ECE 445

Fall 2024

Project Proposal: Monitoring System for Bicycles

Sachin Bhat, Tommy Park, Agrim Kataria

Problem:

Bicycle monitoring systems are becoming increasingly important for enhancing cyclist safety on the road. However, most bicycles currently lack advanced safety features that help cyclists navigate traffic, avoid collisions, and deter theft or vandalism. According to the National Highway Traffic Safety Administration, thousands of cyclists are injured or involved in accidents each year due to cars and other vehicles in their blind spots. Additionally, bicycles parked in high-risk or unfamiliar areas are often vulnerable to theft or vandalism. Our idea is to build a comprehensive bicycle monitoring system that can be easily installed on any bike, providing cyclists with improved safety and security both while riding and when their bike is parked.

Solution:

We propose to develop a blindspot detection and monitoring system equipped with sensors that alert cyclists when another vehicle or obstacle is in their blind spot. The system will use a combination of LEDs mounted on the handlebars or helmet and audible alerts, such as a beeping noise, to notify cyclists of potential hazards. A compact control unit will process real-time sensor data to determine if an obstacle is present in the blind spot. We will use ultrasonic sensors with adjustable sensing distances, which can be customized through a companion mobile application. Additionally, the mobile application will offer riding analytics, such as the number of close calls, the side on which more detections occur, and other metrics to help cyclists improve their riding habits. To further enhance bicycle safety, we will integrate an "Away From Bike" (AFB) subsystem. This subsystem will utilize the same sensors to detect motion or activity around the bicycle while it is parked. If any motion is detected, the system will send an alert to the cyclist's mobile app. This feature provides an extra layer of security for cyclists when parked in unfamiliar or high-risk areas, ensuring bicycle safety even when the bike is not in use. By combining blindspot detection, riding analytics, and the AFB system, our solution provides a comprehensive monitoring and safety package that can be easily retrofitted to any bicycle, significantly enhancing road safety and bicycle security.

Visual Aid:

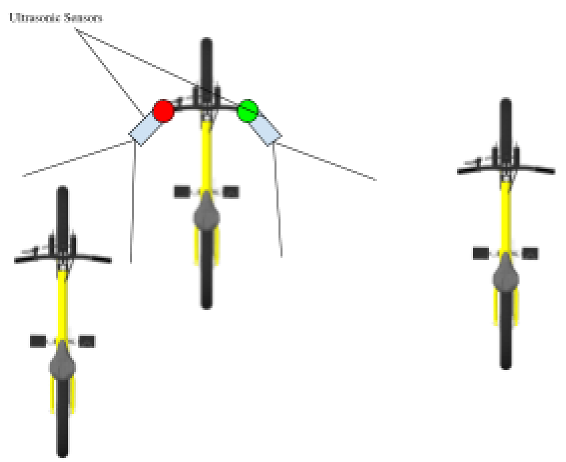


Figure 1: Blind Spot Detection System

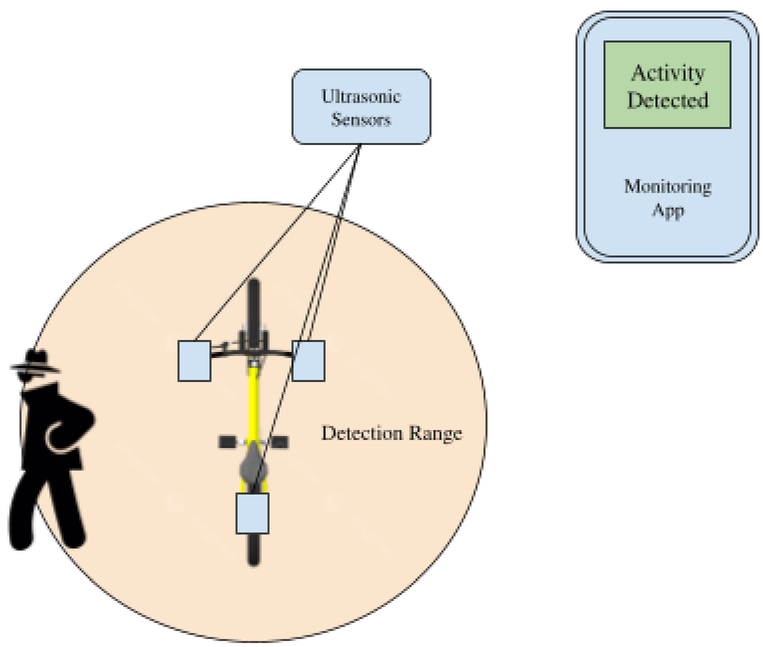


Figure 2: Away from Bike “AFB” System

Our primary system focuses on preventing accidents by alerting cyclists of vehicles or obstacles in their blind spots. This system will utilize a network of ultrasonic sensors strategically positioned on the bicycle's handlebars or helmet to create a real-time awareness zone. These sensors will detect nearby objects within a specified radius—typically up to 3 meters.

Upon detecting an obstacle, the system will issue immediate notifications through:

- **Visual Alerts:** Bright, color-coded LEDs will illuminate to indicate the presence and proximity of an obstacle, with different colors representing various threat levels (e.g., yellow for caution, red for imminent danger).
- **Audible Alerts:** A speaker will emit a beeping sound, drawing the cyclist's attention to potential hazards.

The detection parameters, including sensitivity and alert thresholds, can be customized via a user-friendly mobile application. This personalization allows cyclists to tailor the system to their specific riding conditions and preferences.

In addition to enhancing safety while riding, our solution incorporates the AFB subsystem, which focuses on bicycle security when parked. This subsystem will utilize the same ultrasonic sensors to monitor the area around the bicycle for any unexpected motion or activity.

If movement is detected within a specified range (approximately 1 meter), the AFB system will send an alert to the cyclist's mobile application, ensuring that they are informed of any potential threats. This feature is particularly beneficial for cyclists parking their bikes in unfamiliar or high-risk areas, providing peace of mind and an added layer of protection against theft or vandalism.

The AFB subsystem operates seamlessly alongside the blind spot detection system, allowing for a comprehensive safety and security solution. Both systems will be integrated within a compact control unit, which will process data from the sensors and manage alerts effectively.

By combining these two advanced monitoring systems, our bicycle safety solution not only improves awareness of surroundings while riding but also enhances the security of the bike when it is not in use. This dual functionality ensures that cyclists can enjoy their rides with confidence, knowing they have robust protection both on the road and when parked.

High-Level Requirements:

- **Blind Spot Detection and Alerting**:

  The system shall accurately detect obstacles or vehicles in the cyclist's blind spot within a 3-meter radius and provide real-time visual (LED) and audible alerts within 1 second of detection. This detection will be carried out using ultrasonic sensors mounted on either side of the bicycle to cover blind spots effectively. The system will have an adjustable detection distance, configurable by the user through the mobile application, allowing for customization based on the cyclist’s preferences or riding environment. The LED alerts will flash in a color-coded system (e.g., yellow for approaching objects, red for immediate danger) to provide intuitive, glance-based warnings, while the speaker will emit tones for alerts.

- **Away From Bike (AFB) Monitoring**:

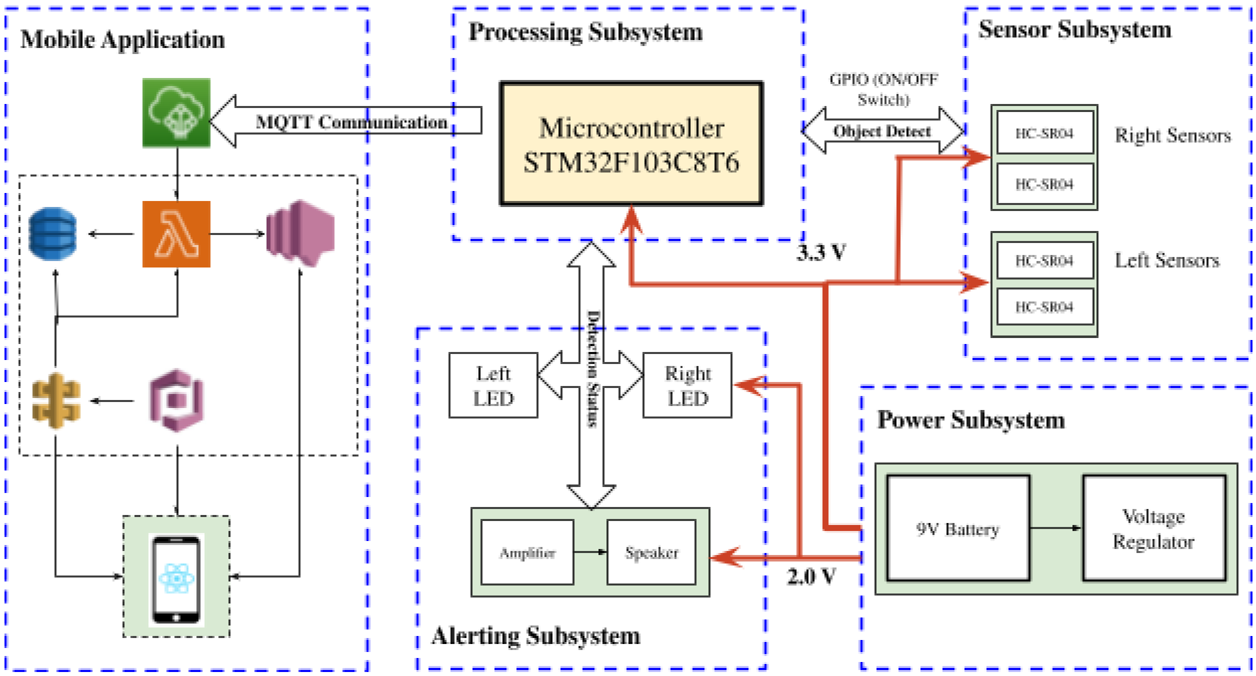
  When the bicycle is parked, the system can be transitioned into AFB mode, monitoring motion around the bike and providing real-time security alerts. The AFB system will use ultrasonic sensors strategically placed around the handlebars to create a 1-meter detection range. If any motion is detected within this range, the system shall send an alert to the cyclist's mobile application within 3 seconds. The AFB system will focus on key areas vulnerable to theft, such as the handlebars, and use both passive infrared sensors and ultrasonic sensors to enhance detection accuracy, minimizing false positives triggered by passing pedestrians or environmental factors like wind.

- **Alerting System and Mobile App:**

  The mobile application will serve as the system's central hub for real-time data visualization and configuration management. It will provide data on blind spot detections, showing the cyclist where potential hazards are located relative to the bike. The mobile app will allow users to adjust detection distances for both Blind Spot Detection and AFB systems. In AFB mode, the app will enable remote monitoring of the bike via notifications that alerts the user if the bicycle is being approached.

Subsystem Overview:

Block Diagram



	AWS IoT Core
	AWS Lambda
	API Gateway
	DynamoDB
	Cognito
	AWS SNS

Figure 5: Mobile App Legend

Sensor Subsystem

This project will feature an array of ultrasonic sensors (**hc-sr04**) on each bike handle that are designed to detect objects at various distances around the bicycle. The sensors will transmit real-time data to a microcontroller (**STM32F103C8T6**) for processing, allowing for accurate blindspot detection. The sensors will be strategically placed on the handlebars or helmet for optimal coverage. Additionally, the sensors will be used in multiple modes, including active monitoring while riding and motion detection when the bicycle is stationary (as part of the AFB system). Each sensor module will transmit data to the receiver module contained on the microcontroller.

Requirements	**Verifications:**
• The ultrasonic sensors must be able to continuously monitor the area within 3 feet of the bicycle to detect other objects.	• Set up the ultrasonic sensor system and power it on.
• This requires a constant transmission of signals to the receiver module to provide real-time updates. • This requires a constant transmission of signals to the receiver module to provide real-time updates. • Connect the ultrasonic sensors to the receiver module and ensure that data is transmitted to the monitoring system in real-time. • Check for delays in updates to confirm they are less than 100 ms during operation.	• Measure the frequency of the signals being transmitted using an oscilloscope to ensure continuous transmission.

Mobile Application Subsystem

The frontend of the cyclist Blind Spot Detection and AFB mobile application will be created using React Native, and we will use AWS Amplify to develop and deploy the mobile application. The system will allow cyclists to visualize blind spot detections, configure detection distances, and receive alerts when potential hazards are detected or when the bicycle is being approached in AFB mode.

The mobile application will interact with a DynamoDB database that stores the cyclist's configuration settings, detection data, and alert history. Once the STM32F103C8T6 microcontroller has collected the necessary sensor data, it will send this data through a cellular module (e.g., SIM800) to AWS IoT Core using MQTT. The STM32 will be equipped with the AWS SDK and will initialize an IoT client to publish data to an IoT topic in JSON format over the cellular network.

Key Steps and Components:

1. AWS IoT Core will receive the data transmitted from the STM32F103C8T6 through the cellular module. The data stream will include blind spot detection results and alerts about potential hazards.
2. AWS IoT Core will trigger an AWS Lambda function that processes the incoming sensor data and stores it in Amazon DynamoDB. This function will also generate notifications and alerts, which will be sent to the mobile app.

3. Amazon Cognito will handle user authentication, ensuring that each cyclist’s data and settings are securely stored and retrieved. The mobile app will authenticate users through Amazon Cognito to access real-time data, alerts, and configuration options.
4. AWS API Gateway will act as an intermediary, exposing RESTful APIs that the mobile app will use to fetch and configure data. This includes requests to adjust detection distances and settings for both the Blind Spot Detection and AFB systems.
5. The API Gateway will invoke another AWS Lambda function that queries DynamoDB for real-time data on blind spot detections and AFB mode alerts, presenting this data to the user on the mobile app.
6. Amazon SNS (Simple Notification Service) will be integrated to send push notifications to users when the AFB system detects that the bicycle is being approached, or when a hazard is detected in the cyclist's blind spot.

Real-Time Data Streaming and Performance:

To achieve real-time performance and alerts, the system is designed to stream data from the STM32 microcontroller through AWS IoT Core, with near-instant processing by Lambda and storage in DynamoDB. By leveraging serverless architecture, including Lambda and DynamoDB, we ensure that the data processing and notifications occur with minimal latency, typically under 30 seconds for critical alerts and sensor data.

Key Requirements:

- **AWS IoT Core**: Acts as the entry point for sensor data from the STM32 microcontroller, handling MQTT messages.
- **AWS Lambda**: Processes the data, stores it in DynamoDB, and handles notifications.
- **Amazon DynamoDB**: Stores user configurations, sensor data, and alert history.
- **Amazon Cognito**: Manages user authentication and authorization.
- **Amazon SNS**: Sends real-time notifications to the mobile app.
- **AWS API Gateway**: Exposes APIs for mobile app interactions with user data and system configurations.

The system ensures scalability and reliability, with AWS IoT Core and Lambda being able to handle bursts of data from the cellular network. DynamoDB will store data efficiently and is automatically scalable, ensuring the system can handle future growth in user base or detection events.

By using this architecture, the mobile app will provide real-time hazard detection and notifications, giving cyclists the ability to configure their systems and monitor their bikes remotely, ensuring a safer and more informed cycling experience.

Requirements	**Verification**
• The STM32 microcontroller must send sensor data to AWS IoT Core in JSON format within 30 seconds.	• Simulate sensor data from the STM32 and verify it reaches AWS IoT Core within 30 seconds using logs from the

• Test the MQTT message flow using the AWS IoT Test Client to confirm timely data transmission. • Validate data timestamps in the AWS IoT Core message payloads to ensure timely transmission.	microcontroller and AWS IoT Core console metrics.
• The Lambda function triggered by AWS IoT Core should store sensor data in DynamoDB within 30 seconds. • Configure AWS CloudWatch to monitor the time taken by Lambda to store data in DynamoDB and ensure it meets the 30-second requirement.	• Deploy a test Lambda function and simulate IoT Core messages to verify that data is processed and stored in DynamoDB within 30 seconds.
• The cellular module should maintain a stable connection to transmit data from the STM32 microcontroller to AWS.	• Conduct a series of tests with various data loads to ensure reliable transmission from the STM32 via cellular, measuring any connection drops or delays.
• Ensure Lambda function transmits data to DynamoDB that does not exceed 1MB payload size.	• Set up test cases in Lambda to simulate different data sizes and monitor them using AWS CloudWatch to ensure the size is below 1MB.
• Configure the mobile app to fetch and display real-time sensor data and notifications.	• Conduct tests with the mobile app fetching data from DynamoDB via API Gateway, and ensure real-time updates are displayed correctly on the user interface.
• The format of the data stored in DynamoDB must be consistent and valid JSON to prevent errors.	• Validate that data sent from Lambda functions to DynamoDB follows a consistent schema by querying the DynamoDB table and checking data structure.
• Users must be able to configure	• Test the mobile app's user interface for

detection distances via the mobile app and store the settings in DynamoDB.	modifying configuration settings (e.g., detection distances) and ensure the changes are stored in DynamoDB and applied correctly.
• Real-time notifications (alerts) must be sent to the mobile app when the AFB system detects potential hazards.	• Simulate hazard detections in the AFB system and verify that real-time push notifications are received on the mobile app using Amazon SNS.
• The mobile app must authenticate users via Amazon Cognito before accessing the system's data.	• Test the user authentication flow via Amazon Cognito to ensure users are successfully authenticated before accessing system data and settings.

Alerting Subsystem

The alerting system includes two LEDs mounted on the handlebars that will light up based on data received from the sensors to visually alert the cyclist of potential hazards. In addition, a small speaker or vibrating motor connected to the control unit will output a synchronized audible or tactile alert (e.g., beeping sound or vibration) when an object is detected in the blind spot. The alerting system will function both while riding (for blindspot warnings) and when the bicycle is parked (as part of the AFB system). The intensity and frequency of the alerts can be configured through the mobile app to suit user preferences.

Requirements	**Verifications**
• The LEDs must accurately flash when there is an object in the blind spot of the bike (within 3 feet).	• Position the object within 3 feet of the sensor on the left side of the bike and observe the LEDs.

• The LEDs must flash on the side that the object appears in. The flash will be red if there is a detection.	• Then place the object beyond 3 feet and verify the LEDs remain off.
• The system must be able to send notifications to the mobile application when the AFB system is toggled on. • Place an object on the right side and check if only the right-side LEDs flash red. • Simulate both sides detecting objects at the same time and verify that the LEDs on both sides flash red.	• Place an object on the left side and check if only the left-side LEDs flash red.

Power Subsystem

The system will be powered using rechargeable lithium-ion batteries, which will supply 5V to both the sensors and the microcontroller. These batteries are chosen for their high energy density, efficiency, and rechargeable capability, ensuring that the system can operate for extended periods without frequent recharges. An on/off switch will be included to help conserve battery life when the system is not in use, ensuring longer-lasting power. Additionally, the mobile application will provide real-time updates on battery status, including alerts when the battery is running low and needs recharging. We must have a regulator that can provide the 3.3V to the microcontroller.

Requirements	**Verifications**
• The power subsystem must have separate systems for powering the microcontroller and each sensor module.	• Test the separation of power subsystems by isolating the power delivery for the microcontroller and the sensor modules, and ensure no

	interference between the two systems.
• The power delivered to the microcontroller must be 20 mA per I/O pin and 50 mA for the 3.3V pin.	• Measure the current delivered to an active I/O pin and verify that it is 20 mA. Measure the current on the 3.3V pin and verify that it is 50 mA.
• The HC-SR04 ultrasonic sensor has a maximum current draw of 20 milliamps (mA), so the power subsystem must deliver that to each sensor in the module. • Test multiple I/O pins simultaneously under load to ensure the total current delivery remains within acceptable limits.	• Test the power delivered to each HC-SR04 sensor module and verify that the current draw does not exceed 20 mA per sensor.

Tolerance Analysis:

Microcontroller Capability:

The ATmega328P operates at a clock speed of 8 MHz, which provides the processing power needed to handle the ultrasonic sensors' data in real-time.

Sensor Data Rate Calculation:

- Each HC-SR04 sensor sends an echo pulse that requires processing by the microcontroller to calculate the distance.
- The pulse length corresponds to the distance measured (in microseconds) and needs to be interpreted at least 20 times per second per sensor for real-time monitoring.

4 sensors operating:

- Data Rate per Sensor:
  - 20 measurements/second × 2 data points (trigger and echo) × 4 sensors = 160 data points per second.
- Processing Requirements:
  - Each measurement consists of the time for the microcontroller to handle the trigger signal, listen for the echo, compute the distance, and update the alert status.
  - Time for Processing:

        - The ATmega328P typically requires around 1,000 clock cycles per measurement due to calculations, comparisons, and triggering events.

Microcontroller Clock Cycles:

- Clock Speed: 8 MHz = 8 million cycles per second.
- Cycles Available per Measurement:
    - 8,000,000 cycles/sec ÷ 160 measurements/sec = 50,000 cycles per measurement.
- Required Cycles:
    - Only 1,000 cycles are required per measurement, meaning that the microcontroller still has 50x more processing time available than required for each data point. This provides ample headroom for additional tasks like updating the alert system, sending data to the mobile app, and handling AFB system triggers.

I/O and Communication Capabilities:

- The ATmega328P supports multiple communication protocols such which will be used to communicate with the alerting system and mobile app.
- Data Rate to Mobile App:
    - Assuming an alert packet size of 32 bytes, each alert sent to the app occurs less frequently compared to sensor measurements.
    - Even if the system sends up to 10 alerts per second, this would only require 320 bytes per second (2.56 Kbits/sec), which is easily manageable within the communication capabilities of the microcontroller.

Power System Requirements:

- The HC-SR04 sensors require a maximum current draw of 20 mA each, and the microcontroller's I/O pins are rated to handle 20 mA per pin.
- With 4 sensors operating, the power system must provide:
    - Sensors: 4 × 20 mA = 80 mA.
    - Microcontroller and peripherals: approximately 50 mA for the microcontroller + additional load from the alerting system (LEDs, speaker, etc.).
    - Total Power: Approximately 150-200 mA, within the capabilities of a standard rechargeable lithium-ion battery, which provides sufficient operating time.

Ethics and Safety:

In accordance with the IEEE Code of Ethics, we as a group pledge to “uphold the highest standards of integrity, responsible behavior, and ethical conduct” when designing, testing, and building our monitoring system.

We are committed to user safety by ensuring our system enhances safety without introducing new risks. False alerts could cause erratic cyclist behavior, potentially leading to accidents. Rigorous testing under diverse conditions (e.g., weather variations) is necessary to minimize such risks [IEEE Code of Ethics, Section I.1].

Collecting and storing riding metrics requires strict data privacy measures. We will use encryption and provide transparent privacy policies, allowing users to control their data. Ethical responsibility includes protecting user information from unauthorized access and misuse [ACM Code of Ethics, Section 1.6].

Safety Concerns:

The system must operate reliably under various conditions to avoid false or missed alerts. We will adhere to IEEE 1609 standards for reliable vehicular communication and conduct environmental testing to ensure consistent performance [IEEE 1609.0].

EMI can affect system reliability and safety. We will test for compliance with FCC regulations to ensure our system does not interfere with other devices [FCC Part 15].

Using lithium-ion batteries requires compliance with safety standards like UL 1642. We will implement overcharge and short-circuit protection to mitigate the risk of fire or explosion.

Our system will comply with applicable state and federal regulations, such as the Illinois Vehicle Code, to ensure it does not interfere with road safety requirements. We will also follow ISO 4210 standards to ensure the system's structural safety on bicycles