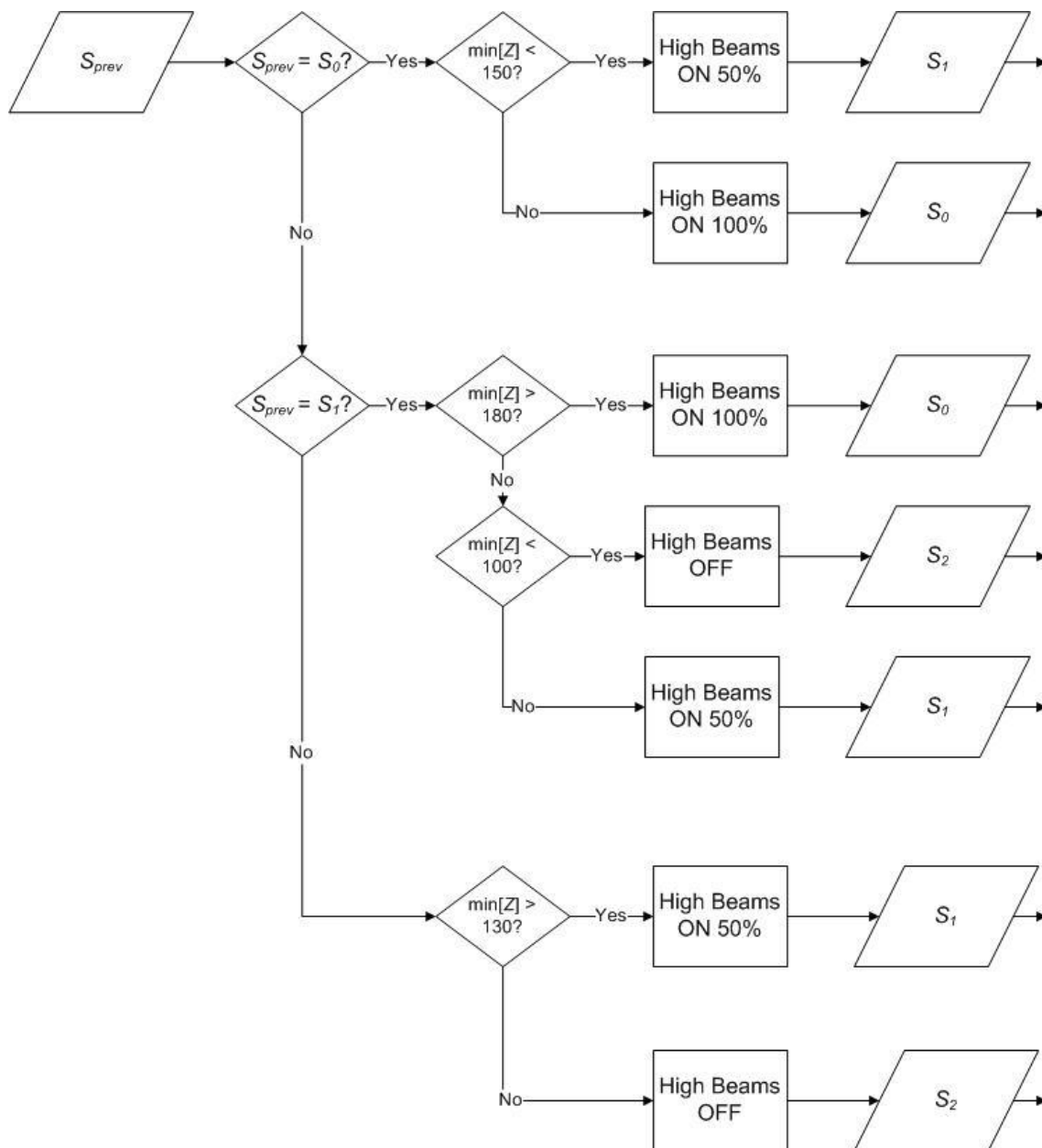


Figure 25: Flowchart of the decision-making procedure

In addition, our software includes an override feature that will keep the high beams off, i.e., output decision S_2 , in times when the surroundings are too bright to be appropriate for high beam use. This is done simply by calculating the mean of the intensity of the frame $I(x,y)$. Based on our experimental evidence from driving sequences taken during daytime and night time, we have determined that when

the mean is above 30, the software skips all of the vehicle detection procedures and simply outputs a decision to turn off the high beams. That is,

$$\text{output } S_2 \text{ when } \mu_I = \frac{1}{320 \times 240} \sum_{y=1}^{240} \sum_{x=1}^{320} I(x, y) > 30. \quad (21)$$

5. Test Plan

5.1 Unit Test

It will be necessary to test each individual component, to ensure that each one functions properly. The parts can be integrated with one another once the following tests are conducted and the expected results are achieved.

5.1.1 Automobile Relay

The relays are a crucial component in our hardware system. They allow the car components to be powered directly by the car battery but triggered by a very low signal voltage. Fig. 26 shows the wiring diagram of a SPDT car relay.

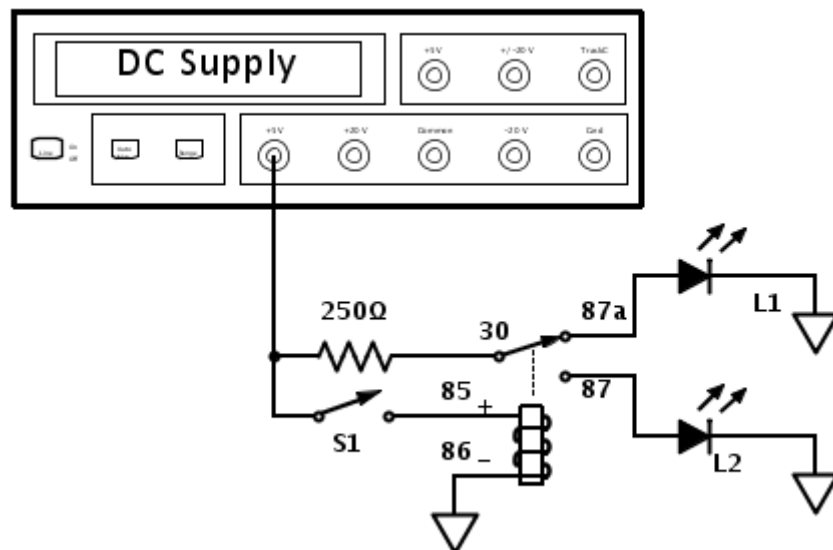


Figure 26: Relay Component Test Setup

- Procedure:
1. Locate the pin number beside each prong on the SPDT car relay.
 2. Strip and wrap a wire securely onto each prong.
 3. On a breadboard, build the circuit shown in Fig. 26 above without connecting to the power supply.
 4. Attach the DC power supply to a DMM.
 5. Adjust the voltage on the power supply until the DMM reads 5V.
 6. Adjust the current output on the power supply to the minimum without turning off the voltage supply.
 7. Now connect the power supply to the circuit. Flipping the SPST switch S1, the relay will switch between LED L1 and L2 turning them on and off accordingly. The 250Ω resistor will ensure that the 5V power supply does not pass more than 20mA of current through the LEDs.

5.1.2 Battery

Before connecting the high beam system to the car battery, the condition of the battery should be checked. For the proof-of-concept model, a second battery will be used to provide the necessary power for the hardware system including the headlights.

- Procedure:
1. Check for any rust or corrosion on the battery before connecting anything to it.
 2. Verify the charge on the battery by connecting a DMM across the negative and positive terminals of the battery. The DMM should read approximately 11-12V.

5.1.3 Headlights

The headlights can be easily tested by connecting a power supply to the prongs behind the headlight. The headlight we are testing has 3 prongs. The middle prong is ground while the other two are either low beam or high beam.

- Procedure:
1. Connect the DMM to the power supply.
 2. Adjust the voltage on the power supply until the DMM reads approximately 12V.
 3. Measure the current from the power supply. Make sure the current is within 4A.
 4. Turn off the power supply.
 5. Connect the ground wire to the centre prong of the headlight.
 6. Connect the power wire to one of the other prongs.
 7. Turn on the power supply, the headlight should turn on.
 8. Remove the power wire and connect it to the other prong. Verify that different headlight brightness turns on.

5.1.4 Multifunction Switch

The car multifunction used in this model was removed from an old Honda civic. It contains the OFF, low beam, high beam, and flash high beam functions that all car models will have. The car's multifunction switch is used to model how our system will work with the original car's features.

5.1.5 Circuit

The main hardware circuit consists of the car's multifunction switch, the GPIO output from the processor, the relay circuit and the car headlights. To verify the functionality of the circuit, the test will be split into two sections.

User Override Command

The user should be able to override the automated system whenever it is desired. When the driver needs to turn on low beam, high beam, or flash, our modified headlight circuit should have no effect on the original functions of the car.

- Procedure:
1. When the user turns the multifunction switch to the off position, nothing should turn the headlights on:
 - a) FPGA board should either be off or when the board does output a signal, the circuit should not respond;
 - b) Turning on the auto button should have no effect on the system.
 2. When the user switches to low beam, only the low beam will turn on. Step 1a and 1b can also be used to check that the low beam lights remain unchanged.
 3. When the user switches to high beam, the headlights will always be on high beam. The user should not be able to turn on the auto system. The outputs of the processor should also have no effect on the brightness of the high beam headlights. The high beam headlights should be on at full brightness.

Auto System

The model auto system should be tested inside a real car. Another car can be parked facing the car under test. The auto system can be turned on by the driver only after the low beam has been turned on. Once the user turns on the auto system, the high beam headlight should be on full brightness until the system detects other cars in traffic.

- Procedure:
1. Turn on low beam headlight.
 2. Turn on camera and FPGA board.
 3. Switch the auto button to the on position. The high beam headlights should turn on at full intensity.
 4. Once the auto system is set up, we will turn the headlights of the other car on. The system should process this and dim the high beam headlights.

5.1.7 Video Camera

To be able to accurately track vehicles that are up to 180m away, a reasonable resolution video is required. Since the software algorithm identifies vehicles based on matching headlight pairs, it is ideal to minimize saturation for light sources. This can be done by manipulating the shutter speed and aperture number of the camera. The following procedure should be performed to properly calibrate the camera so that it can provide a good input to the software system.

- Procedure:
1. Turn on the camera.
 2. Monitor a sample vehicle that is 200m away with headlights.
 3. Verify that two distinct light sources from the sample vehicle are recognizable. If the camera captures the headlights as a single blob of light, the camera settings need to be modified for adequate performance.
 4. Repeat the step 3 with the sample vehicle's high beams on. Adjust the shutter speed and aperture number even further to satisfy the aforementioned criteria.

5.1.8 Camera Car Stand

The camera stand should be firmly placed on top of the dashboard behind the windshield. It must be stable enough to hold the camera in a fixed position and durable enough to withstand low temperatures and any bumps and turns on the road

- Procedure:
1. Place the sticky rubber stand on top of the dashboard.
 2. Attach a mass that is approximately twice the weight of the camera to check if it can hold it in place without any tilt.
 3. Keeping the same mass attached, test the adhesive strength of the stand in low temperature.

5.1.9 Altera DE2-70 Board

Before any video processing is done on the DE2-70, we must ensure that live video can be streamed into the board's onchip memory, where it can be modified by the Nios II processor.

- Procedure:
1. Design a basic system with Altera's SOPC builder as described in Section 3.3, except instead of GPIO, the output frames will be sent to the VGA controller.
 2. Write a simple program for the Nios II processor to modify the frame buffered in memory, such that the blue and green color components of a square block in the

- center of the frame are set to “FF”, while the red component is unadjusted.
3. Connected a camera to the Video In port and a monitor to the VGA port.
 4. Turn the camera on and stream live video to the board after the board has been programmed.
 5. Verify that the picture is displayed on the monitor with a discolored (red) block at the center. Verify that the picture is updated in real-time on the monitor by filming something in motion

5.1.10 Software System

The role of the software system is to analyze each frame of the video input and make a decision to adjust the high beams. It is most conveniently tested on a PC to run C code integrated into a MATLAB via a MEX wrapper.

- Procedure:
1. Use MATLAB’s “VideoReader” function to decode a video into raw frames.
 2. For each frame, pass the frame onto the algorithm in C using a MEX wrapper.
 3. MEX wrapper will transfer outputs back into MATLAB upon completion.
 4. Using MATLAB’s plotting tools, display all data visually to get a better understanding of the functionality. Bounding boxes can be drawn as green rectangles on top of the frame after it is displayed and the distance measurement can be written in red text underneath the bounding box.
 5. Display any other useful outputs, such as the current decision, on MATLAB’s terminal.
 6. Test on a variety of different input sequences and ensure that everything functions according to the specifications.

5.2 Integration and Requirements Test

Integration tests will be carried out in two different steps. The first step involves using a series of synthetic tests where the outcomes are very predictable. The first of these tests will be to ensure that system detects vehicles at the required distances and adjusts the high beams accordingly. In this step, pre-recorded videos where all distances are measured beforehand will be utilized as the video inputs.

- Procedure:
1. Record a calibrated video of a car stationed 200m away from the camera with its headlights on. The car will move slowly towards the camera and will eventually drive past it. Markers will be placed at the 200m, 150m and 100m points for us to verify the accuracy of our system.
 2. Input the video into the fully integrated system and set the system to auto mode. Verify that the headlights initially have the high beams on with 100% intensity.
 3. As the car in the video approaches, verify that the high beams dim to 50% of their maximum intensity when the car is 150m away and turn off completely when the car is 100m away. As the car exits the

4. Repeat the same test with a car moving forward away from the camera to ensure that the same results are achieved through tail light detection.

Next, the override feature of the software system will be tested to ensure that high beams remain off during times where the environment is too bright for their use.

- Procedure:
1. Feed a live video input taken during daytime.
 2. Set the system to auto mode. Verify that the high beam headlights immediately turn off.
 3. Cover the lens completely to ensure a pitch black input video. Verify that the high beam headlights immediately turn on to 100% of their maximum intensity.

The performance of the system is quantified by the time it takes for the input frame to enter the FPGA to the time the high beams adjust their setting. This is virtually impossible to measure precisely, given our limited resource. However, a good estimate can be made given the assumption that the largest delay will be caused by the software algorithm executed by the NIOS II processor inside the DE2-70 board. Thus, the following procedure can give a reasonable measure of performance.

- Procedure:
1. Attach a probe to the “Start of Packet” signal from the Video In Decoder component on the DE2-70 board using Altera’s Signal Tap II Logic Analyzer.
 2. Attach a second probe to an LED output.
 3. Add a slight modification to the software to turn on an LED after a frame is finished processing.
 4. Set the Signal Tap II Logic Analyzer to begin monitoring on the rising edge of the “Start of Packet” signal from the Video-In Decoder.
 5. Feed any input video into the system.
 6. Read the time when monitoring has started and the time when the LED’s signal went high. Verify the time between these two events is 30ms or less.

The second step involves using the system as intended in real-world situations and making qualitative observations on its performance and overall functionality. These tests can be unlimited in scope and are an attempt to imitate regular use of the device as if it were the retail product.

- Procedure:
1. Place the camera on its mechanical stand on the windshield, behind the support of the rear-view mirror.
 2. Feed the live video input from the camera to the FPGA.
 3. The system will not be fully integrated into the car we are testing with. Instead, the FPGA board and the rest of the hardware circuitry will be placed on top of the dashboard. Headlights will be secured on top of the car’s front hood. Extension wires will be used to connect the model headlight to the hardware system.
 4. Drive down a highway at night and have a second group member observe the

actions of the high beam.

5.4 Failure Test

The tests described in this section will investigate some of the foreseeable, unintended ways a user could use our system. Some solutions have already been implemented to prevent the expected error. Other human errors may not be resolvable at the moment due to the given time and resource.

Situation 1: The user tries turning on the Automated High Beam System during the day or when high beams are not needed.

Solution: The hardware circuit is designed so that only after the low beam has been turned on can the auto system be activated. The program will also analyze the live video and if it determines that the environment is bright enough, it will turn off the high beam.

Situation 2: User forgets to turn on the camera or the processor.

Solution: The current system will simply not respond to the changes in traffic because the default output from the processor is going to be to turn the high beam on to maximum intensity. If possible, we will also add an LED by the auto button. This LED will provide feedback from the processor after it has checked that all peripherals are functioning correctly.

Situation 3: The user accidentally turns off the auto button while driving

Solution: Since we specified that the Automated High Beam System can only be turned on when low beam is on, if the user turns off the automated system, the low beam headlights will still be on to light the road. The button will also be placed closer towards the base of the car's headlight multifunction switch. This way, the button will be out of reach from the driver unless the driver intends on turning the system on/off

Situation 4: During poor weather conditions, precipitation on the windshield may affect the quality of the input video.

Solution: Warning labels will be placed on the product to remind drivers to use caution when driving under poor weather conditions. Depending on the situation, the user may have to keep the high beam headlights always on/off. We will leave that up to the driver's own discretion.

5.5 Environmental Considerations

Most of the components used to build this proof of concept model are from recycled material that can be reused again in future projects. Major hardware components such as the camcorder, DE2-70 FPGA board, car battery will be returned to the original owners. Old car components such as the car relays, the headlights and the multifunction switch can be returned to the junkyard where they were from. If the Engineering Science Student Society decides to keep those parts for future 440 projects, then these components can be donated to them as well. Many electronics components can also be donated to the ESSS.

6. Conclusion

This document provides a detailed design specification of Lumos Technology's Automated High Beam System. It describes the purpose of each hardware components and the theory behind the video-processing algorithm used to implement system. This specification also discusses the choices we made in the design to meet the original functional requirements. In addition to the proposed solution, various test plans are provided to ensure that all functions of the system meeting our original design requirements. The engineers at Lumos Technology expect that proof-of-concept model will be ready by December 17th 2012.

7. References

- [1] Canon.(2012 November).Canon VIXIA HF M40 HD camcorder figure and specification [on-line]. Available:
http://www.usa.canon.com/cusa/consumer/products/camcorders/consumer_camcorders/vixia_hf_m40#Specifications
- [2] Altera.(2012 November).DE2-70 Development and Education Board figure and information [on-line]. Available: <http://www.altera.com/education/univ/materials/boards/de2-70/unv-de2-70-board.html>
- [3] Altera.(2010 July). Basic Computer System for the Altera DE2-70 board [on-line]. Available: ftp://ftp.altera.com/up/pub/Altera_Material/9.1/Computer_Systems/DE2-70/DE2-70_Basic_Computer.pdf
- [4] Terasic.(2012 November). Altera DE2-70 Board figure [on-line] Available:
<http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=53&No=226&PartNo=3>
- [5] Altera.(2010 July). Video IP Cores for Altera DE series Boards [on-line]. Available:
ftp://ftp.altera.com/up/pub/Altera_Material/10.1/University_Program_IP_Cores/Audio_Video/Video.pdf
- [6] 896H-1CH-D1 Relay Picture, [on-line] Available: <http://ca.mouser.com/ProductDetail/Song-Chuan/896H-1CH-D1-12VDC/?qs=sGAEpiMZZMvOgUnvsteuZ25VEb9%252b/Cy5>
- [7] 896H-1CH-D1 Relay Specification, [on-line] Available: <http://www.mouser.com/ds/1/378/896h-13415.pdf>
- [8] H13 Bulb Picture, [on-line] Available: <http://www.polyperformance.com/rally/Hella-OE-Replacement-Bulbs-p-27683.html>
- [9] Electus Distribution, (2001), Relay Driving Basics [on-line] Available:
http://www.jaycar.com.au/images_uploaded/relaydrv.pdf

- [10] Car DVR Suction Mount, [on-line]Available: http://www.ebay.ca/itm/Mini-Camera-Car-DVR-Suction-Mount-Holder-for-Car-Window-Camera-Camcorder-DV-/261121635920?pt=LH_DefaultDomain_0&hash=item3ccc0ff650
- [11] Y. L. Chen, "Nighttime Vehicle Light Detection on a Moving Vehicle using Image Segmentation and Analysis Techniques," *WSEAS Trans. Computers*, vol. 8, no. 3, pp. 506-515, Mar. 2009.
- [12] R. O'Malley, E. Jones, M. Glavin, "Rear-Lamp Vehicle Detection and Tracking in Low-Exposure Color Video for Night Conditions," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, pp. 453-462, June 2010.
- [13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 2/e*, New Jersey: Prentice Hall, 2002.
- [14] R. O'Malley, M. Glavin, E. Jones, "Vehicle Detection at Night Based on Tail-Light Detection", *1st International Symposium on Vehicular Computing Systems*, Trinity College Dublin, July 2008.