Software Requirements Specification (SRS)

Adaptive Driving Beam (ADB) System

Authors: Anthony Ghaith, Matthew Vazquez, Almostafa Aalabdulrasul, Seth Neubauer, Naod Ghebredngl

Customer: Mr. Jacob Rhodes, Program Manager - Electronics, Flex-N-Gate

Instructor: Dr. Betty H.C. Cheng

1 Introduction

Adaptive Driving Beam (ADB) System is used to help improve driving safety and performance when driving at night or during weather conditions where visibility isn't clear. This document describes the requirements, functionality, and use cases of this system. It begins with background information of the system, followed by the requirements. The requirements are then modeled in diagrams. Finally, a demonstration of the system shows the intended scenarios.

1.1 *Purpose*

The purpose of this document is to clearly define the requirements for developing the ADB system. This document covers all the specifications for the customer, as well as allowing for the development team to understand the requirements of the system before beginning construction.

1.2 *Scope*

ADB is an embedded system in a vehicle that automatically adjusts the headlight beams to maximize visibility for the driver in many situations. To do this, the ADB system utilizes the vehicle's built in vehicle and environmental detection measures to dynamically change the state of the beam. The ADB system is activated and deactivated by the driver with a button. When the ADB system is active, it analyzes the sensor data and adjusts the beam accordingly. If there are no other vehicles in front of the vehicle and weather conditions are clear, then the beam is set to maximum illumination. If there are other vehicles present or heavy weather conditions, then the beam is set to a lower illumination. The ADB system also looks at the road ahead and points the beam in the direction that the vehicle is traveling in. It accounts for changes in road curve as well as elevation to point the beam in the direction of travel.

1.3 *Definitions, acronyms, and abbreviations*

- **Adaptive Driving Beam (ADB):** The ADB is the entire system described throughout the document. Dynamically adjusts the vehicle's headlight distribution to maximize road visibility while preventing glare for other road users [6].
- **Environmental Detection Subsystem (EDS):** The EDS utilizes a combination of cameras and sensors to detect other vehicles, road signs, and environmental conditions such as fog or rain [6].
- **Vehicle Positioning Subsystem (VPS):** The VPS is the Subsystem to determine the vehicle's position relative to the road and other vehicles, helps adjust the beam's direction during curves or when ascending/descending slopes [6].
- **Human-Machine Interface Subsystem (HMI):** Provides feedback to the driver regarding the current status of the ADB system. Displays information such as when the system is active, what adjustments are being made, and any warnings if manual intervention is required [6].
- **Cybersecurity Subsystem (CSS):** Monitors and protects the ADB system from unauthorized access and cyber threats. Ensures secure communication between the ADB components, implements authentication protocols for system access and updates, and provides real-time threat detection and mitigation to prevent potential security breaches [6].
- **Beam Control Subsystem (BCS):** Responsible for dynamically adjusting the shape and direction of the headlight beams based on inputs from the EDS. Ensures that high beams are dimmed where they could affect drivers, while maintaining maximum illumination in other areas [6].
- **Vehicle Infotainment System (VIS):** The VIS is a display typically mounted to the right of the steering wheel.
- **On-Board Diagnostics II (OBD-II) port:** The ODB-II port is a self diagnostic system built into a vehicle. It’s a standardized protocol that allows extraction of diagnostic trouble codes and read time data [7].
- **Global Positioning System (GPS):** The GPS is a satellite based radio navigation system. It can pinpoint three dimensional position to meter level accuracy and time to the 10 nanosecond level, worldwide [8].
- **Light-Emitting Diode (LED):** LED, in electronics, a semiconductor device that emits infrared or visible light when charged with an electric current [9].

1.4 *Organization*

The rest of the SRS document contains functions our product has, expectations for our user, constraints of the system, assumptions and dependencies of the system, requirements our system must follow to satisfy our customer, models to showcase the system, and finally a prototype to demonstrate our systems' functionality. This will be organized into six sections, with the first being an introduction. The second section consists of an overall description of the system. The third section contains specific system requirements per our customer. The fourth section includes all the models. The fifth section contains both a link to our prototype, and sample scenarios being described and shown. Finally, the sixth section includes all the references that were needed in order to complete this document.

2 Overall Description

The following section will describe how the ADB system functions. Section 2.1 contains the context of the product, understanding the constraints on the entire system whether that's the interface (hardware, software, user, etc…) or adaptations to the system. Section 2.2, summarizes the main functions that the software will use to perform adjusting the headlights of the driver's vehicle. Section 2.3, will describe the average user expected to use the system. Section 2.4 will explain into more detail about the constraints on the system. Section 2.5 will explain the assumptions and dependencies required on the system. Section 2.6 will talk about requirements that are determined to be beyond the scope of the current project and may be addressed in future versions/releases.

2.1 *Product Perspective*

The ADB system provides visual clearance when driving and improves safety to the driver and to the people around them when operating a motor vehicle. Although this product is mainly software based, the system still requires components that are built into the car. The system itself is composed of smaller subsystems that work together to help provide the driver with maximum visibility when driving.

The ADB system will be activated/deactivated with an on/off switch button located to the left of the steering wheel. When the user activates the ADB system it will begin automating the headlights system, when they turn off the system it will revert to the driver manually adjusting the headlights themselves.

The vehicle must have a camera and sensors for detecting other vehicles and environment conditions. In addition to these hardware constraints, the headlight units must be capable of dynamic adjustments such as matrix LED technology, digital micro-mirror devices, and moving lenses [6].

As for software constraints, the ADB system must have software to allow communication between the other subsystems (BCS, EDS, CPS, HMI, CSS). In addition, there must be software that will help connect the ADB system to the hardware to maintain real time calculations, allowing for maximum visibility for the driver at all times.

With the system primarily being software related, memory constraints become an issue, the system embedded into the vehicle must have enough memory to maintain the safety critical system to allow for constant safety of the driver and the external actors within the environment that interact with the ADB system.

The following diagram illustrates how information flows through the ADB subsystem. External entities, such as the driver, the road environment, and vehicle platform sensors, provide inputs that activate the system and supply environmental and positional data. These inputs are processed by the EDS and VPD, which forward information to the BCS. BCS sets the optimal headlight pattern, sending commands to the beam itself. A dedicated CSS monitors internal communication and reports anomalies. Finally, the HMI communicates beam status, warnings, and fallback events back to the driver.

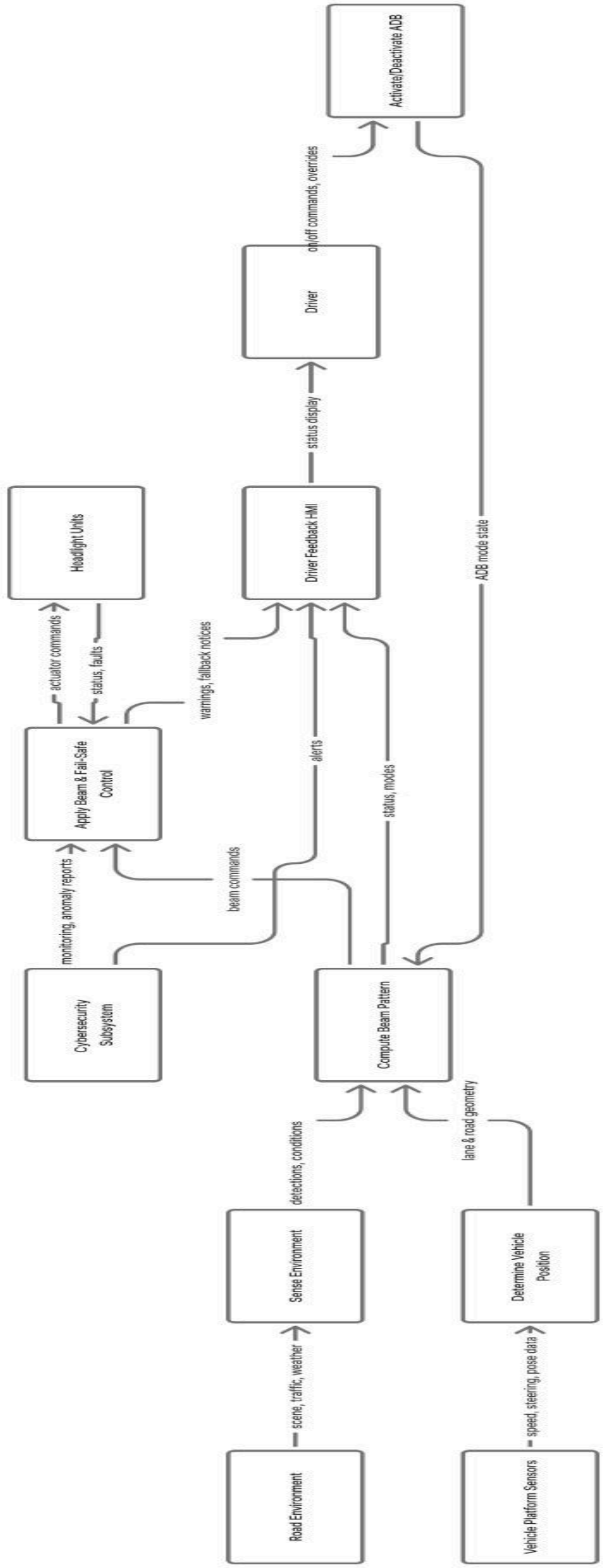


Figure 0: Product Perspective Flow Chart

2.2 Product Functions

The ADB system is designed to optimize nighttime driving safety by automatically adjusting the vehicle's headlight distribution based on real-time environmental and road conditions. The system continuously monitors the vehicle's surroundings to maximize forward visibility while preventing glare for other drivers. This section describes the high-level functions performed by each major subsystem of the ADB system. [6]

Beam Control Subsystem

The BCS is responsible for dynamically shaping and directing the vehicle's headlight beams. Using data from the EDS and VPS, it determines where high beams may be safely projected and where they must be dimmed to avoid causing glare. This subsystem ensures that road illumination for the driver is maximized, while maintaining as little glare as possible for other drivers.

Environmental Detection Subsystem

The EDS utilizes front-facing cameras and other onboard sensors to detect vehicles, road signs, lighting conditions, and adverse weather such as fog or heavy precipitation. It provides continuous, real-time assessments of the environment to support safe beam adjustments.

Vehicle Positioning Subsystem

The VPS determines the vehicle's orientation relative to the road by using sensor data, vehicle dynamics, and mapping inputs. This includes detecting curves, hills, slopes, and lane geometry. By providing this information to the BCS, the system ensures that headlight beams are directed appropriately even when navigating complex or uneven roadways.

Human–Machine Interface Subsystem

The HMI communicates system status and behavioral changes to the driver. It displays indicators for ADB activation, beam pattern adjustments, fallback to low-beam mode, and any alerts requiring driver awareness or intervention. The goal of the HMI is to present clear, accessible, and timely feedback so the driver remains informed about the system's operation.

Cybersecurity Subsystem

The CSS safeguards the ADB system against unauthorized access, malicious commands, and internal communication anomalies. It enforces secure data transmission among ADB components, monitors for threats in real time, validates system integrity, and ensures that any detected security issue triggers safe fallback behavior without compromising driver safety.

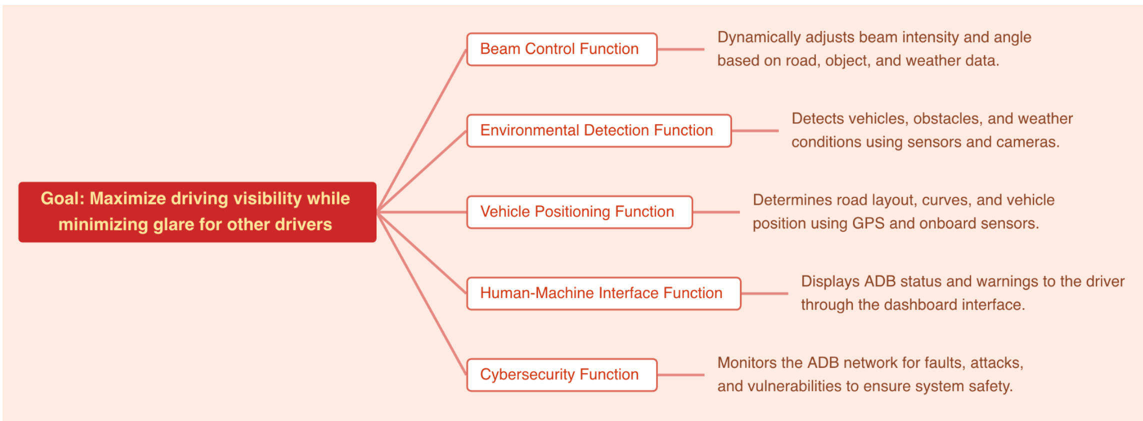


Figure 1: Goal Diagram

Figure 1 showcases the goal of the ADB system, along with the functions our system implements in order to achieve this goal.

2.3 User Characteristics

The primary users of the ADB system are licensed drivers operating vehicles equipped and safety tested with the ADB system installed. The system needs to be activated with a button, but no further input is required from the driver. The system expects the driver to stay on the road, in the appropriate lane, following all laws and regulations of operating a vehicle.

2.4 Constraints

Since the ADB system controls the vehicle’s headlights and affects visibility, there are several safety-critical constraints. Firstly, the system must be able to fail safely by reverting to a standard low-beam mode if any major component, such as the front-facing camera or sensors malfunctions. Communication between the ADB subsystems and the vehicle’s control modules must be low latency to ensure that the headlight beam is adjusted in real time as surrounding conditions change. The ADB must also not interfere with manual lighting controls or any other existing vehicle safety systems. Furthermore, environmental factors such as fog, snow, or heavy rain can limit the accuracy of detection systems (cameras or sensors unable to detect). In such cases, the system should automatically reduce functionality or deactivate to maintain safety. The system must also adhere to all automotive lighting standards and ensure that glare to other drivers is prevented. Lastly, as a cybersecurity-aware system, the ADB must ensure that all internal and external communications are authenticated and protected from unauthorized access that could compromise system integrity or vehicle safety.

2.5 Assumptions and Dependencies

For the development of the ADB system, it is assumed that the vehicle is equipped with functional hardware components including front-facing cameras, environmental sensors, and headlights capable of dynamic beam adjustment. The electrical and communication systems within the vehicle are assumed to be fully operational, allowing communication between the ADB and its subsystems, potentially other modules as well. The driver is expected to understand the system, particularly how to enable it, and when the system will revert to low-beam mode. Furthermore, the ADB software and firmware are assumed to be properly installed, securely updated, and free of corruption.

2.6 Apprportioning of Requirements

To ensure that the scope of the current ADB system remains manageable, certain features have been identified as outside the scope of this version of the system. These features may be revisited and incorporated in future iterations of the ADB system once foundational functionality is established.

- **Vehicle Detection Mechanisms:**
  The development of proprietary algorithms for detecting oncoming or leading vehicles is outside the scope of this project. The present system relies on the vehicle's existing, built-in detection capabilities to identify other road users.
- **Weather Condition Detection:**
  The creation of sensing or classification methods for detecting weather conditions (e.g., fog, precipitation severity, dust, glare conditions) is deferred. This version of the system assumes the availability of relevant environmental data through the vehicle's standard sensor suite.

These deferred requirements do not affect the core safety goals of the ADB system. They instead represent potential areas for enhancement if necessary.

3 Specific Requirements

Global Invariant Requirements

1. In the event of a sensor failure, the ADB system shall:
    1.1. Revert to low-beam mode.
    1.2. Alert the driver of a sensor failure with a visual pop-up message through the VIS.
2. Users must be able to activate/deactivate the ADB system by pressing the on/off button located on the left hand side of the steering wheel.
    2.1. The HMI will send a visual pop-up message to the VIS that the ADB has been activated/deactivated.
    2.2. When the ADB button is pressed and the ADB system is inactive:
        2.2.1. EDS checks if the environment is dark enough to activate the ADB.
            2.2.1.1. If the environment is dark enough, then the ADB system activates into low beam mode.
            2.2.1.2. If the environment is not dark enough, the ADB system will be deactivated.
    2.3. When the ADB button is pressed and the ADB system is active:
        2.3.1. The ADB system will become deactivated.
3. The system must only respond to secure communication between the ADB subsystems (ADB, EDS, VPS, HMI, CSS, BCS).
4. The ADB system should never exceed 3000 lumens.

Hardware Requirements

1. The ADB shall adjust the headlight beam's shape and direction when:
    1.1. No other vehicle within 160 meters in front of the driver [1].
        1.1.1. High Beam mode is activated and lumen strength set to 3000 lumens [1].
    1.2. A vehicle is detected within 160 meters in the oncoming lane [1].
        1.2.1. Low Beam mode is activated and lumens strength set to 1000 lumens [1].
    1.3. The driver is trailing a vehicle by 95 meters in the same lane [1].
        1.3.1. Low-beam mode activated and lumen strength set to 1000 lumens [1].
2. The EDS will detect other vehicles, pedestrians, road signs, and environmental conditions through a combination of built in sensors and cameras such as:

2.1. Radar sensors, located on the front of the vehicle, will be used to detect vehicles 95 meters in the same lane and 160 meters in the opposing lane.

2.2. Front view cameras will be used to capture the forward facing image and detect other vehicles, pedestrians, road signs, and environmental conditions using deep learning.

2.3. Front view cameras will be used to capture the forward facing image and detect:

2.3.1. Vehicles.

2.3.2. Pedestrians.

2.3.3. Road signs.

2.3.4. Environmental conditions (Rain, Fog, Curve in road, incline/decline in road).

Software Requirements

1. The ADB system shall constantly monitor the current driving conditions and adjust the beam accordingly.

1.1. The BCS controls:

1.1.1. illumination levels.

1.1.2. direction of headlight beams.

1.2. The BCS receives inputs from the EDS.

1.3. The EDS and the VPS will work in conjunction to determine where the vehicle is in relation to the road and other road vehicles.

2. The VPS will work with the EDS to map out the relative position of the vehicle.

2.1. The EDS will detect the position of other vehicles and the road.

2.2. The VPS will take these inputs and calculate the position and direction of the vehicle relative to these entities.

3. The ADB system must be capable of responding to environmental changes, such as fog, precipitation, and road curve.

3.1. In the event of heavy fog or heavy rain:

3.1.1. Switch to low beam mode.

3.1.2. Set lumen strength to 1000 lumens.

3.2. The EDS will monitor the curve of the road and send messages to the BCS.

3.3. The BCS will interpret these messages from the EDS to adjust the brightness and directions of the beams.

4. The HMI provides feedback to the driver of the status and warnings of the ADB.

4.1. If all conditions are met (no oncoming traffic within 160 meters, not trailing a vehicle within 95 meters, clear weather conditions, environment is dark enough):

4.1.1. activate high beam mode and set strength to 3000 lumens [2].

4.2. If an error is detected in the system:
    4.2.1. The HMI will send a warning visual pop-up message to the VIS with an audio cue.
    4.2.2. Low beam beam shall be activated and lumen strength set to 1000 lumens.
4.3. If an oncoming vehicle is detected by the EDS:
    4.3.1. The HMI will send a visual pop-up message to the VIS.
    4.3.2. Low beam mode shall be activated and lumen strength set to 1000 lumens.
4.4. If extreme weather conditions such as fog or heavy rain are detected by the EDS:
    4.4.1. The HMI will send a visual pop-up message to the VIS.
    4.4.2. Low beam mode shall be activated and lumen strength set to 1000 lumens.

Cybersecurity Requirements

1. The CSS should call for secure and encrypted communication between:
    1.1. Subsystems (ADB, EDS, VPS, HMI, CSS, BCS).
    1.2. Authentication protocols for software and firmware system updates.
    1.3. Real-time threat detection mechanisms.
2. The ADB shall be equipped with a CSS that should be capable of:
    2.1. Protection of the ADB system.
    2.2. Monitoring of the ADB system.
    2.3. Detection of the ADB system.
    2.4. Mitigation of cyber threats.
3. The system shall log cybersecurity events for diagnostic and forensic purposes
    3.1. The CSS logs shall be accessible through the OBD-II port by a mechanic at a local dealership.

4 Modeling Requirements

This section presents visual diagrams of the system based on the requirements stated above. The models demonstrate the communication and functionality within the system. The included diagrams are the Use Case diagram, Domain Model, Sequence diagrams, and State diagrams.

4.1 Use Case Diagram

A use case diagram is used to demonstrate the simple functions and user interactions with the system. The use case description tables detail each use case.

Figure 2 is the use case diagram for the ADB system. It is meant to display the higher level observable behavior of the system. The diagram contains an actor representing the car driver and a box representing the system. Inside the box is the functionality of the system. The driver actor has a connection to two circles: Turn off and turn on the system. The “turn on” circle has a connection to two other circles: Detect objects and Detect Road. Both of these circles connect to one circle called adjust beam settings.

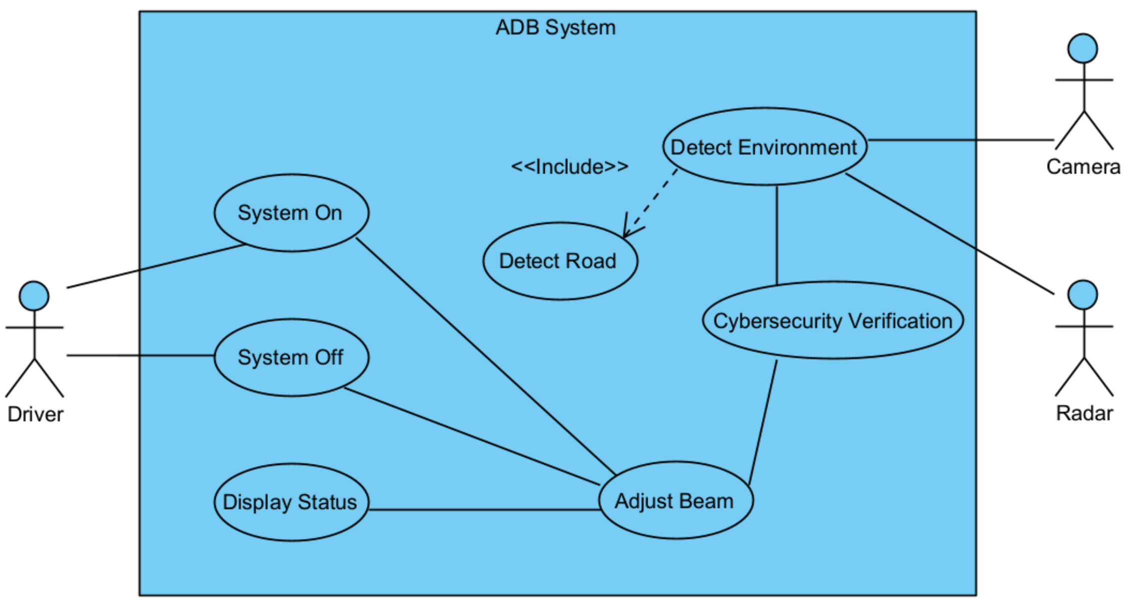


Figure 2: Use case diagram for the ADB system

Use Case:	System On
Actors:	Driver
Description:	After the user/driver turns on the car they have the option to press a button to activate the ADB system (turn on), from here the ADB system will then engage and begin to change the beam pattern based on the situation its currently in
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Invariant 2.2
Use cases	-

Table 1: Use case description for *System On*

Use Case:	System Off
Actors:	Driver
Description:	When the system runs into an issue where it struggles to calculate where other vehicles are in its current situation it will disengage (turn off) the ADB system to prevent harm to other actors in that current scenario. If there is a cyberattack the ADB will automatically turn off. Lastly, the user has the option to manually turn off the ADB system if they want to simply manually adjust the headlights.
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Invariant 2.3
Use cases	System must be active

Table 2: Use case description for *System Off*

Use Case:	Detect Environment
Actors:	Camera, Radar
Description:	Uses inputs from camera and radar to detect weather conditions and oncoming vehicles
Type:	Primary (essential)
Includes:	Detect Road
Extends:	-
Cross-refs:	Hardware 2, Software 1.3, 4.3, 4.4
Use cases	-

Table 3: Use case description for *Detect Environment*

Use Case:	Detect Road
Actors:	-
Description:	Sends data about road curve and relative position on the road to the EDS
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Software 1.3
Use cases	-

Table 4: Use case description for *Detect Road*

Use Case:	Cybersecurity Verification
Actors:	-
Description:	Secures messages between subsystems
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Invariant 3, Cybersecurity 1, 2, 3
Use cases	-

Table 5: Use case description for *Cybersecurity Verification*

Use Case:	Adjust Beam
Actors:	-
Description:	Uses inputs from the EDS and the VPS to dynamically adjust the angle and brightness of the beam.
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Software 1, 3.2, 3.3
Use cases	-

Table 6: Use case description for *Adjust Beam*

Use Case:	Display Status
Actors:	-
Description:	Displays status and warning messages to the user
Type:	Primary (essential)
Includes:	-
Extends:	-
Cross-refs:	Invariant 2.1, Software 4
Use cases	-

Table 7: Use case description for *Display Status*

4.2 Domain Model and Data Dictionary

The domain model and data dictionary are described in this section. The domain model is represented by a class diagram that visualizes the attributes, operations, and relationships between objects. The data dictionary provides a detailed description of each class and its properties in a table format.

4.2.1 Domain Model

Figure 3 is the class diagram for the domain model. The parent class is the ADB system, which owns all subsystems. Each subsystem inherits from an interface called SecureSubsystem. This allows for communication between each subsystem and the Cybersecurity subsystem. The EDS and VPS receive inputs from the camera and radar sensor classes. The EDS receives inputs from the VPS. The BCS receives input from the EDS. The BCS sends output to the beam class. The HMI receives messages from other SecureSubsystem classes.

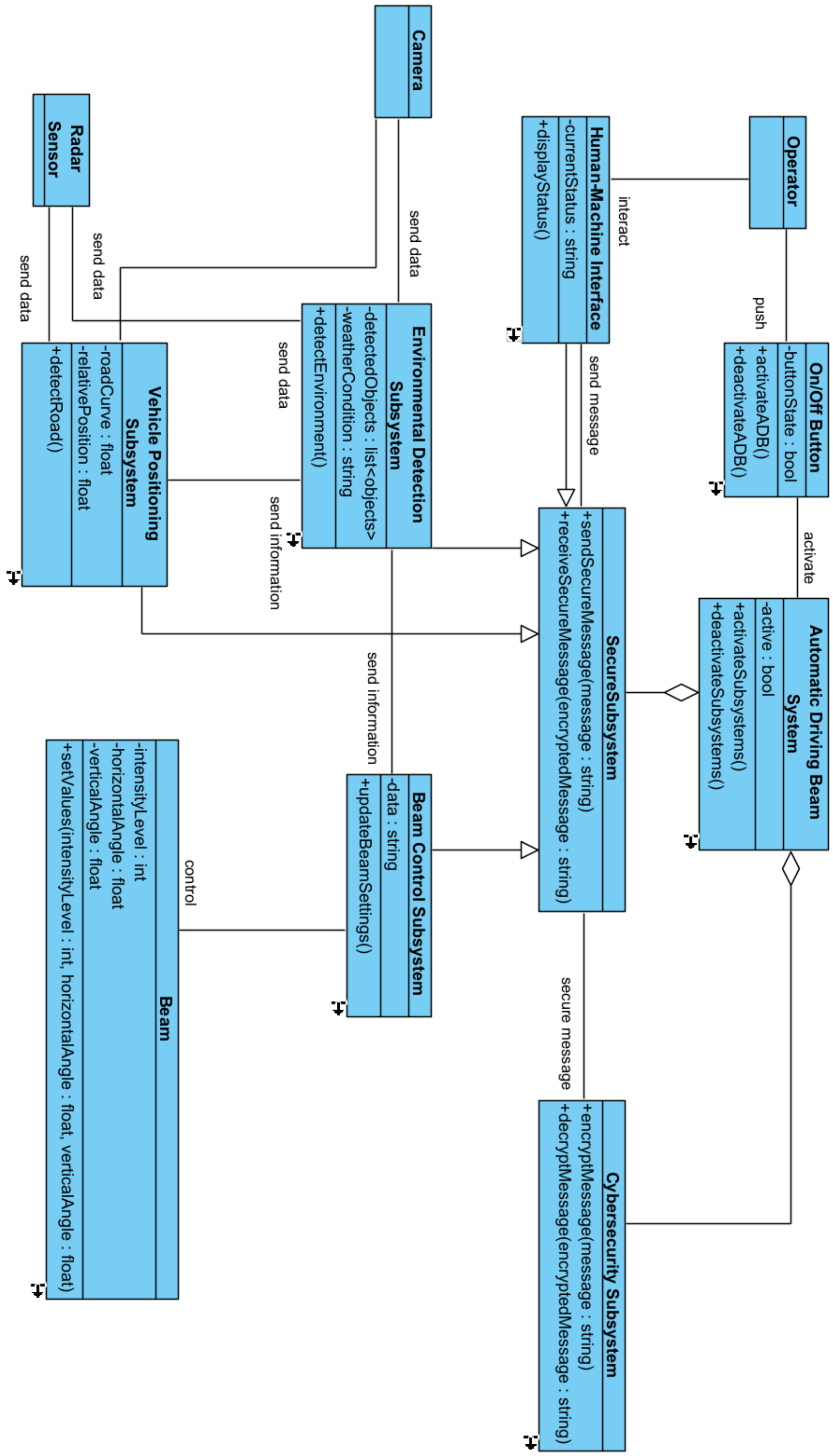


Figure 3: Domain Model for the ADB system

4.2.2 Data Dictionary

The following tables describe the data dictionary for the domain model. Each table represents each class. It describes the class, its attributes, operations, and relationships with other classes.

Element Name		**Description**
Beam		Controls light emission direction, pattern, and on/off behavior.
Attributes		
	intensityLevel : int	The current illumination level of the beam
	horizontalAngle : float	The current horizontal angle that the beam is pointing in
	verticalAngle : float	The current vertical angle that the beam is pointing in
Operations		
	setValues(intensityLevel : int, horizontalAngle : float, verticalAngle : float)	Sets new values for intensity level, horizontal angle, and vertical angle
Relationships	Beam Control System - Receives control messages	
UML Extensions		

Table 8: Data dictionary entry for *Beam*

<table>
<tr><th colspan="2">Element Name</th><th>Description</th></tr>
<tr><td colspan="2">Beam Control Subsystem</td><td>Determines proper beam settings based on environment and road conditions.</td></tr>
<tr><td>Attributes</td><td></td><td></td></tr>
<tr><td></td><td>data : string</td><td>The current conditions received from the EDS</td></tr>
<tr><td>Operations</td><td></td><td></td></tr>
<tr><td></td><td>updateBeamSettings()</td><td>Calculates the appropriate values for the beam’s illumination levels and direction</td></tr>
<tr><td>Relationships</td><td colspan="2">ADB system – Aggregation
Secure Subsystem – Inheritance
Environmental Detection Subsystem - receives environmental data
Beam - sends control messages</td></tr>
<tr><td>UML Extensions</td><td colspan="2"></td></tr>
</table>

Table 9: Data dictionary entry for *Beam Control Subsystem*

Element Name		**Description**
Camera		Captures and sends image data for environment analysis and detection.
Attributes		
	frameRate : int	Description1
	cameraId : string	Description2
Operations		
	sendImageData()	Description3
Relationships	Environmental Detection Subsystem - sends image data Vehicle Positioning Subsystem - sends image data	
UML Extensions		

Table 10: Data dictionary entry for *Camera*

Element Name		**Description**
Cybersecurity Subsystem		Monitors communications and activates responses to detected security threats
Attributes		
Operations		
	encryptMessage(message : string): string	Encrypts a message from a subsystem to securely send it to another subsystem
	decryptMessage(encrypted Message : string)	Decrypts a secured message sent from a subsystem
Relationships	ADB system – Aggregation Secure Subsystem – encrypts and decrypts messages	
UML Extensions		

Table 11: Data dictionary entry for *Cybersecurity Subsystem*

Element Name		**Description**
Environmental Detection Subsystem		Detects oncoming/following vehicles and analyzes weather/environment.
Attributes		
	detectedObjects : list<object>	List of objects on the road
	weatherCondition : string	Current weather condition
Operations		
	detectEnvironment()	Detects the current objects on the road and weather condition
Relationships	ADB system – Aggregation Secure Subsystem – Inheritance Beam Control Subsystem – sends environmental data	
UML Extensions		

Table 12: Data dictionary entry for *Environmental Detection Subsystem*

Element Name		**Description**
Human Machine Interface		Interface between operator and ADB system for displaying information and capturing input.
Attributes		
	currentStatus : string	The current status message to be displayed
Operations		
	displayStatus()	Displays the correct status message
Relationships	ADB system – Aggregation Secure Subsystem – Inheritance Operator – displays messages	
UML Extensions		

Table 13: Data dictionary entry for *Human Machine Interface*

Element Name		**Description**
On/Off Button		Enables or disables the ADB feature.
Attributes		
	buttonState : boolean	The current state of the system (on/off)
Operations		
	activateADB()	Turns on the ADB system
	deactivateADB()	Turns off the ADB system
Relationships	Operator – Operator pushes to activate system ADB system – activates system	
UML Extensions		

Table 14: Data dictionary entry for *On/Off Button*

Element Name	**Description**

Operator		Human drivers who receive info, and may override ADB systems at any time.
Attributes		
Operations		
Relationships	On/Off button – Operator pushes to activate system Human Machine Interface - reads messages	
UML Extensions		

Table 15: Data dictionary entry for *Operator*

Element Name		**Description**
Radar Sensor		Detects environmental information such as lighting and distance.
Attributes		
	sensorType : string	Description1
	status : boolean	Description2
Operations		
	GetDistanceData()	Description3
Relationships	Environmental Detection Subsystem - sends sensor data Vehicle Positioning Subsystem - sends sensor data	
UML Extensions		

Table 16: Data dictionary entry for *Radar Sensor*

Element Name		**Description**
Secure Subsystem		Parent class for Subsystems that make secure message transfers
Attributes		
Operations		
	sendSecureMessage(message : string)	Sends messages securely by encrypting them with CSS
	receiveSecureMessage(encryptedMessage : string)	Receives messages securely by decrypting them with CSS
Relationships	ADB system – Aggregation Cybersecurity Subsystem – sends messages to secure	
UML Extensions		

Table 17: Data dictionary entry for *Secure Subsystem*

Element Name		**Description**
Vehicle Positioning Subsystem		Determines lane, heading, and path to support beam direction decisions.
Attributes		
	roadCurve : float	The curvature of the road ahead in degrees
	relativePosition : float	The relative horizontal position of the vehicle to the road
Operations		
	detectRoad()	Updates the current road curvature and relative position of the vehicle
Relationships	ADB system – Aggregation Secure Subsystem – Inheritance Environment Detection Subsystem – sends positioning data	
UML Extensions		

Table 18: Data dictionary entry for *Vehicle Positioning Subsystem*

4.3 Sequence Diagrams

Various sequence diagrams are introduced in this section. Each diagram represents a unique scenario that the ADB system may encounter.

Sequence diagrams follow this notation: Each class represented by a box and a vertical line called a lifeline. Actors are represented by a stick figure. Each lifeline is connected to another through messages.

4.3.1 Scenario One

Scenario One: The ADB system operates effectively on a straight highway at night, with no other vehicles present. Demonstrate the full high-beam illumination

Figure 4 demonstrates how the system operates in optimal driving conditions (clear weather, not oncoming vehicles, and a straight road). The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. From these operations, the EDS determines that detectedObjects is empty and weatherCondition is "Clear", so it calls updateBeamSettings() on the BCS. The BCS is then able to call setValues() on the beam to set the beam to maximum intensity and point forward. After setting the new beam settings, the BCS calls displayStatus() on the HMI to update the current status of the system.

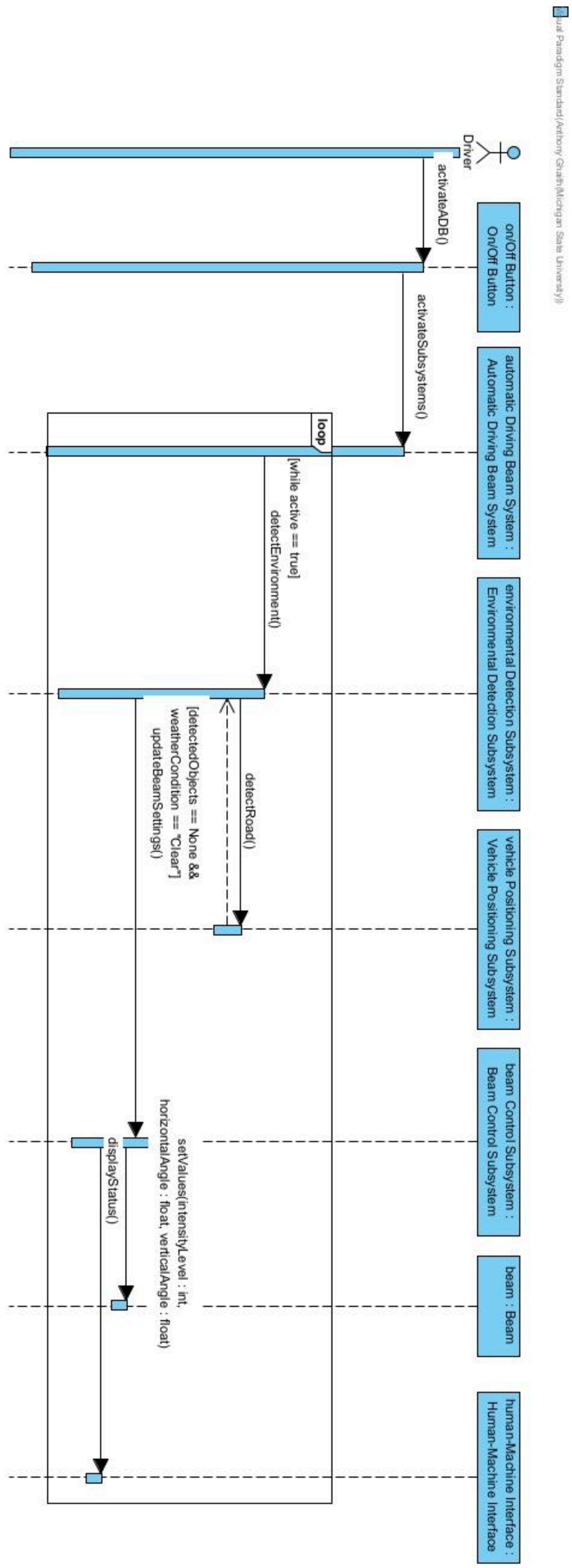


Figure 4: Sequence Diagram Scenario 1

4.3.2 Scenario Two

Scenario Two: The system adjusts the beam pattern when an oncoming vehicle is detected, ensuring no glare while maintaining maximum illumination on other road areas.

Figure 5 demonstrates how the system operates when an oncoming vehicle is detected. The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. From these operations, the EDS determines that detectedObjects is not empty, so it calls updateBeamSettings() on the BCS. The BCS is then able to call setValues() on the beam to set the beam to minimum intensity and angled downward. After setting the new beam settings, the BCS calls displayStatus() on the HMI to update the current status of the system.

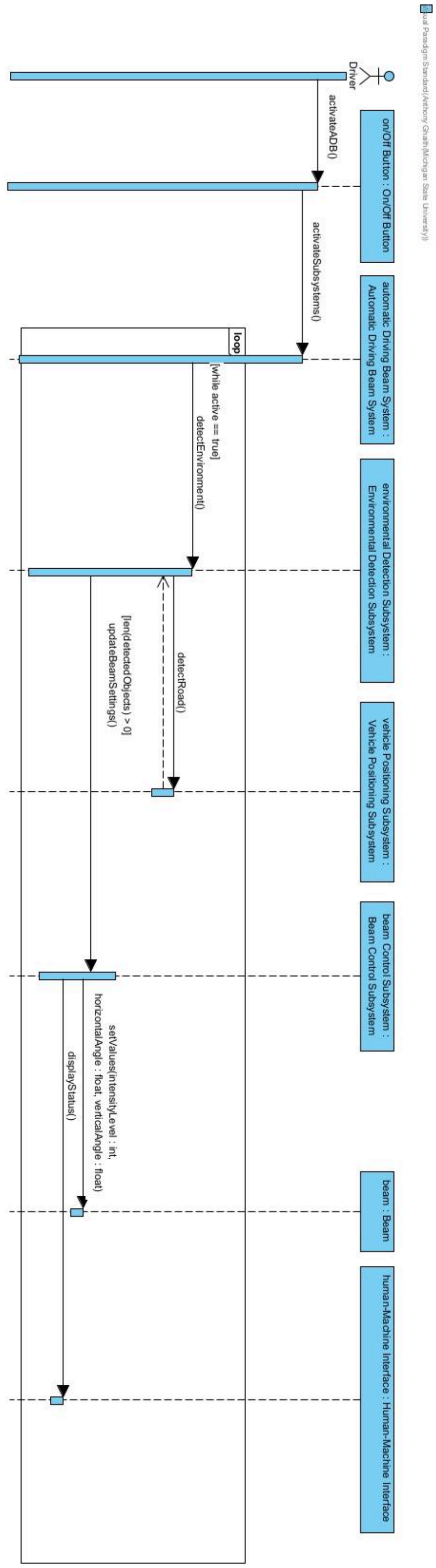


Figure 5: Sequence Diagram Scenario 2

4.3.3 Scenario Three

Scenario Three: Demonstrate the system's ability to handle curves and inclines, adjusting the beam direction to keep the road well-lit without blinding oncoming traffic.

Figure 6 demonstrates how the system operates when driving on curves and varying slopes. The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. From these operations, the EDS determines that roadCurve is not 0, so it calls updateBeamSettings() on the BCS. The BCS is then able to call setValues() on the beam to angle it in the direction of the road curve. After setting the new beam settings, the BCS calls displayStatus() on the HMI to update the current status of the system.

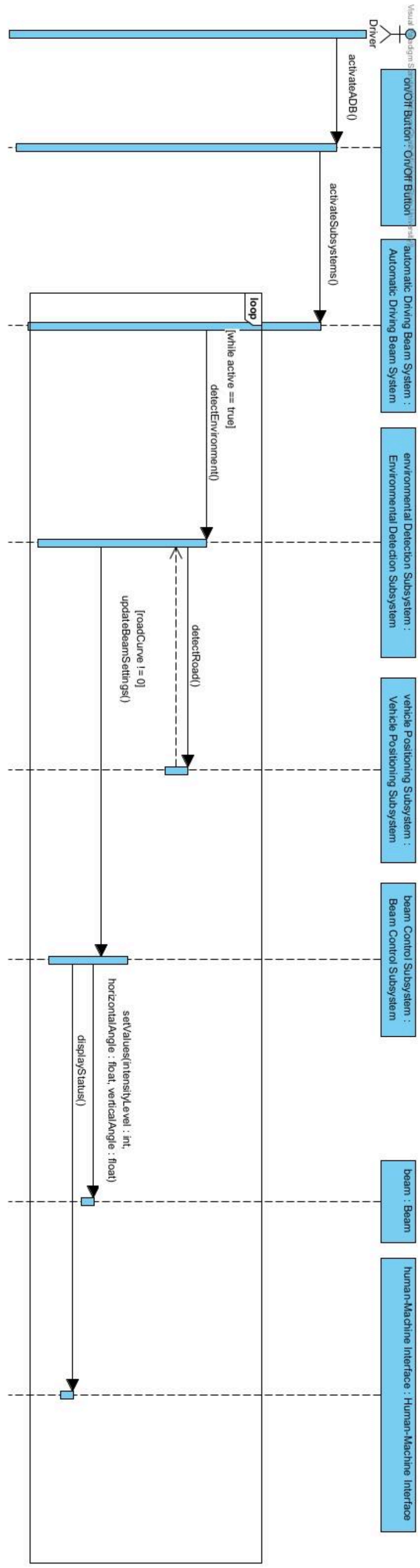


Figure 6: Sequence Diagram Scenario 3

4.3.4 Scenario Four

Scenario Four: The ADB system experiences a sensor failure and safely reverts to low-beam mode. Demonstrate the transition and any associated driver alerts.

Figure 7 demonstrates how the system securely sends messages between subsystems and how it behaves when a failure is detected. The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. The EDS then calls sendSecureMessage(), which calls encryptMessage() on the CSS. With the encrypted message, the EDS can call updateBeamSettings() on the BCS. The BCS calls receiveSecureMessage() to read the message. It calls decryptMessage() from the CSS, but the encrypted message was corrupted when sending to the BCS, so the CSS returns an error. The BCS sets the system to low-beam mode and calls displayStatus() on the HMI to alert the driver of a system error. The BCS then calls deactivateSubsystems() on the ADB system to shut it down.

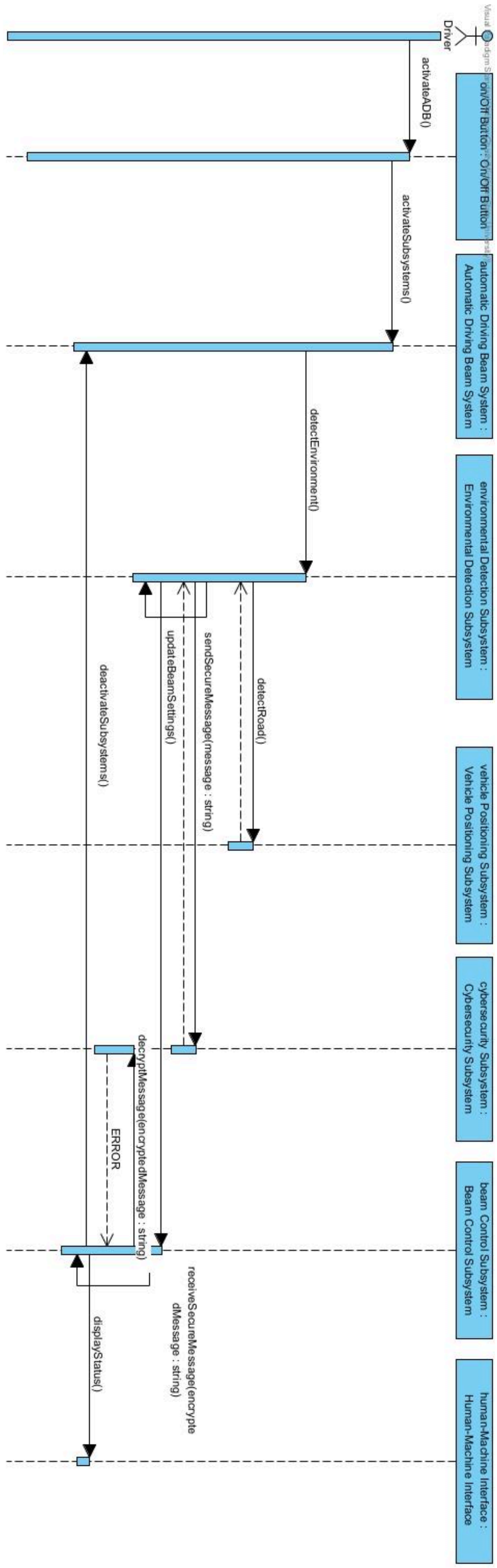


Figure 7: Sequence Diagram Scenario 4

4.3.5 Scenario Five

Scenario Five: The vehicle drives through dense fog, and the ADB system optimizes the beam pattern by reducing the intensity of the high beams to prevent light reflection off the fog. Show how the system maintains visibility while preventing glare that could impair the driver's vision.

Figure 8 demonstrates how the system operates in dense fog. The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. From these operations, the EDS determines that detectedObjects is empty and weatherCondition is "heavyFog", so it calls updateBeamSettings() on the BCS. The BCS is then able to call setValues() on the beam to set the beam to minimum intensity and angled downward. After setting the new beam settings, the BCS calls displayStatus() on the HMI to update the current status of the system.

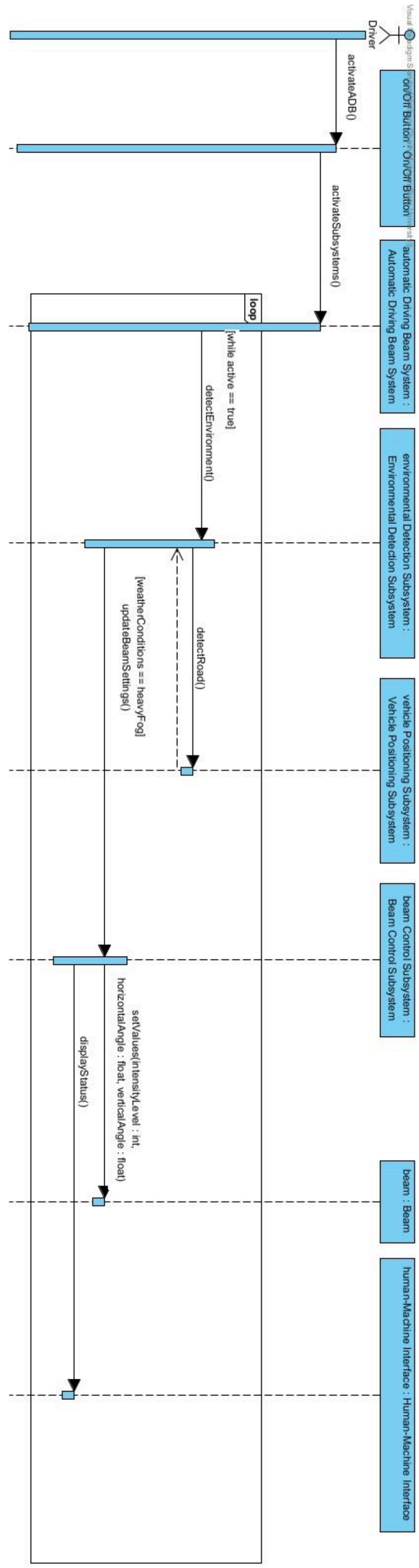


Figure 8: Sequence Diagram Scenario 5

4.3.6 Scenario Six

Scenario Six: The ADB system responds to heavy precipitation by deactivating to eliminate glare, alerting the driver

Figure 4 demonstrates how the system responds to heavy precipitation. The sequence diagram starts with the driver pressing the on/off button. This activates the ADB system, which then activates all the other subsystems. After all subsystems are active, the run time loop can begin. The ADB system calls detectEnvironment() on the EDS, which also calls upon detectRoad() from the VPS. From these operations, the EDS determines that weatherCondition is “heavyRain”, so it calls updateBeamSettings() on the BCS. The BCS is then able to call setValues() on the beam to set the beam to minimum intensity. It calls displayStatus() on the HMI to alert of the system deactivating. The BCS then call deactivateSubsystems() on the ADB system.

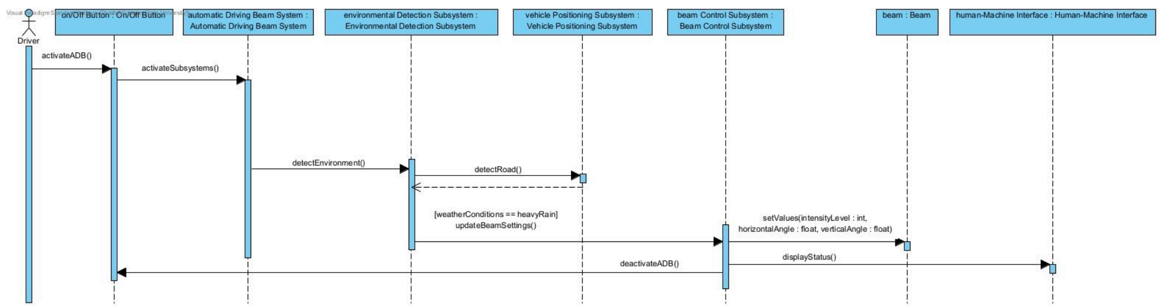


Figure 9: Sequence Diagram Scenario 6

4.4 State Diagrams

This section contains state diagrams that describe how different components of the ADB system move from state to state. In the following state diagrams, states are represented by blue rectangles, and transitions between states are denoted by arrows. The text on each arrow indicates the transition condition that causes the state to change. In a state diagram, the beginning state is depicted by a black-filled circle with an arrow leading to the initial state.

In Figure 10, the ADB system starts in the “Off” state. The system switches to the “On” state when the driver presses the on/off button which calls activateADB(). The system switches back to the “Off” state when the driver presses the on/off button which calls deactivateADB().

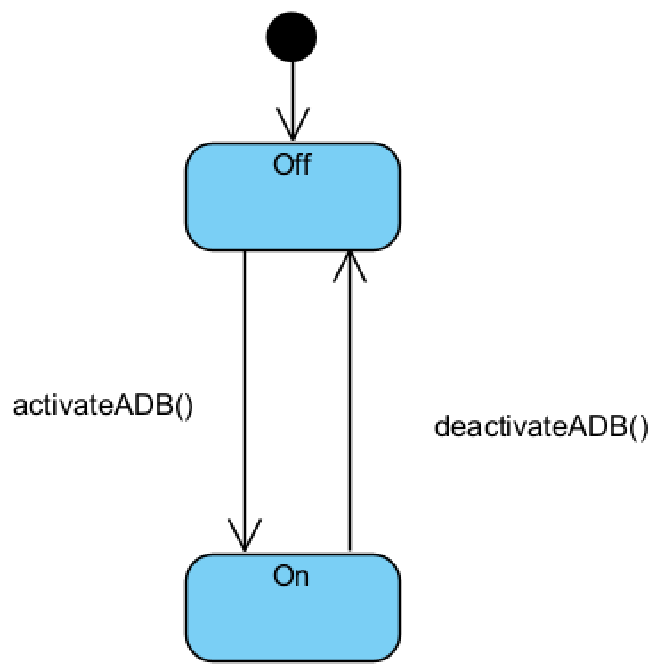


Figure10: State Diagram for the ADB system

In Figure 11, the EDS starts in the "Detecting Objects" state. It stays in this state by continuously calling detectObjects() until the whole system is deactivated.

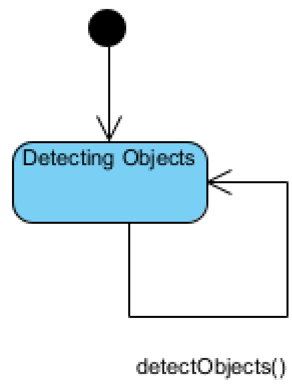


Figure 11: State Diagram for the Environmental Subsystem

In Figure 12, the CSS starts in the "Waiting for message" state and remains there until it receives a message from another subsystem. It has two transition options: encryptMessage() and decryptMessage(). encryptMessage() switches the state to "Encrypting message" and decryptMessage() switches the state to "Decrypting message." When finished encrypting, it sends the message back to the subsystem with sendSecureMessage() and returns to the "Waiting for message" state. When finished decrypting, its send the message back to the subsystem with receiveSecureMessage() and returns to the "Waiting for message" state.

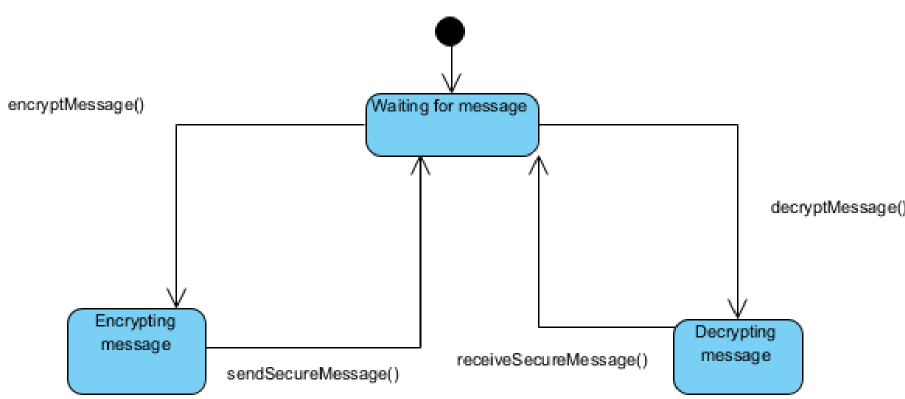


Figure 12: State Diagram for the Cybersecurity Subsystem

In Figure 13, the VPS starts in the "Detecting Road" state. It stays in this state by continuously calling detectRoad() until the whole system is deactivated.

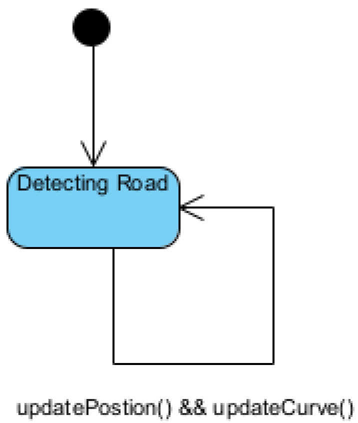


Figure 13: State Diagram for the Vehicle Positioning Subsystem

In Figure 14, the HMI subsystem begins in the "Waiting for Message" state. It transitions to the "Displaying Message" state when receiveSecureMessage() is called on it. It remains in this state while displayStatus() is called.

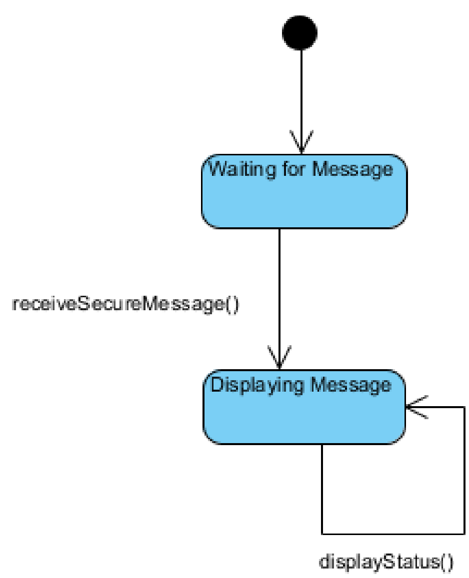


Figure 14: State Diagram for the Human Machine Interface Subsystem

In Figure 15, the BCS begins in the "Waiting For Message" state. It transitions to the "Updating Beam Settings" when the EDS calls updateBeamSettings() on it. The BCS transitions back to the "Waiting For Message" state when it completes updating and calls sendSecureMessage() to send a message to the HMI.

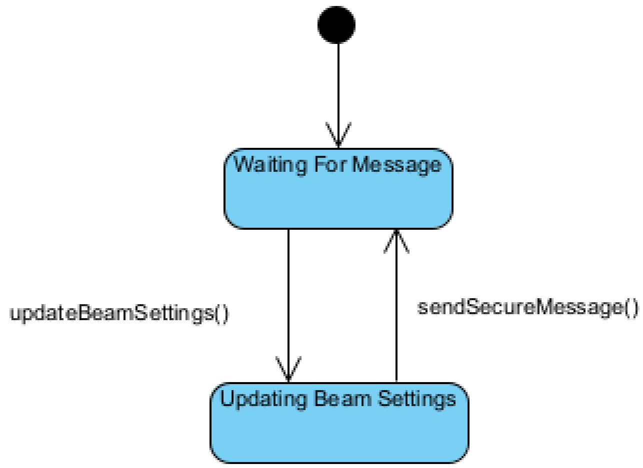


Figure 15: State Diagram for the Beam Control Subsystem

5 Prototype

The prototype will demonstrate multiple different scenarios that the ADB system will handle. There are a total of 5 different scenarios that are showcased in the prototype. They demonstrate the ADB's ability to adapt to different scenarios that a vehicle will face on the road, showcasing the system's ability to maximize the visibility across a range of conditions while still minimizing the glare onto other vehicles.

5.1 *How to Run Prototype*

The prototype can be run on this page https://neubauerseth.itch.io/adb2-prototype [10]. It is also available on the project website [5]. To run the prototype, visit the Itch.io page, which can be accessed via this SRS and the project website, then hit the play button. The prototype will run a demonstration of the ADB system functionality across five different scenarios. There are four different buttons located in the top right corner that allow users to initiate a scenario such as clear sky, rain, fog, and sensor failure.

5.2 Sample Scenarios

5.2.1 Scenario 1: Clear Highway

This scenario represents the ADB system at full visibility. This occurs when there are no objects detected, weather conditions are clear, and the system has no errors. The ADB system adjusts headlight patterns for full visibility of the road, ensuring optimal visibility for the driver with no objects present for glare.

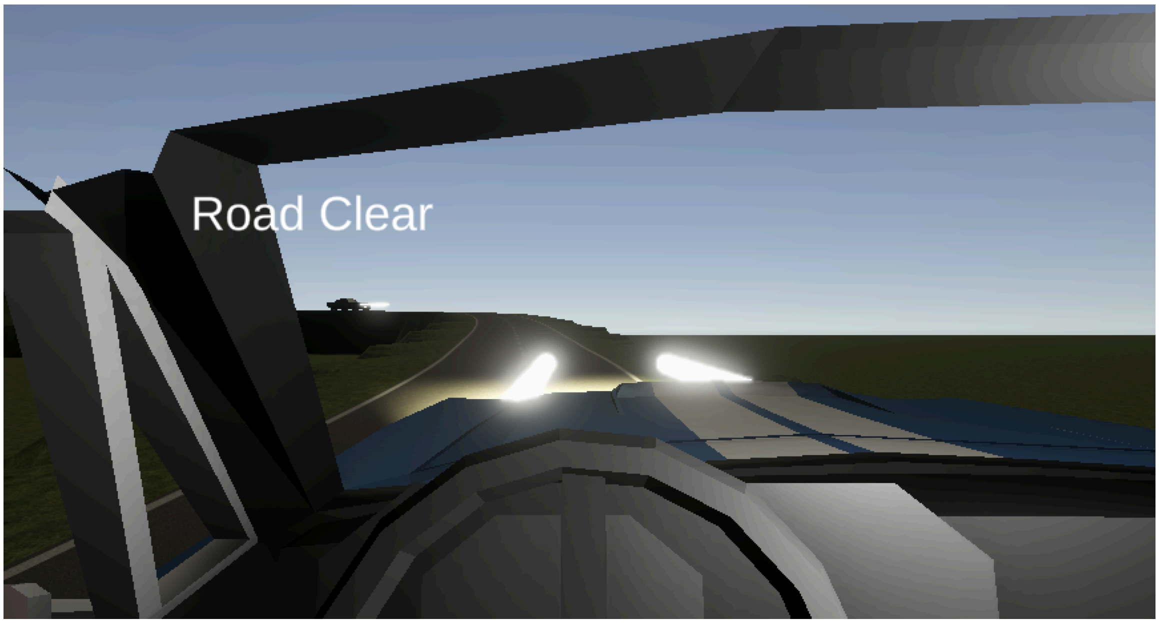


Figure 16: Scenario 1 Prototype

5.2.2 Scenario 2 : Oncoming Vehicle

This scenario represents when the system needs to adjust the beam pattern when an oncoming vehicle is detected, ensuring no glare while maintaining maximum illumination on the rest of the road. Figure 17 shows the system when the road is clear, activating high beam mode, illuminating the area in front of the driver, figure 18 shows when there is an oncoming vehicle, in order to prevent glare onto the oncoming vehicle, the system uses the many led lite headlights in order to create dark spots on the approaching vehicle, while still maintaining maximized illumination on the rest of the road.

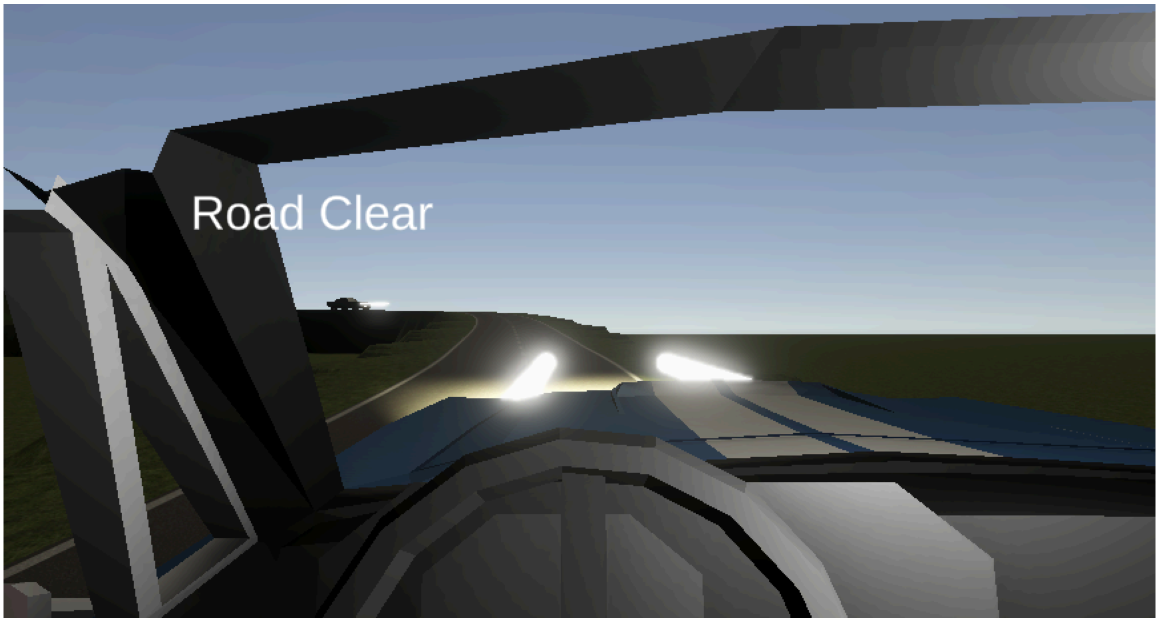


Figure 17: Scenario 2 High Beam Condition

Figure 18: Scenario 2 Low Beam Condition

5.2.3 Scenario 3: Curves and Inclines

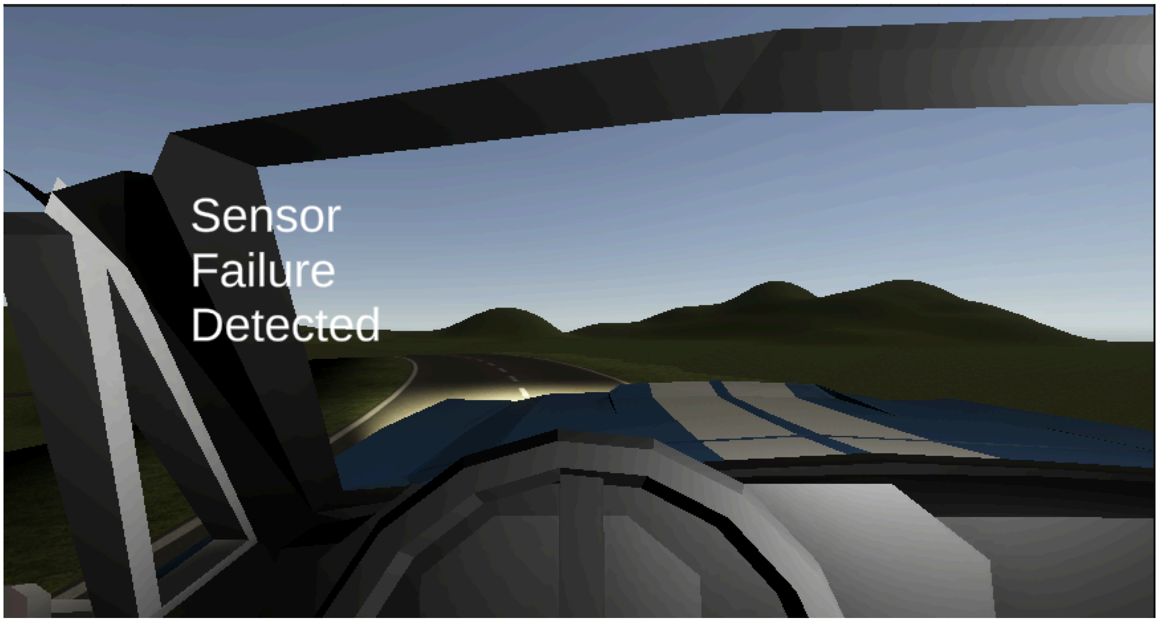
This scenario showcases the ADB system's ability to handle curves and inclines. The system uses GPS and various sensors to detect changes in the elevation and curvature of the road.

These changes are then relayed to the BCS, ensuring that the beam pattern perfectly matches the road condition to ensure maximized visibility while still minimizing the potential glare to other vehicles on the road. Figure 19 showcases the ADB system automatically adjusting to the left turn in the road ahead, ensuring maximized visibility while entering the turn.

Figure 19: Scenario 3 Adjust to Curves and Inclines

5.2.4 Scenario 4: Sensor Failure

Scenario 4 describes when the system experiences a sensor failure, due to blockage, miscommunication, damage, etc. The system responds by notifying the driver of the failure and reverting to low-beam mode. Once sensor failure goes away or is repaired, the system shall revert to full ADB functionality after a 5-second hysteresis period.



20: Scenario 4 reverts to low beam on system failure

5.2.5 Scenario 5: Heavy Fog

Scenario 5 showcases the car reacting to heavy fog. The ADB system responds by lowering the illumination level of the high beams in order to maintain visibility while still preventing glare. Figure 21 demonstrates the ADB system automatically dimming the high beam pattern once fog is detected. Once fog is no longer detected, the system has a 5-second hysteresis period before reactivating the full ADB system.

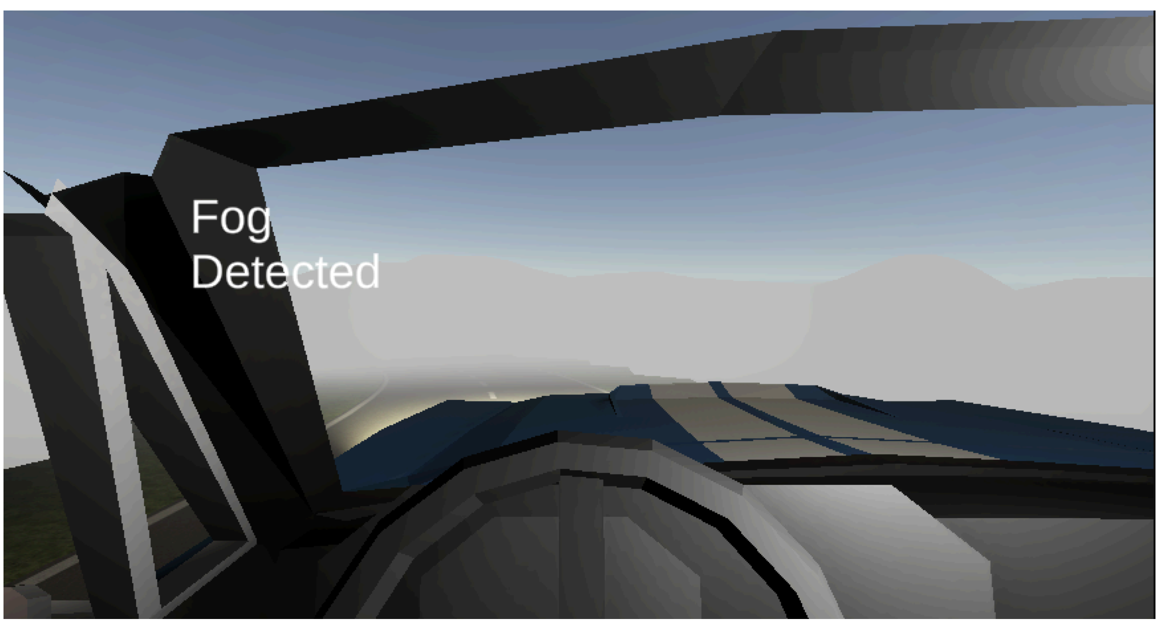


Figure 21: Scenario 5 adjusts to heavy fog

5.2.6 Scenario 6

Scenario 6 demonstrates the system's ability to react to heavy precipitation. When the heavy precipitation is detected, the ADB system reverts to low-beam mode in order to ensure reduced glare and increase safety. Once the heavy precipitation is no longer detected, the system waits a standard 5-second hysteresis period before re-activating the high-beam mode.

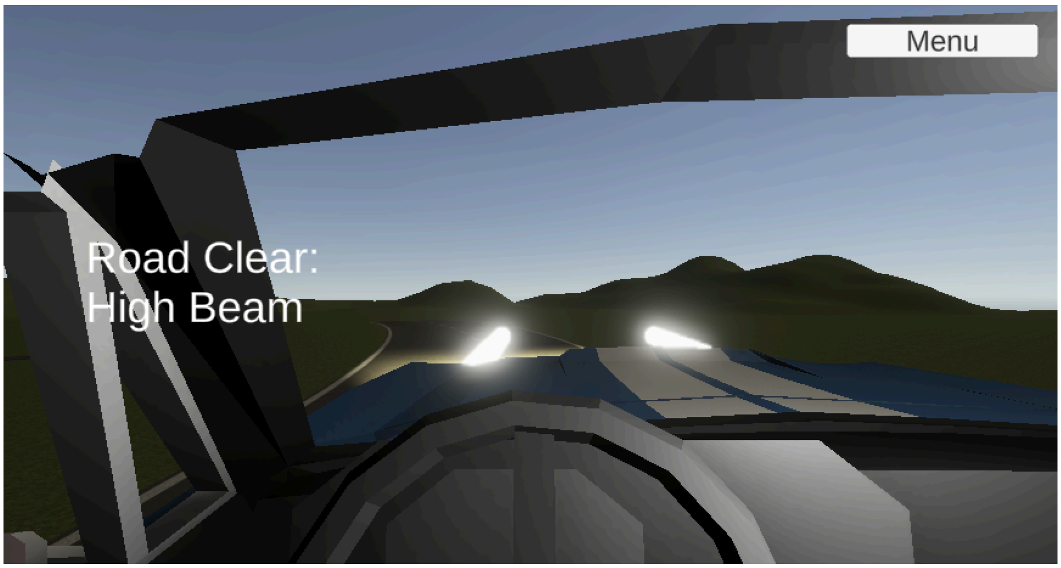


Figure 22: Scenario 6 Clear weather conditions

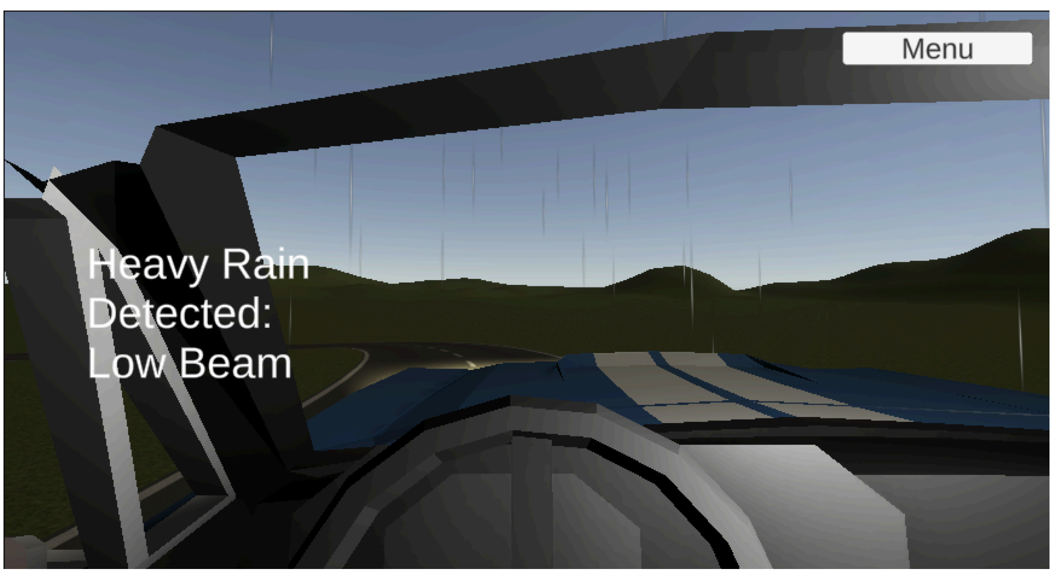


Figure 22: Scenario 6 Heavy Raining Condition

6 References

[1] D. Thakore and S. Biswas, "Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks," Proceedings of IEEE Military Communication, Atlantic City, October 2005.

[2] "California Driver's Handbook, " State of California DMV, 2025, Web..

[3] "Michigan Headlights Laws," Kensun, 2025, Web.

[4] Erich Hairston, Dean Dawson, Gabrie Italia, Evan Shoup, Kaela Burger, "Software Requirements Specification (SRS)", https://www.cse.msu.edu/~cse435/Handouts/SRS_examples/Final-SRS-Burger-Hairston-Dawson-Italia-Shoup.pdf , Used in Section 2, 2.1, and 5 for template modeling purposes

[5] S. Neubauer, "ADB2," *ADB2 Team Page*, 2025. [Online]. Available: https://www.egr.msu.edu/cse435/adb2/

[6] J. Rhodes, "Adaptive Driving Beam (ADB) System," 2025.

[7] M. Falch "ODB2 Explained - A simple Intro [2025]", CSS Electronics, Jan 2025, [Online]. Available: https://www.csselectronics.com/pages/obd2-explained-simple-intro

[8] C.G. Manning, "GPS", NASA, [Online]. Available: https://www.nasa.gov/directorates/somd/space-communications-navigation-program/gps/

[9] The Editors of Encyclopaedia Britannica, "LED", "Encyclopaedia Britannica, [Online]. Available: https://www.britannica.com/technology/LED

[10] A. Aalabdulrasul, A. Ghaith, N. Ghebredngl, M. Vazquez, S. Neubauer, Unity Prototype [Online] Available: https://neubauerseth.itch.io/adb2-prototype

7 Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University (chengb at msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.