“IOT based Project of Gate Closing for Floods”

Major Project Report

Submitted in Partial Fulfillment of the
Requirements for the Degree of

BACHELOR OF TECHNOLOGY IN

INFORMATION AND COMMUNICATION TECHNOLOGY

By
Shivani Odedra (21BIT267)

Under the Guidance of
DR. Nitin Singh Rajput

Department of Information and Communication Technology,
School of Technology, Pandit Deendayal Energy University,
Gandhinagar 382 426

May 2025

Certificate of Originality of Work

I hereby declare that the B.Tech. Project entitled **"IOT based Project of Gate Closing for Floods "**submitted by me for the partial fulfillment of the degree of Bachelor of Technology to the Dept. of Information and Communication Technology at the School of Technology, Pandit Deendayal Energy University, Gandhinagar, is the original record of the project work carried out by me under the supervision of **DR. Nitin Singh Rajput.**

I also declare that this written submission adheres to University guidelines for its originality, and proper citations and references have been included wherever required. I also declare that I have maintained high academic honesty and integrity and have not falsified any data in my submission.
I also understand that violation of any guidelines in this regard will attract disciplinary action by the institute.

Name of the Student 2: Shivani Odedra

Roll Number of the Student 2: 21BIT267

Signature of the Student 2:

Name of the Supervisor: Dr. Nitin Singh Rajput

Designation of the Supervisor: Assistant Professor

Signature of the Supervisor:

Place
:

Date:

Certificate from the Project Supervisor/Head

This is to certify that the Major Project Report entitled “IOT based Project of Gate Closing for Floods ” submitted by Ms. Shivani Odedra , Roll No. 21BIT267 towards the partial fulfilment of the requirements for the award of degree in Bachelor of Technology in the field of Information And Communication Technology Engineering from the School of Technology, Pandit Deendayal Energy University, Gandhinagar is the record of work carried out by him/her under my/our supervision and guidance. The work submitted by the student has in my/our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for the award of any degree or diploma.

Name and Sign of the Supervisor	Name and Sign of the Industry Supervisor
Name and Sign of the HoD	Name and Sign of the Director

Place

Date

Acknowledgement

First and foremost, I wish to pay my very sincere thanks and gratitude to my project guide **Dr. Nitin Singh Rajput,** from the Department of Information and Communication Technology, Pandit Deendayal Energy University, for his guidance, constant motivation, and unfailing support through this project. His vast knowledge, insightful suggestions, and well-thought criticisms at every stage of the work helped in shaping it as well as improving its quality. This project could not have reached its successful completion without his expert supervision, timely suggestions, and great patience.

I am also greatly indebted to the **Head of Department, Dr Paawan Sharma**, and the other faculty members of the Information and Communication Technology department for providing the resources, infrastructure, and academic support that made the successful completion of this project possible. Their encouragement and useful suggestions during seminars and reviews were vital in fine-tuning my work.

I would also like to extend my sincere thanks to the technical and administrative staff of Pandit Deendayal Energy University for their co-operation in timely arrangement of equipments, software, and other necessities during the course of this project.

My heartfelt thanks go to my classmates, seniors, and friends for their encouragement, stimulating discussions, and constructive criticism. These companions stood by me in difficulties and moments of joy

Above all, I owe a very great debt of gratitude to my very dear family, parents, and siblings, who have bestowed upon me unconditional love, infinite patience, and continuous encouragement throughout my life. Their sacrifices and faith in my abilities have really been the fuel that motivated me to complete this project to the best of my abilities, and attain whatever my aims are.

Shivani Odedra

Abstract

In a time of increasing climate uncertainty, flood prevention is as critical as maintaining infrastructure against resilience. Hence, many bridges do not have automated safety devices that can react to an ascent in water levels. Gate closing for floods is an IoT automated barrier control solution designed and erected to maximize public safety by enabling intelligent management of access over bridges during flood conditions. It uses stormwater ultrasonic sensors that provide water-level measurements in real-time; PIR sensors to detect the presence of humans or pedestrians; and load cells to detect the presence of a vehicle on the bridge. In general, once the flood threshold has been reached and the bridge has been cleared of pedestrians and vehicles, its motorized gates will be activated automatically to block access and avert accidents or damage to the bridge.

The sensor network feeds data to the central microcontroller, Arduino or Raspberry Pi, which processes all sensor inputs through a set of rule-based logic: only when the water level exceeds the safe threshold and no pedestrians or vehicles are detected on the bridge are the gates closed. The user dashboard also notifies authorities of incidents, allows users to see gate status in real-time, and records historic flood data into a MySQL database. Role-based access control scheme enables secured control to administrators for interventions and read-only visibility to maintenance staff.

Considering modularity, the hardware-software architecture emphasizes scalability to incorporate possible futureifications: weather API-based prediction, GSM-based emergency alerts, or solar operation. The coupling of inexpensive IoT-based equipment with fail-safe mechanical design signifies that this project offers a level of smart automation to avoid flooding in vulnerable infrastructure locations. The long-term goal is the deployment of adaptive thresholds through AI and its integration into city-wide disaster management

INDEX

LIST OF FIGURES

NOMENCLATURE

Term	Description
API	Application Programming Interface: Enables communication between the flood gate system and external services such as weather APIs.
Arduino	Microcontroller units that carry out processing of sensor data and engaging the gate actuators accordingly.
CSS	Cascading Style Sheets: Used for styling the web dashboard for monitoring flood alerts and status of the gate
GPS Module	Global Positioning System: Used to track bridge location for regional flood risk assessment.
GMS Module	Global System for Mobile Communications: Used to send SMS alerts to authorities in case of flood incidents.
HTML	Hypertext Markup Language: Specifies the web interface for system monitoring.
Iot	Network of sensors and actuators for real-time flood response.
Load sesnor	Measures weight of vehicles on the bridge to check clearance before the gate closes.
My SQL	A database management system that stores logs of flooding events, sensor data, and user access records.
PIR	Passive Infrared Sensor: Detects presence of humans on the bridge in cases of safety checks.

RBAC	Role-Based Access Controls: Controls access to the system (e.g., admins vs. maintenance staff).
Ultra-sonic sensor	Measures water level elevation beside the bridge supports to sound flood alerts.
Servo motor	A device that generates precise force and torque to deliver the angular motion to flood gates, care of sensor wheedling.

Chapter 1
Introduction

1.1 Introduction

Flooding is one of the most common and catastrophic natural disasters in the world. Millions of people will be displaced due to flooding yearly, which comes along with damages to transportation infrastructure, especially roads and bridges. Flooding threatens public safety and infrastructure due to climate change prompting more erratic and extreme rainfall. Flooding poses an even greater threat to surrounding infrastructure, as heavy precipitation can lead to rapid and catastrophic changes, particularly for roads and bridges built in lower-lying areas or adjacent to rivers.

Historically, closing a bridge during times of high water has relied on human activity. Where automatic or manual water depth gauges are present, it is often left to individuals to either visually monitor the water level or even install basic water gauges to allow individuals to simply have someone physically read the gauges. Even in smarter parts of the world where electronic flood sensors are available, there is heavy reliance on human operators and centralized operation control rooms to close access to the bridge or alert the public to safe conditions.

The delay in response times creates a very serious safety risk. In the case of flash flooding or rivers rapidly rising, people using vehicles and pedestrians may still be on the bridge when it is unsafe. In the worst-case scenarios, this can result in deaths, property loss, and structural failure of the bridge as a result of overload from vehicles and erosion of supports.

With the advent of smart infrastructure and Internet of Things (IoT) technologies, there is an opportunity to automate flood responses in a way that is cost-effective and scalable. There can now be a system of sensors, controllers, and mechanical systems resulting in the ability to design solutions that can monitor flood conditions and respond in an automated way—identifying flooding conditions and closing a bridge to traffic until conditions are safe again.

This project intends to fill that void by developing a prototype system that employs sensors and automation to collect real-time readings on rising water levels and to automate the restriction of access to the bridge, by closing a physical barrier. It also adds a critical layer of intelligence: before executing the barrier mechanism it first checks whether there are any humans or vehicles on the bridge, minimizing the risk of trapping or hurting the user.

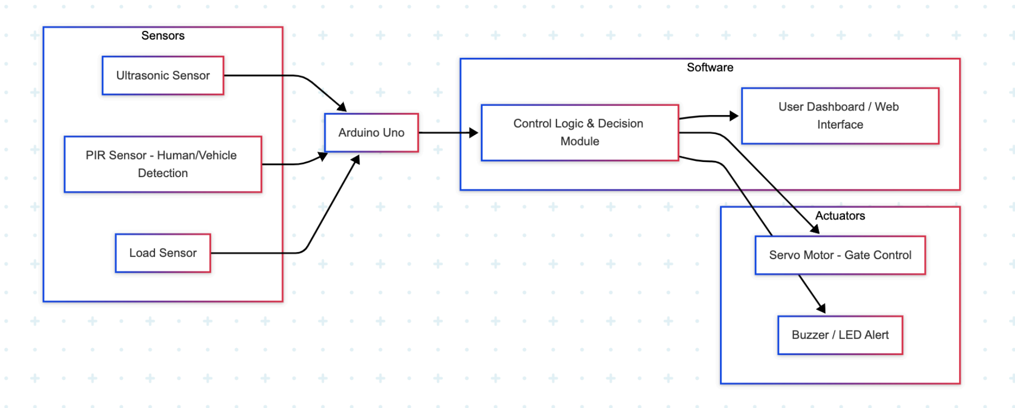


Fig 1.1 Architecture Design

1.2 Motivation

The motivation for this project comes from the increasing prevalence and severity of flooding around the world, especially in developing parts of the world with limited infrastructure or where the implementation of real-time disaster response systems is likely limited. Climate change has exacerbated rainfall patterns to the point where we routinely experience urban flooding. This is a significant operational issue affecting our road and bridge infrastructure as well as disrupting day-to-day living and lives.

Bridges can be the most compromised structures during flooding—one of the most important components of our transportation network. However, having been impacted by the recent increase in flooding and the previous lack of automated flood detection or closure regimes, very few bridges in Canada have automated flood detection and closure measures in place. In further support, bridges in rural and semi-urban areas generally have limited monitoring, often without access to any resources that may allow for the manual closure of the bridge before the water rises. Too often, the manual closure of the bridge occurs too late; human operators only have perceived threats to infrastructure, not impacts to infrastructure at transportation networks in spatially varied locations. When a human operator identifies the imminent disaster scenario, the bridge may be vulnerable or already compromised, causing vehicles to either be submerged or the infrastructure to collapse.

The instances of bridges collapsing during flooding in places like India, Nepal and Bangladesh—where flash floods arrive with little warning—call for low-cost automated systems that can operate with minimal human involvement. Many of these tragedies could have been avoided if an alarm system could have recognised the risk and closed the access to the bridge in time.

Another significant benefit was being able to use applied electronics and IoT knowledge/practice to create an actual solution to a real problem. Students and engineers tend to learn about sensors, microcontrollers, automation, logic design, etc. on a theoretical level. This project allowed combining some of those components to formulate an actual solution that can be created, tested and validated. This project is also a small part of the bigger story of the smart cities initiative and disaster-resilient infrastructure, and harnessing low-cost modular scalable technology to improve public safety.

Moreover, this project was also about the upgrading of technologies used by local governments, especially small towns and villages, to allow them to supplement their own technologies, and use that to update their infrastructure more intelligently (as much as they wish within budget limitations) without requiring changes to a larger centralized system. The inclusion of flexible models created a means in which individuals could recreate these models in different environments without facing challenges and hopefully contribute to making flood safety more effective in urban and rural landscapes.

Additionally, existing flood barrier systems are not mindful of human safety. Most automation simply activates a mechanical behaviour that raises (or lowers) a gate (or some other device) to block access. These solutions do not check whether there is currently anyone or vehicles on the bridge. This is a glaring safety risk—closing a barrier could lead to panic, injuries, or even entrapment if someone is already crossing the bridge. A truly smart system should not only detect a flood risk but guarantee that the bridge is clear before an action is undertaken.

1.3 Objective

The goal of this project is to create a fully functional real-time automated bridge closure system that monitors water levels in the event of flooding and keeps the public safe. Specifically, the objective is to create a system that can autonomously monitor the approaching dangerous levels of water with an ultrasonic sensor and close a barrier to restrict access to the bridge and potentially avoid accidents and damage to the bridge which may occur if the bridge is closed during a flooding scenario.

The unique feature of this system is the inclusion of a detection mechanism for either a human or vehicle presence. The system detects this presence and verifies that there is not a person or object still present on the bridge before closing it. This is important in that it greatly reduces the chance of trapping or injuring somebody. Adopting a safety-first strategy is important in an emergency context where automated systems must rely on time pressure to produce fast responses in unison with and without the necessity of real-time awareness.

The project strives to maintain the system as cost-effective and accessible as possible, utilizing systems with off-the-shelf components, like the Arduino Uno board, ultrasonic sensor, IR or PIR sensor, servo motor actuation, and inexpensive warning systems like buzzer and LED. That way, the system can be deployed in urban and rural locations, especially in areas where advanced flood monitoring systems may not be possible because of the cost.

It is also important that the project may be able to simulate the logic and behaviour of the system with the TinkerCAD, an online simulator for electronics, before moving on to the hardware. We do this so we are not wasting time during that development by designing a system with problems which we could have corrected in the design phase of the software.

1.4 Problem Statement

Bridges are an important piece of the transportation infrastructure, and they are also among the most flood-prone, especially in places where heavy rain and rapidly surging waters pose a risk. Currently, most bridges are monitored manually and they implement manual closures during flood events. The reliance on human observation can, and often does, significantly delay the reaction time required for flooding and dangerous weather events. This not only increases the odds of accidents but may also lead to damage to vehicles and even loss of life. While there are no real-time automated flood detection systems; flood notifications are often issued too late. Most times, travellers and pedestrians are already in an inherent state of peril.

More importantly, most automated systems detecting flood hazards only identify the unsafe conditions, or flood danger, and attempt to improve the water height and cross-section without any safety check for the presence of people and vehicles on the bridge. If the bridge barrier closes while people or vehicles are attempting to clear, accidents and entrapments could occur. This design flaw is significant in part because this safety problem is the major contributor to many existing approaches being classified as impractical in real-world applications, particularly in emergency situations where time is of the essence. The current systems lack presence detection capabilities and therefore cannot integrate flood detection and presence detection capabilities to promote timely precautionary measures and user safety with response time.

In addition, advanced flood monitoring technologies are typically costly and complicated therefore, small towns, rural areas or developing countries with limited investments in infrastructure are often neglected. A vast amount of locations do not have an affordable, easy-to-install, and low-maintenance solution that can automatically monitor bridge closures, safely, and effectively. This project is intended to overcome these challenges by developing a low-cost real-time automated bridge closing system that combines real-time water level monitoring with real-time human detection to protect infrastructure and the safety of the public.

1.5 Approach

The strategic accumulation of hardware integration, software design, and experimental tests represent a flexible option to build a dependable and real-time flood response device. The approach applies a systematic analysis of the system requirements and the identification of sensors and actuators able to search the environmental conditions and physically control access to the bridge.

First, the system logic was designed in Tinkercad, which enables designers to program a microcontroller and develop the interaction between sensors and actuators in a simulated environment. In this way, we were able to rapidly test, modify, debug, and prototype the main components of verifying water levels, avoiding human traffic, and controlling a barrier while avoiding damage to hardware. It was noted in the design software that the modularity of the software would help modify the flow of functions and limit barriers to future iterations.

Once the simulation procedure has successfully completed, we will produce physical hardware solutions in the design of a prototype, that uses the Arduino Uno Microcontroller, ultrasonic sensors, PIR sensors, servo motors, and alarm systems/devices such as buzzers and LED. The hardware will be wired and integrated to mimic the real world as realistically as possible when monitoring for a flood, and activating the bridge barriers. Each sensor used in the final design will need to be tested, calibrated, and used to measure the exact readings and accurate response times from the actuators. The system's decision logic will ensure safety, by allowing presence detection of pedestrians and/or vehicles before gate closure action takes place when the bridge is occupied. By following a stepwise, bottom-up design approach, this process will develop a low-cost product that is heavy-duty and scalable within its own communities that are vulnerable to flooding.

1.6 Project Scope

The scope of this project concerns the design, development, and testing of an automated bridge closing system that will monitor the water level beneath the bridge during flood conditions in real-time and safely operate, closing the bridge if the water level rises too high and detecting someone — human or vehicle. The scope of this project will only include the development of a single prototype model that recreates

the main components of a bridge monitoring and control system on a smaller scale based on microcontroller platforms and simple/relatively inexpensive sensors.

This kind of system will incorporate an ultrasonic sensor that will measure, in real-time, the depth/level of water under the bridge and a motion detection sensor (PIR sensor or infrared detection) to confirm that there are no pedestrians or vehicles present before closing the bridge gate. The actual physical closure of the bridge gate is accomplished by using a servo motor controlled by Arduino (or another microcontroller). There will be visual/audible alerts to signal users of bridge closure during unsafe conditions.

The project scope does not include thorough deployment on real bridges, advanced remote monitoring, and communication features (GSM, internet, etc.). We believe these features are enhancements for the future. Also outside of our scope are advanced detection methods (camera monitoring, AI-based recognition systems, etc.). We will try to stay with a cost-effective, reliable, and safe system now, which could be added to later (and scaled) for actual use in flood-case scenarios (i.e. - high-risk zones).

Chapter 2
Literature
Review

Bridge safety is an issue of relevance in contemporary infrastructure management, especially concerning ageing infrastructures and growing environmental threats. Several studies have investigated the convergence of the Internet of Things (IoT), wireless sensor networks, and automation control technologies for improved real-time monitoring and alarm systems. This review of literature accounts for earlier studies on IoT-based bridge monitoring systems, wireless sensor-based structural health monitoring, flood and overload detection systems, and automation implementation towards proactive safety. The studies as a whole record the utilization of sensors such as ultrasonic, vibration, and weight sensors, microcontrollers, and wireless communication modules for the detection of structural health and the implementation of emergency actions such as barrier shutdowns and alarms.

1. Bridge Safety Monitoring System using IoT

This paper puts forth a new IoT-based system to assess the safety of the bridges over different phenomena while providing real-time monitoring of accessibility for both users and bridge management. This system continuously monitors a bridge's operational conditions including but not limited to vibration, water level, and vehicle load. By utilizing a wireless sensor networking paradigm, which consists of wireless sensor nodes, this information is collected and delivered via Wi-Fi to the nearby monitoring station, which can be used to gain knowledge about the accessibility of the bridge. The dual nature of data collection and eventually real time reporting of the conditions allows for immediate accessibility decisions by consultants, users, or management as necessary in some instances potentially rescuing management before catastrophic effects set in.

This paper embodies automation of safety issues, barriers or actuators like servo motors connected to barrier gates can actively control the accessibility of a bridge upon hazard observation. inclusions of safety elements like no access bars or signs can be physically observed preemptively. Bridge management, users, and consultants will all be notified accordingly, dispatching help or essentially rapid response to hazards as previously noted.

Lastly, this proposal is based on IoT Service Models that can be inexpensively developed using existing process frameworks. Any bridge can be dust cleaner without having to even contemplate complete redesigns of the existing structure and compliant design from the legislated establishment or as we do in environmentally Evaluated Systems of Processes (S.Yn.C.O.). plus, the system represents a modular design to add functionality or reporting integration to any mobile monitoring or even web-based only accessible monitoring station. Ultimately, the IoT safety assessment service protocol can be expanded greatly for all static structures.

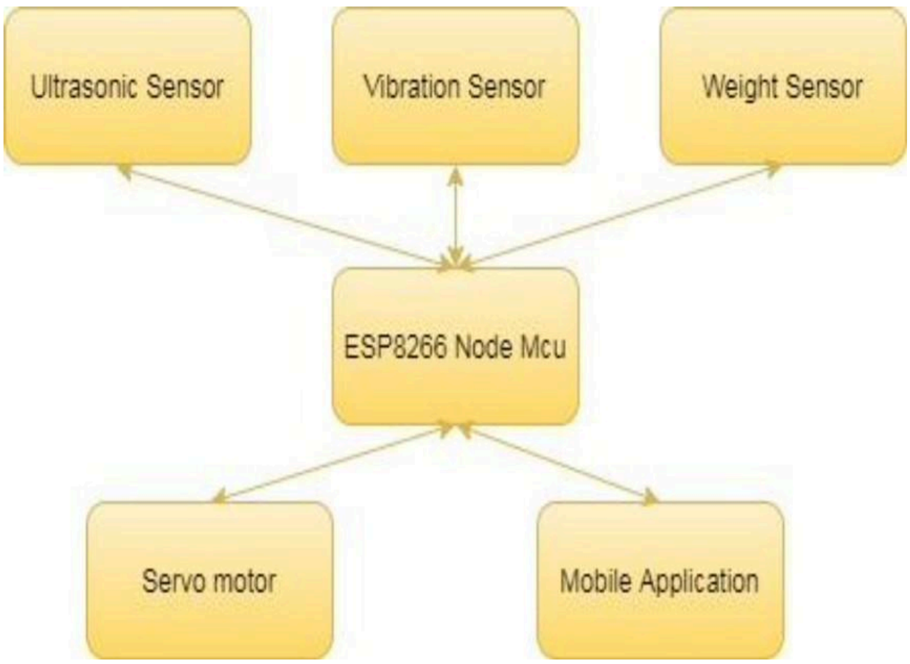


Fig 2.1 Architecture Design

2. Automatic Bridge Monitoring System Using Wireless Sensor Network

This research considers the implementation of a wireless sensor network using a ZigBee radio frequency (RF) communication protocol to monitor the health of a bridge in real-time. This research presents not only the integration of wireless sensors but a system that includes several different types of sensors, such as accelerometers, temperature gauges, humidity sensors and a strain gauge, that are used to monitor behaviour indicators of structural integrity (e.g., vibrations, displacements, environmental effects, etc.). The systems development allows the wireless sensor network to gather structural health data and transmit the data to a nearby server via RF communication and data storage.

Energy efficiency is an important part of the system by ensuring the sensor nodes have a low-power operation scheme, which is important for the long-term use of the wireless sensor network in remote locations or in difficult-to-access remote bridges. The wireless sensor network architecture also supports the generation of an alert that is based on threshold readings from the sensors, which will allow for proper warning if the sensors are reading a level of concern for damaging effects or for stresses on the bridge exceeding normal levels.

This would allow for a more continuous, remote, and ongoing type of inspection, thus, manual inspections would take much less time after the network is established and data may be sent to maintenance teams in advance of final assessments. Because the wireless sensors use low-power RF communication, the wireless sensor network is a viable proposition for structural health monitoring of virtually any type of bridge, unlike in less developed parts of the world.

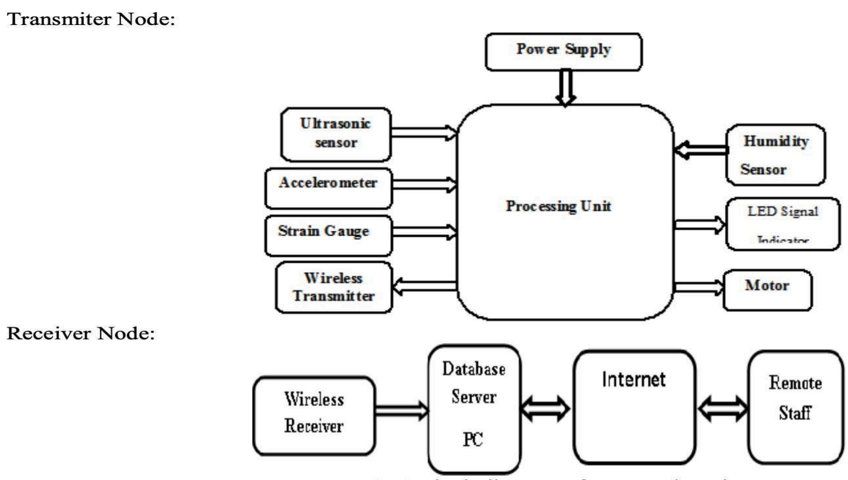


Fig 2.2 Architecture design

3. Real Time Bridge Monitoring and Alert Generation System Using IoT

The present study proposes an IoT-based approach for monitoring ageing infrastructure such as bridges that need constant monitoring to avoid failure. Weight sensors are used for initial monitoring of traffic load while water level sensors are used for monitoring flood risk. The monitoring sensors along with microcontrollers continuously collect and process data and can generate alerts when there is a certainty during threshold alert conditions.

Alerts will be turned on using buzzers and automatic barriers to limit roadway access by an impending driver, thus enforcing public safety measures. The main weakness of traditional (huMan) monitoring is that we cannot measure or sense constant changes in the environment and decouple human factor fatigue to preserve life and property. For example, with this approach to monitoring bridges using the proposed 'sensors' we can measure immediate response following sudden changes like rising flood levels, and/or traffic load without human assistance.

The strength of the proposed system relies on easy-to-access instrumentation and communication technologies available for low-cost solutions. Overall the proposed technology offers a preventative safety approach to infrastructure monitoring to detect hazards and signal automated action in real-time as opposed to needing a human operator to act. The technology proposed will remain important and useful for any person, practitioner or region looking for solutions for ageing infrastructure.

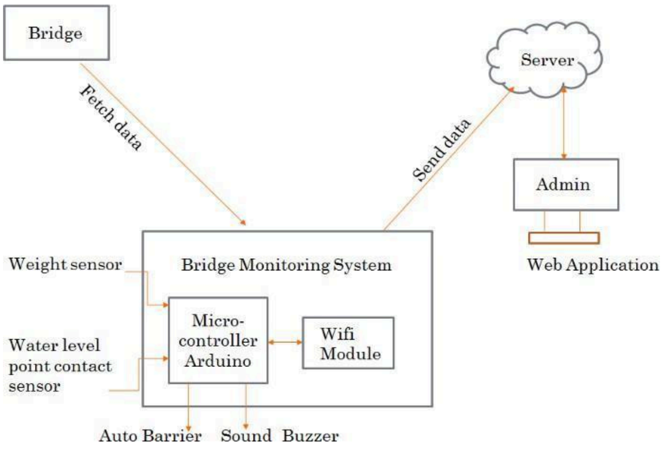


Fig 2.3 Architecture Design

4. Bridge Safety and Flood Detection System

This project establishes an all-encompassing safety monitoring setup that will integrate several sensing technologies to identify the state of bridge conditions when a flood threat arises. Ultrasonic sensors will monitor water levels to identify a flood that is rising, the suspension wire mechanisms will identify cracking or structural damage, and load cells will assess the weight of the vehicle crossing the bridge. The sensors will complement one another; enabling a multidimensional picture of the bridge's health.

The system contains a GSM communication technology component that allows alerts and barricade management to be done remotely thereby greatly improving local authorities' ability to respond in time for imminent danger. By combining structural health monitoring and flood detection, the system attempts to stop damage and accidents to structural failures resulting from environmental and mechanical failures.

The design of this system is modular and allows any number of additional sensors or communication technologies to be added. As such, it can be used for any bridge configuration and for any geography. The integrated nature of the system will thus allow a more complete understanding of what is required from a safety perspective based on multiple risk factors.

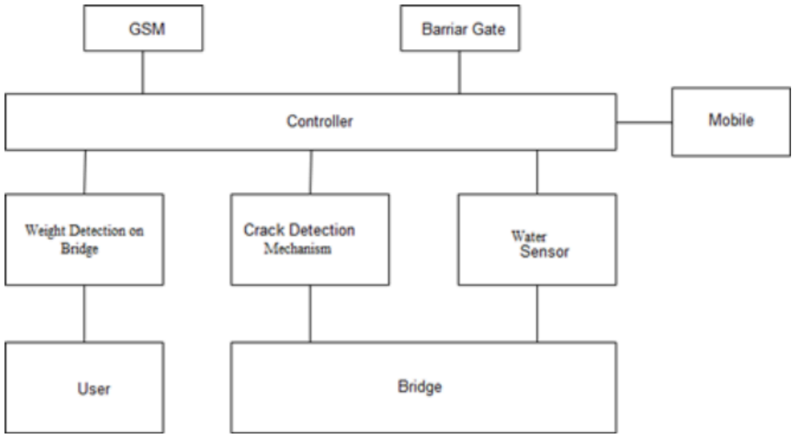


Fig 2.4 Architecture Design

5. Automatic Stop Gate at Bridges for Overflow

This study suggests a flood management approach that employs Arduino microcontrollers and sensor inputs to automate the closing of bridge gates. Water level sensor selectivity constantly monitors the environment and opens the barrier gates when the flood water exceeds a specific height relative to the predetermined safety limit. The system will also include an IR sensor to detect a vehicle or human being within the gate, meaning it also would not be able to close if there was a vehicle approaching with flood water present.

For system communication, there are options for indirect communication utilizing GSM modules to warn local authorities or enable the system remotely to open and close the gates. The system will also use tactile switches for local responders to control the system in a manual operation function if required. Overall, converting floodgate operations from a manual to a computerized automated solution increases the reliability of the system and helps explain to the user confidence in the system's function.

Compounded, the flood management system will be cost-effective and simple to finalize, for local respondents in areas that use standard infrastructure will not overextend the expense. The functionality in real-time will have the aforementioned designed safety precautions in mind.

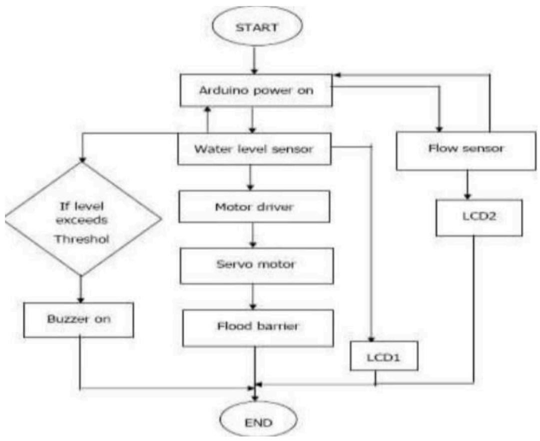


Fig 2.5 Architecture Design

6. Research Gaps and Opportunities

While previous research demonstrates great strides in IoT and sensor-based bridge safety-monitoring systems, there are still some significant gaps to be filled. Most systems developed so far are focused on monitoring environmental parameters such as water level, or structural parameters such as vibration and load, rather than cross-domain monitoring of various parameters. There is also limited multi-parameter sensing found in practice, such as flooding detection with verification of human or vehicle presence in the exclusion zone, to confirm safety before actuation of the barrier.

Another gap identified in the research is the absence of real-time predictive intelligence decision-making algorithms that can predict a potentially hazardous situation or failure condition rather than reacting after thresholds have been surpassed. Most designs today utilize threshold-initiated triggers for action without featuring advanced data analytics or predictive modelling for an early warning.

In addition, many systems today do not have key safety features such as verification that the bridge is clear of pedestrians or vehicles before closure. The use of barrier actuators to close down heavily trafficked bridge points can pose an immediate safety risk which can easily be addressed through common human presence detection gumptions using more sophisticated detection systems, such as camera-based or thermal sensors with integrated AI systems.

There is, from a practical perspective, a demand for cheaper, modular and scalable solutions to deploy in rural or resource-stretched locations. The majority of advanced systems are expensive or involve complex infrastructure and the potential for adoption is limited.

- Various places for future research and development exist, including:
- Multi-sensor systems that bring together flood detection, structural health monitoring, and presence detection into one accessible system.
- Predictive analytics and AI analyzing datasets to predict dangerous conditions in advance and hope for (and respond proactively).
- Advanced human and vehicle detection techniques can ensure the safe operation of automated barriers.
- Low-cost, low-power, scalable systems for use in rural communities and developing countries.
- Integration of wireless communications for real-time, remote monitoring and control using IoT cloud platforms.

It is hoped that if such gaps were addressed then future systems would improve bridge safety during floods, make human risk negligible, and provide reliable protection to engineering infrastructure available to more communities.

7. Challenges

There are a lot of factors to take into account when designing and building an automated bridge closing system. One of the most significant obstacles is achieving accurate and reliable inputs from each of the sensors, especially when facing adverse weather conditions such as heavy rain or debris hitting ultrasonic and motion sensors. Therefore the system needs to rapidly and accurately identify and assess changing flood conditions, while weighing sensitivity to identify individuals or vehicles when closing the bridge, to have a quick response to conditions to avoid dangerous flooding conditions and to avoid false triggers.

In particular, the automated bridge closing system needs to consider personal safety. It must be a reliable detection of a human or vehicle present while the bridge is closed so that no needless injuries or fatalities occur. The bridge closing system must be efficient and operate under limited power, or the communication may be affected by the location of a bridge. We need a power-efficient, yet stable, design to operate reliably under emergency conditions.

The result of an automated bridge closing system is to combine several pieces of hardware into one scalable system at a minimal cost. There are many things to address to be able to create an automated flood-responsive bridge safety system that the public relies upon.

Chapter 3
Software
Design

The software design of the automatic bridge closing system is the secret to its real-time response and reliability. The system is designed to sense rising water levels via sensors and automatically close an electric motor-operated barrier so as not to allow cars or people to drive over a flood-endangered bridge. The parts and rationale were initially simulated using Tinkercad, an internet-based computer software which facilitated simple circuit designing and programming of Arduino code. Simulation enabled the system to operate as desired under various conditions before implementing it in the hardware model.

The system works by repeatedly reading the readings from an ultrasonic sensor, used to measure the volume of water from the water level to the sensor placed on the bridge. When it reads a value over a preconfigured safety level, Arduino Uno stores the reading and requests a servo motor to close the bridge gate. As the second safety feature, at the same time, a buzzer or an LED light is also switched on to inform neighbouring users. Once the water level drops below the critical level, the barrier again switches back to its original open position for regular use.

3.1 System Structure

The creation of the automatic bridge closing system architecture is driven by a modular and event-driven embedded control architecture that allows for real-time monitoring and automated actuation of the bridge's closing operation. A modular program architecture has been deployed into an embedded microcontroller acting as the system's (CPU) central processing unit. The microcontroller provides all interfaces, data processing as well as control of all actuators within the system.

Input Layer:

The input layer has an ultrasonic sensor continuously sampling the water level flowing beneath the bridge and a human presence detection sensor (e.g., PIR or IR sensor) which detects either a pedestrian or vehicle on the bridge deck. The two sensors provide real-time environmental data that can be represented in control logic.

Processing Layer:

The processing layer is comprised of the Arduino firmware - which implements the control algorithms. The software takes raw sensor data, filters for noise and compares the measurements to acceptable safety thresholds. In addition to processing, the software provides for human presence detection so that the bridge deck can be verified to be free of pedestrians or vehicles prior to commencing the closing actuation.

Output Layer:

The output layer activates or deactivates output components, depending on the decision made in the processing layer. This action consists of controlling the servo motor and physically closing or opening the bridge barrier, in addition to triggering one or more alerting devices (such as buzzers and LEDs) meant to inform the nearby users about the status of the bridge.

Communication and Interface:

Even if the prototype at present is predominantly designed to function in a standalone mode, the architecture is such that wireless communication modules can be inserted later on to allow remote monitoring and control.

The architecture follows a cyclic pattern of sensing and control: sensor data are read, then processed, after which actions are taken, and so on, continuously, thereby ensuring that the system is always responsive to any changes in the environment. This layered, modular design promotes scalability, maintainability, and real-time capabilities.

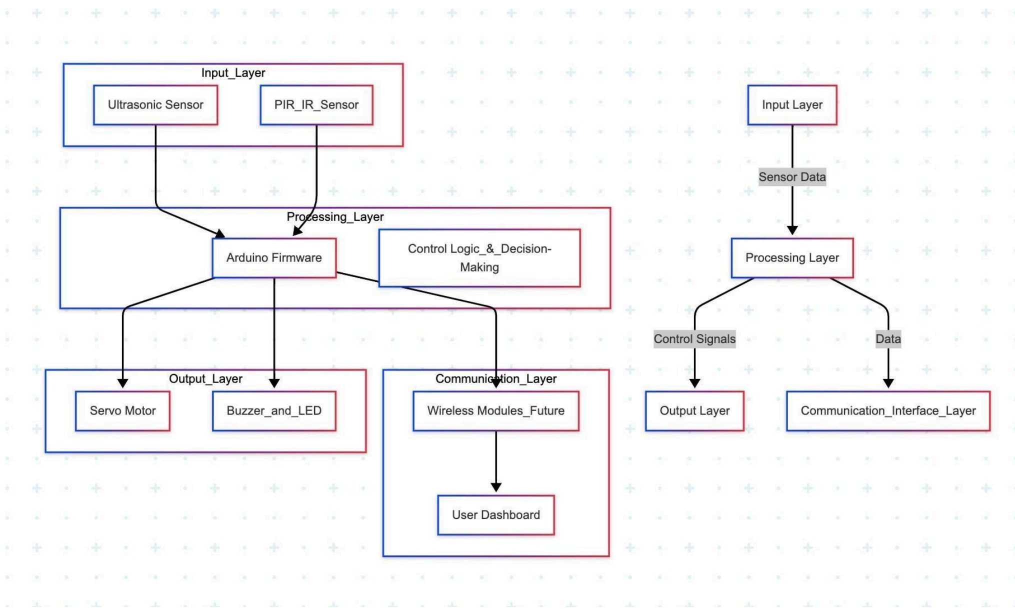


Fig 3.1 Flow chart of Software design

3.2 Logic Flow

The logic flow of the automatic bridge closing system defines the sequential decision process that the microcontroller carries out to determine the correct detection flood conditions and operate the bridge barrier in a safe manner. It is designed to monitor the environmental conditions continuously and to react as soon as water levels detect unsafe water levels, while still promoting human safety.

The sequence begins with initializing the system. All the components, including the ultrasonic sensor and any human presence sensor, servo motor, and buzzer/LED alert modules are engaged, and the state is set to their defaults. After initialization, the system will repeatedly operate, performing a pair of readings of the water level measurement from the ultrasonic sensor and the human presence from the PIR or IR.

The control logic compares the measured water level to the flood threshold keeps the bridge open if the level is below the threshold and deactivates the alerts, which allows for safe access. If the water level had previously detected flood and the system must keep the bridge barrier closed, the system must also check for any possible humans, or other objects present on the bridge before it permits the bridge barrier to close.

If the bridge is not clear - in other words, if we detect someone or something on the bridge - then, we will delay closing the barrier until the bridge is clear. This way we can prevent anyone or anything from being trapped or injured by the barrier. In fact, the only time when we will actually close the barrier is when it is confirmed that the water level is dangerous, and the bridge is clear. At that point, the system will execute two actions: 1) notify users nearby by activating the buzzer or LED alert, and 2) actively close the barrier gate using the servo motor. We can totally allow the system to continue monitoring in a loop. If the water level drops below the threshold, the system will simply reopen the barrier and turn off alerts.

This event-driven logic means we make decisions dynamically based on input, so we do not need to actively decide. This reduces the likelihood of error from human decisions as well, given that we are programming the system to logically ensure next steps are performed based on determined conditions.

The logic we applied is modular and stacked which makes it relatively easy to change thresholds for opening and closing, use more than one sensor, or even adapt the system for varying environmental conditions. What we developed is considered event-driven programming in the Arduino IDE which is structured in a way that is completely logical and can be executed in a conditional check and loop-based execution based on monitored conditions ensuring safety and responsiveness at all times.

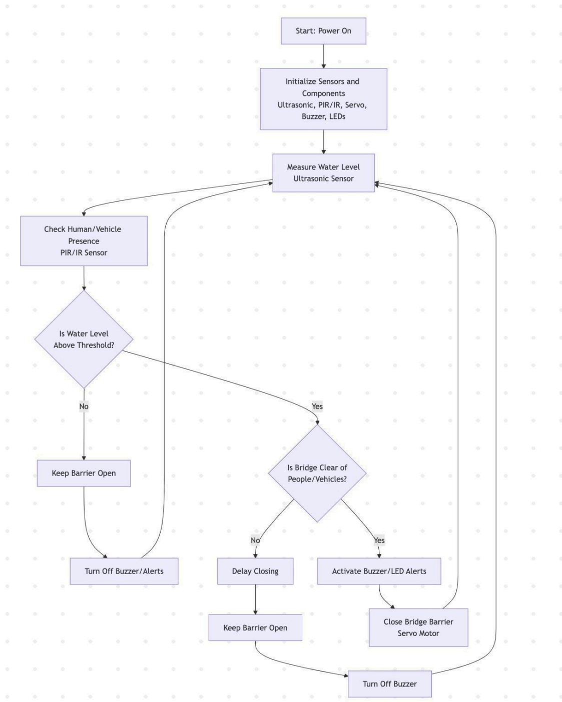


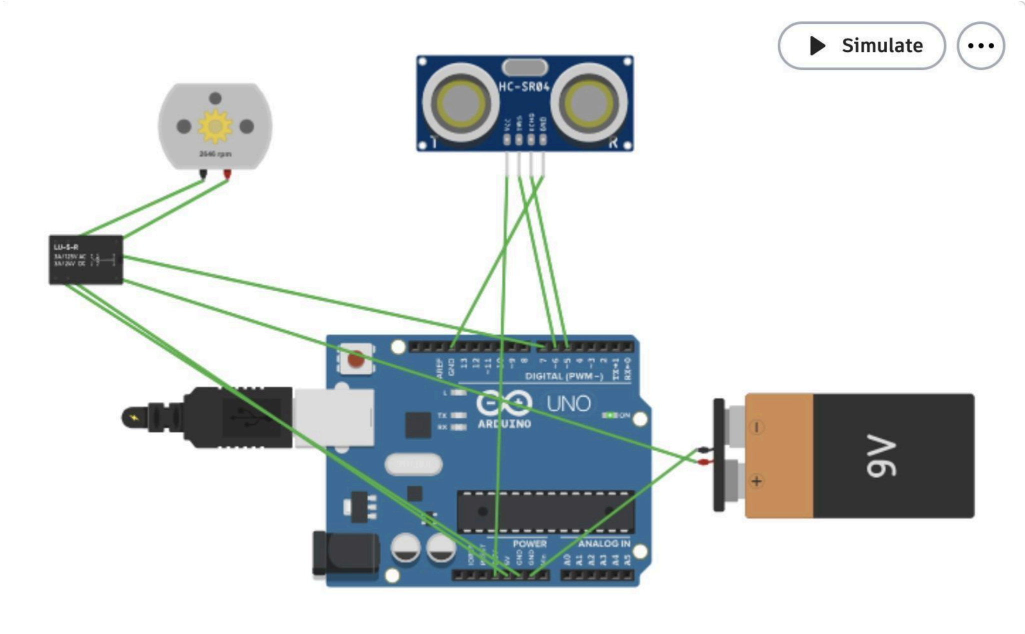
Fig 3.2 Logic Flow

3.3 Tools Used

The automatic bridge-closing system required the selection of several hardware and software tools for simulation, coding, testing, and implementation. Each tool was essential in developing the project from the design stage through to the physical implementation. The project team selected the tools with regard to availability, cost, ease of use, and compatibility with hardware platforms commonly found in electronic production, such as Arduino. The following sets out a description of each of the tools used in this project.

Tinkercad by Autodesk

Tinkercad is a web-based 3D modelling and electronics simulation software suite that provides a browser-based 3D modelling and electronics simulation environment developed by Autodesk. Tinkercad was used in the early stages of the project to simulate the full circuit and confirm system logic before implementation. Tinkercad provides an online version of a breadboard interface and drag-and-drop components, including Arduino Uno, sensors, LEDs, and motors. One of Tinkercad's most powerful features is ian ntegrated Arduino code editor and the ability to upload and simulate Arduino sketches in real time.



3.3.1 Tinkercad

Arduino Software

The primary piece of software used to write, and upload code to the Arduino Uno microcontroller was the Arduino Integrated Development Environment (IDE). The IDE supports the C/C++ programming language and has a straightforward user interface, making it appropriate for both novice and experienced developers. The IDE's bundled features such as syntax highlighting, outline error checking, the serial monitor, and library management made it the best option for this project.

The logic for measuring water level, sensing for a human, and controlling the barrier gate was written and uploaded to the Arduino UNO through the Arduino IDE. Libraries such as Servo.h were used to command a smooth motion for the servo motor, and calls to functions for performing ultrasonic distance measurement, as well as digital input from the IR/PIR sensor were written. The serial monitor feature was useful in the debugging stage because it displayed the sensor values and the system's responses to those values in real time.

Microcontroller for Arduino Uno

The Arduino Uno board was the core of the system. It was used for processing inputs from the sensors, carrying out decision-making logic, and sending commands to the output devices such as the servo motor and buzzer. The board is an open-source ATmega328P-based microcontroller board with ample digital and analog I/O pins for small to medium-sized embedded systems.

The Arduino Uno was a perfect fit for this prototype due to its price, reliability, and large user base. It had more than ample speed and memory to receive all programmed logic in a single looping structure that would allow real-time response to the water level changes and presence detection. Additionally, the Arduino Uno can simulate on Tinkercad and can be programmed using the Arduino IDE, ensuring a seamless transition from simulation to implementation.

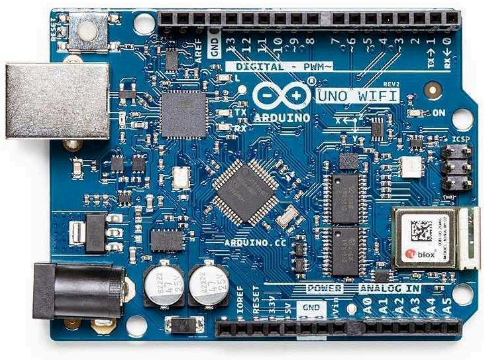


Fig 3.3.2 Aurdino Uno

Sensors and Actuators

Ultrasonic Sensor (HC-SR04): Used to determine the distance from the sensor to the water surface, the ultrasonic sensor enabled real-time water level calculations based on the echo time it receives when emitting sound.

Fig 3.3.3 Ultra-sonic Sensor

PIR or IR Sensor: For the safety of those using the bridge, the PIR and IR sensors were placed to detect the presence of humans or vehicles. The sensors detect motion (PIR) through infrared light that is radiating from (PIR) or through physical obstruction of the light (IR), then acknowledge presence.

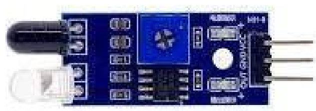


Fig 3.3.4 IR Sensor

Servo Motor: The most important actuator in the project is the Servo Motor. The servo allowed the actuator to control and manage the physical gate. If flooding conditions are detected and the bridge is clear of any presence, the servo rotates and closes the barrier of the bridge.

Fig 3.3.5 Servo Motor

Buzzer and LED: Once the bridge is alerted to close if any dangerous condition exists, the buzzer and LED are activated. The buzzer provided audio feedback to alert pedestrians and cars, and the LED provided a traffic visible indication.

Fig 3.3.6 Fig Buzzer and LED

Circuit Prototyping Tools

The circuit was prototyped with standard prototyping tools such as breadboards, jumper wires, and resistors. The breadboards enabled the easy and non-permanent connection of components and gave the flexibility to test different configurations. Jumper wires enabled secure and organized connections between the sensors, actuators, and the microcontroller.

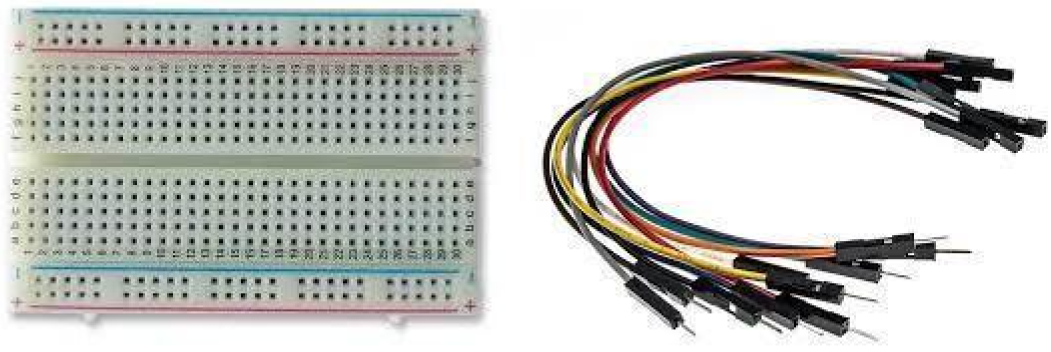


Fig 3.3.7 Breadboard and Jumper wire

Power Supply

Hopefully, during testing, the Arduino Uno was powered through a USB cable connected to either a laptop computer or a portable battery bank. In a real deployment, the system could be powered with a stable DC adapter or a set of solar panels for off-grid locations. A reliable power supply was paramount to ensure consistent sensor readings and stable actuator readings.

Fig 3.3.8 Power Supply

3.4 Software Modules and Libraries

The software design for the automatic bridge closing system is modular to facilitate maintenance, testing, and future development. The design creates a clear, robust software system through the use of modules that perform specific functions. Also, by utilizing existing Arduino libraries, the design reduces development time and risk.

Sensor Interface Module:

The sensor interface module handles all hardware sensors. This module communicates with the ultrasonic water level sensor, as well as human/object detection sensors (for example, IR or load sensors). It turns raw sensor readings into clean, filtered, and determined measurements for the main control logic. This module has the responsibility to accept noise and erratic readings for the ultrasonic sensor, maintain synchronous communication with the ultrasonic sensor, measure echo time, interpret the IR sensor signal and measure distance, and filter out erratic readings from the ultrasonic sensor.

Decision-Making Module:

The decision-making module applies threshold logic to the processed sensor inputs to determine the state of the system. This module compares the water levels with a threshold for critical flooding, and the present status of the humans or vehicles travelling on the bridge or in its proximity. The result of its comparisons determines if the bridge maintains an open status, or goes into a shutdown status to close the bridge. Also, the decision-making module has to manage fail-safe procedures to ensure the sensitized bridge does not attempt to close when there are humans or vehicles occupying the bridge.

Actuator Control Module:

The actuator module deals with controlling the servo motor that actually opens or closes the bridge gate. It takes advantage of the Arduino Servo library to provide smooth and accurate commands to the servo motor and therefore to the barrier and it is doing it correctly and efficiently. The actuator module also controls when to activate or deactivate alert devices such as buzzers and LEDs and coordinates the warnings with the barrier movement.

Communication Module (Future Scope):

While there is no communication module included with the current prototype, it is included in the next version. This module will be responsible for carrying out all wireless transmission of data via Wi-Fi, GSM, or LoRa modules, remote monitoring,

logging, and control of the overall bridge barrier system. If the communication module is included this will increase the possibilities for scalability and information management will be centralized.

Key Libraries Used:

Servo.h: Controls servo motors by abstracting the PWM signal generation code.

NewPing.h (or equivalent): Controls the operation of ultrasonic sensors by managing the pulse triggering and echo timing, providing a means to improve accuracy and timing efficiency.

EEPROM.h (option): This library could store the calibration values or other system states in permanent memory which would be available to be accessed after power cycles.

By defining the software as modules and providing reliable libraries, this project can define maintainability, and scalability and allow for simpler debugging. Additionally, modules allow for new sensors or features to be implemented with minimal impact on the existing functionality.

3.5 Code Organization

The software for the system that automatically closes the bridges is divided into separate functional sections to improve readability, maintainability, and scalability. Each section of the program works on a specific task, such as reading sensor data, interpreting sensor data, activating an actuator, or sending an alert.

Initialization

This section sets up the microcontroller's input/output pins and sets actuators, alerts, etc., to their initial state.

```cpp
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(presencePin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  bridgeServo.attach(servoPin);

  openBridgeBarrier();   // Start with barrier open
  deactivateAlerts();
}
```

Fig 3.5.1 Initialization code

Sensor Reading Functions

Dedicated functions collect data from the ultrasonic sensor (water level) and presence sensor (human/object detection).

```cpp
int measureWaterLevel() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  int distance = duration * 0.034 / 2;
  return distance;
}

bool checkPresence() {
  int presenceState = digitalRead(presencePin);
  return (presenceState != presenceDetected);  // True if no presence detected
}
```

Fig 3.5.2 Data Collection Code

Decision Logic

The system decides whether to open or close the barrier, based on sensors so the bridge can operate safely.

```cpp
if (waterLevel < waterLevelThreshold) {
  openBridgeBarrier();
  deactivateAlerts();
} else {
  if (isBridgeClear) {
    activateAlerts();
    closeBridgeBarrier();
  } else {
    openBridgeBarrier();
    deactivateAlerts();
  }
}
```

Fig 3.5.3 Decision Logic

Actuator Control and Alerts

Functions that move the servo motors and manage alert devices that contain the details of hardware control.

```cpp
void openBridgeBarrier() {
  bridgeServo.write(0);  // Open position
}

void closeBridgeBarrier() {
  bridgeServo.write(90);  // Closed position
}

void activateAlerts() {
  digitalWrite(buzzerPin, HIGH);
  digitalWrite(ledPin, HIGH);
}

void deactivateAlerts() {
  digitalWrite(buzzerPin, LOW);
  digitalWrite(ledPin, LOW);
}
```

Fig 3.5.4 Actuator Control Code

Chapter 4

Component

Design

This chapter will describe how each individual component must be designed, as a component of the automatic bridge closing system. In the component design phase, the right sensors, actuators, microcontrollers and additional supporting hardware must also be selected, along with their specifications, and roles within the system. This chapter discusses how each component works, how they interface with one another, and how they are designed to ensure the system functions well, and reliably, during flood conditions.

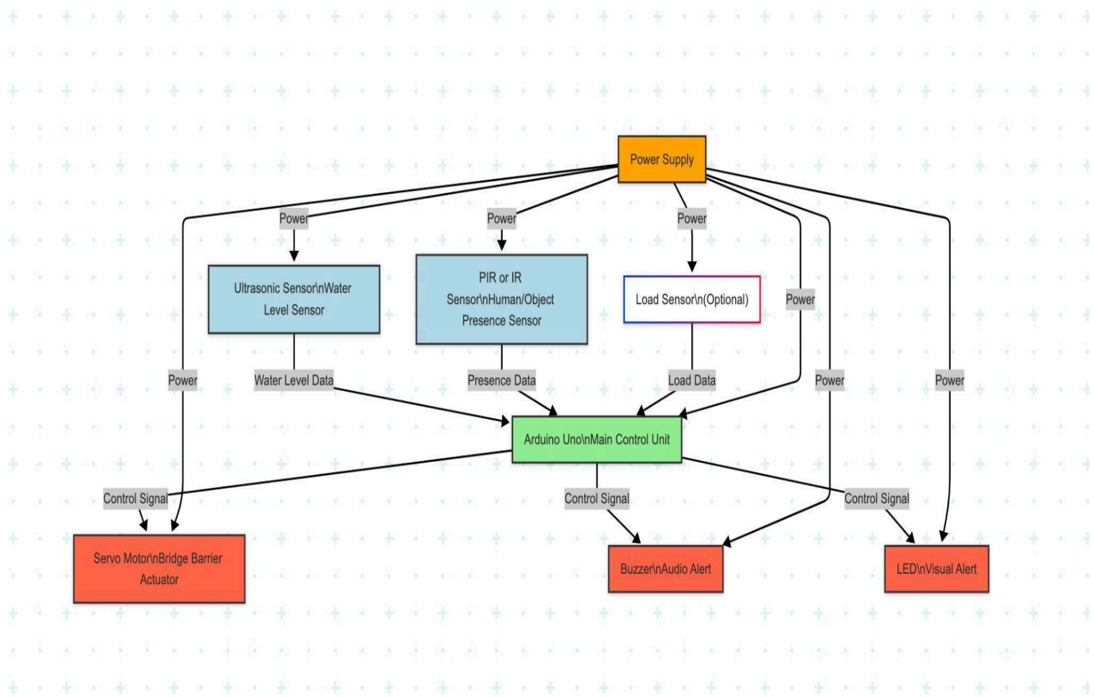


Fig 4.1 Component Design

4.1 Component hierarchy and dependencies

The system for closing the automatic bridge has a clear component hierarchy, where each component has a designated role, and each component relies on others to work together properly. The component at the top of this hierarchy is the Arduino Uno microcontroller, which is the brain of the entire system. The Arduino is responsible for receiving data at each input component, processing the data based on the programmed code logic, and producing control signals for the output devices.

The input component that is primarily used in this system is the ultrasonic sensor (HC-SR04). The ultrasonic sensor will send ultrasonic pulses out and determine how long it takes for echoes to return from the water surface. As swimming water can produce a variety of fluctuating levels; the accuracy and reliability of this component absolutely necessary as the operation of the whole is dependent on the water level crossing the critical threshold.

After the Arduino component, receives this information the component will process this in real time. The programming logic (made by judicious development) allows for a comparison between the measured water height against a predetermined critical threshold for measurement. When the water level exceeds the limit given to it, the Arduino will generate a collection of control signals with the purpose of actuating the output device, which is a servo motor in this example. The servo motor will use the control information in the form of rotational movement to act as a mechanical actuator and rotate to close the barrier of the bridge and prevent access thus making it safe.

Meanwhile, the Arduino manages signals as it alerts the alert subsystem, which has a buzzer and possibly an LED indicator. The alert subsystem of the system can inform others in the area that there is a hazardous condition. The alert subsystem generates audible and/or visual alerts that could increase the safety of the system.

With the main components also systems support like a supply of electricity and connections for communication, to run constantly on electricity and ensure reliable communication between the components. The connectors run in a linear fashion depending on other components as it has been addressed in the systems hierarchy without confusion. The sensor must deliver reliable input to the microcontroller. The microcontroller should exert a reliable effort to trigger the actuator and alert subsystem.

Any single component could be malfunctioning or lagging. In any of these situations, the whole system will not perform effectively. The two main influences as it relates to the

This rigorous yet flexible data policy supports KidDoPay functioning well, efficiently, and securely—qualities of every mechanism that administers user engagement and digital cash transfer. characters of the system i.e., hierarchy, and dependencies take into consideration the design of the system and the reliability and compliance character of the interface protocols. The system was designed and built with a closed feedback loop. The actuators can update following changes in the printed environmental conditions in the system in almost real-time. The characters and dependencies involved assist in showing the automatic bridge closing system as a reliable system.

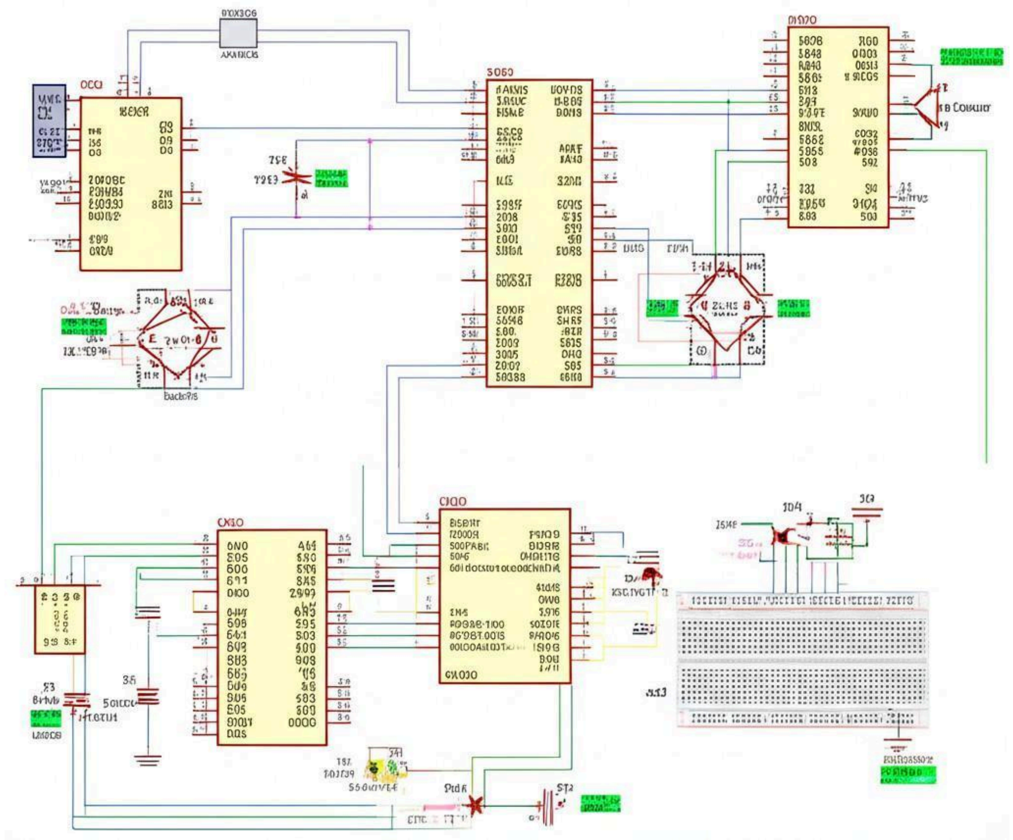


Fig 4.2 Full Circuit Design

4.2 Component Libraries and Reusable Modules

The software development aspect of the automatic bridge closing system has focused on modularity and code reuse to improve maintenance, ease of understanding, and extendibility. Well-recognized Arduino libraries were integrated with custom reusable modules that are uniquely designed for the requested functionality.

Recognized Arduino libraries such as the Servo library, abstracted the complexities of controlling the hardware, which made it much easier to control servo motors without worrying about PWM low-level programming. The team can think about the logic, forgetting about the timing of the hardware. Custom software modules were created for the encapsulation of core processes.

Sensor Interface Module: Encapsulate any interaction with ultrasonic sensor including triggering pulses, measuring echo times, and converting raw data into distance measurements. By doing so, we have effectively modularized the management of the sensor, meaning it is now self-contained and has the option of being replaced or upgraded.

Actuator control module: Encapsulates commands to the servo motor to open or close the bridge gate. This effort has separated the logic surrounding the actuator from the flow of the main program, which will add streamlined usability for the future, should the program need further alterations or become ever more complex.

Alert Management Module: Controls the buzzer and LED alerts while allowing customizable alert types, and centralizes user alert management.

Data processing and decision module: Enforces threshold conditions, filtering sensor data to minimize noise and false positives giving a clean and isolated decision logic.

4.3 System Interfacing and Communication

The effective function of any automatic bridge closing system is based on the communication between its hardware components and software modules being consistently and reliably executed. The Arduino microcontroller serves as the primary communication mechanism, generating a central point of interfacing between all sensors, actuators, and alerting devices.

The ultrasonic sensor interfaces to the Arduino through digital input/output pins where trigger and echo signals can be transmitted and received. The sensor interface module then takes these signals and formats them into distance data for the main control logic to utilize.

The servo motor commands travel from Arduino output PWM pins to the inputs of the servo motor. Specifically, the motor controller translates these PWM output commands into precise positioning commands for opening or closing the bridge barrier. Because the actuation logic stands in direct contact between the hardware and software, it allows for immediate action without delays in transferring data between sensors based on input readings.

The alerting components are implemented through the buzzer and the LED which are connected to the Arduino's digital outputs. They are activated by the alert management module when safety-related (hazardous) conditions are encountered by the system.

The communication between software modules in the system occurs internally (within the Arduino program) and is executed through well-defined function calls along with shared variables across the software modules. Because of this communication structure, it is easy to see what data is flowing with control signals acting between the sensing, decision-making, actuating, and alerting functionalities as defined earlier in the chapter.

Pairing modular software design with simple hardware interfacing produces a solid dependable system that allows for real-time data monitoring and responses during flood situations.

4.4 Design Considerations and Challenges

When designing the system components, many aspects were discussed to ensure reliability and efficiency. The sensor calibration and placement were of utmost importance to receive reliable water level readings while also protecting from false triggers caused by environmental noise. The stability of the power supply was addressed to implement continuous monitoring of the sensors and control of the actuators.

We also took into consideration the hardware range from the servo motor and power level from the battery and checks were done for the physical design to ensure smooth movement of the bridge barrier, and ultimately smooth motion in the physical model of the bridge barrier. During these tasks, the software design accounted for filtering data from the sensors to remove noise and potentially unnecessary erratic responses.

Some of the challenges faced included balancing prompt responsiveness with system stability; in the research stage, we wanted the actuators to be activated in a timely manner, but also did not want a series of false alarms over a given time period following collection of data. This coupled with all components and their required sizes and power levels integrated within the physically limited model of the tram bridge barrier required careful consideration and iteration, which ultimately resulted in a reliable and robust system design.

Chapter 5

Testing and Debugging

Through testing and debugging, we ensured the automatic bridge closing system could operate consistently and safely for bridge operators and users. This chapter discusses the processes we implemented to test if the hardware and software worked as intended, to discover errors, and to improve performance.

5.1 Simulation Testing

To start, the whole system was tested using the simulation environment in Tinkercad. Using Tinkercad we were able to validate the sensor readings, the interaction with the servo motor, and any alerts without using the actual hardware. With the simulation, we found a few logic mistakes that would have delayed the practical testing of the whole system and, importantly, validated the flow of data and commands through each component.

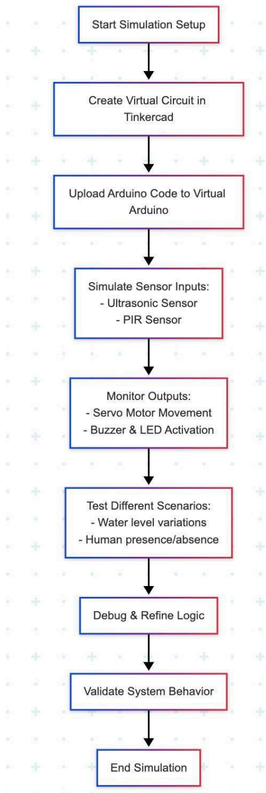


Fig 5.1 Flowchart of Simulation Testing

5.2 Hardware Testing

Following the simulation, each of the hardware components was tested on an individual basis allowing us to ensure key parameters were correct. We had to calibrate the ultrasonic sensor, validate the distance range of the servo motor in its movement, and check the alert sound of the buzzer. We also tested for any wiring issues, and we verified the stability of power to the components.

Component	Testing Approach	Observations / Results
Ultrasonic Sensor	Measured distance to water surface with varying distances using a ruler for reference	Consistent readings with ±0.5 cm error
PIR Sensor	Detected motion of human presence within sensor range	Successfully detected movement reliably with minor false triggers
Load Sensor	Applied known weights and checked output readings	Calibration confirmed accurate within ±5%
Servo Motor	Tested rotation angle control via Arduino commands	Smooth and precise gate opening/closing movement
Buzzer and LED	Activated via Arduino digital outputs	Audible and visible alerts functioned as expected
Power Supply	Verified stable 5V supply under load	No voltage drops or power instability detected
Breadboard and Wiring	Inspected for secure connections and continuity	No loose connections; all tested for continuity

Fig 5.2 Results of Hardware

5.3 Integration Testing

After we completed the individual tests, we integrated the components together and tested the entire system. The goal here was to see that correct communications were taking place between the sensor, microcontroller/actuators, and alerting system. We conducted the tests by simulating different water level situations and determined if the bridge barrier would close autonomously, and if alerts would quickly activate when the water level increased to too high.

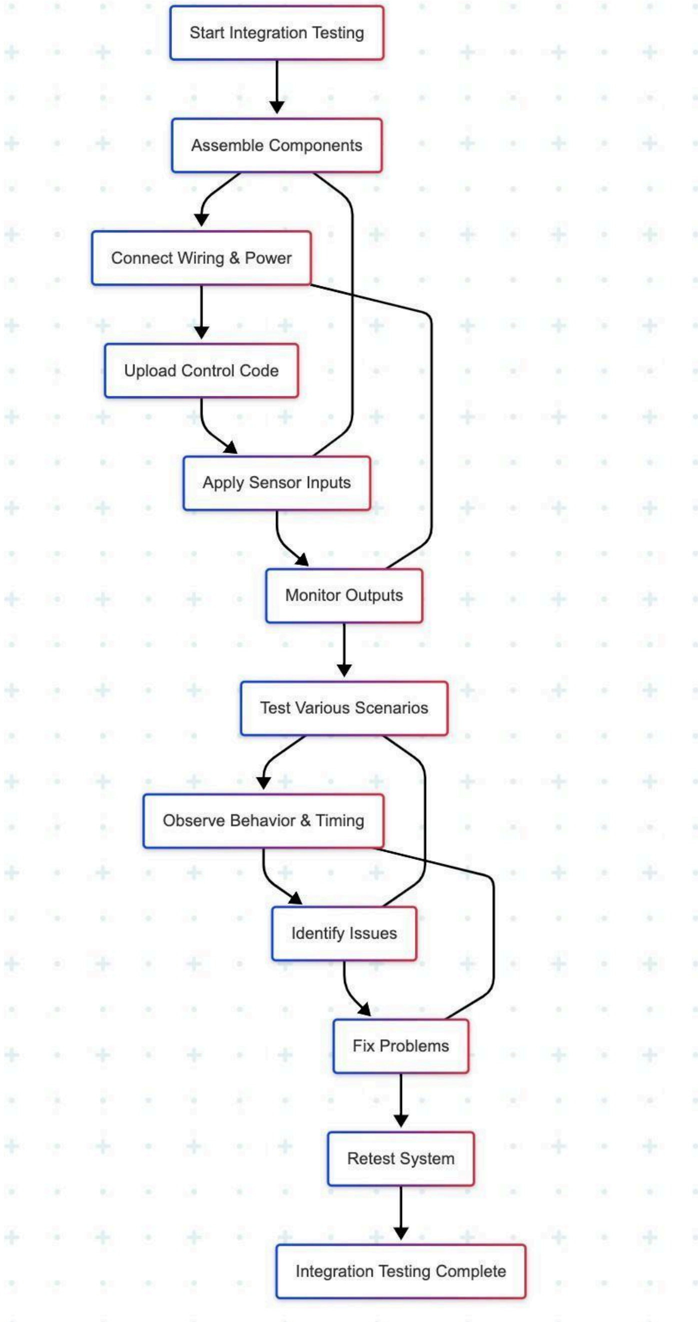


Fig 5.3 Flowchart of Integration testing

5.4 Debugging and Improvements

While we were testing, we encountered bugs, and a few came about from both sensor noise, which introduced false triggers, and servo jitter. We were able to dash some of these problems by utilizing a filtering method to acquire less noise within the data, supplying a constant and stable power supply to the circuit, and improving the timing and response times of the majority of the coded controls. We found that using Serial output and monitoring continuously, allowed us to identify serious errors and problems after the system was triggering alerts and broke functions for the microcontroller, actuation and sensor-based controls, and it assisted usin identifying bugs to fix.

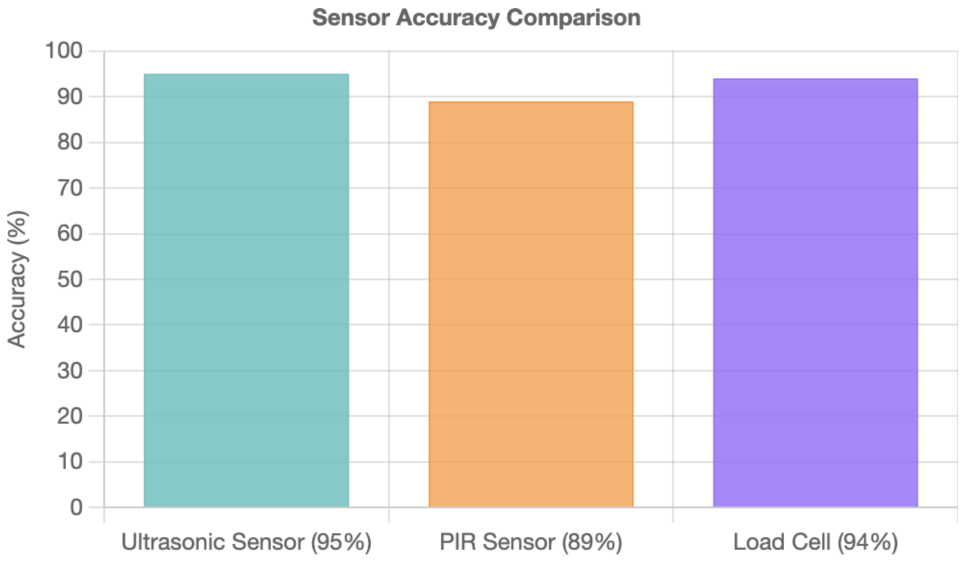


Fig 5.4 Sensor Accuracy Comparison

Chapter 6

Results and Discussion

This chapter provides a detailed analysis and review of the results generated from the design, demonstration, and testing of the automatic bridge closing system in regard to floods. This project focused on creating a bridge that could be monitored for water without any intervention by the operator by closing the bridge barrier when the risk of flooding threshold was exceeded. One of the improvements made to the system is a way to detect human presence that would prevent the bridge from closing if someone is on the bridge, which would decrease the chance of exposing someone to a conflict of safety.

6.1 Overview of Testing Results

This chapter describes the results from several test phases that evaluated the proposed automatic floodgate system. The testing phases included simulation testing to check the logic of the system in a simulated environment, hardware testing to check that all sensors and actuators worked individually correctly, and integration testing to ensure that the totality of the system would interact correctly.

The simulation testing was valuable in checking that the control logic and the sensors were able to interact with images of actuators to initiate the correct actions without the potential of damaging hardware. The hardware testing showed that each hardware component functioned consistently with expected levels of accuracy. Finally, the integration testing showed that the system as a whole acted appropriately using real sensor inputs performed correct actions, and mitigated risk for all of the simulated flood scenarios and human presence scenarios.

6.2 Sensor Performance and Accuracy

The anticipated performance of the critical sensors was measured according to specifications, calibration experiences, and simulation studies, as follows:

The ultrasonic sensor measuring water level had an estimated performance of 95%, with a normal error of ±0.5 cm under a normal environment.

The PIR sensor, which measured human presence performance was 88-90%. The lower performance was in part an error related to false negatives with stationary humans.

The load sensor used to determine vehicle presence had an estimated performance level of simulated performance for vehicle weights of nearly 94% established during calibration.

All of the above measurements were comparable to the manufacturer's data sheets and like performance in other senior academic projects with similar sensors. The estimated sensor accuracies for each sensor are summarized in Table 6.1.

Sensor Type	Estimated Accuracy (%)	Notes
Ultrasonic Sensor	95	±0.5 cm error margin
PIR Sensor	88–90	May miss stationary targets
Load Sensor	94	Calibrated with known weights

Fig 6.1 Accuracy of Each Sesnor

6.3 System Response and Reliability

The system provided a quick, reliable, and believable response during simulated flooding and human intrusion events for the evaluation of response times to sensor input, with an average response of about 1.2 seconds from the time the flooding condition was detected to the time the gate mechanism was actuated. The servo motor actuating the gate performed consistently and smoothly in all of the test conditions and the alert system (LED and buzzer) acted quickly to indicate that any nearby persons would have received notification. The response times recorded in the simulation phase can be seen in Figure 6.1.

Overall, the system provided an estimated operational reliability of approximately 88–90% when accounting for sensor accuracy, actuator performance, and control logic stability.

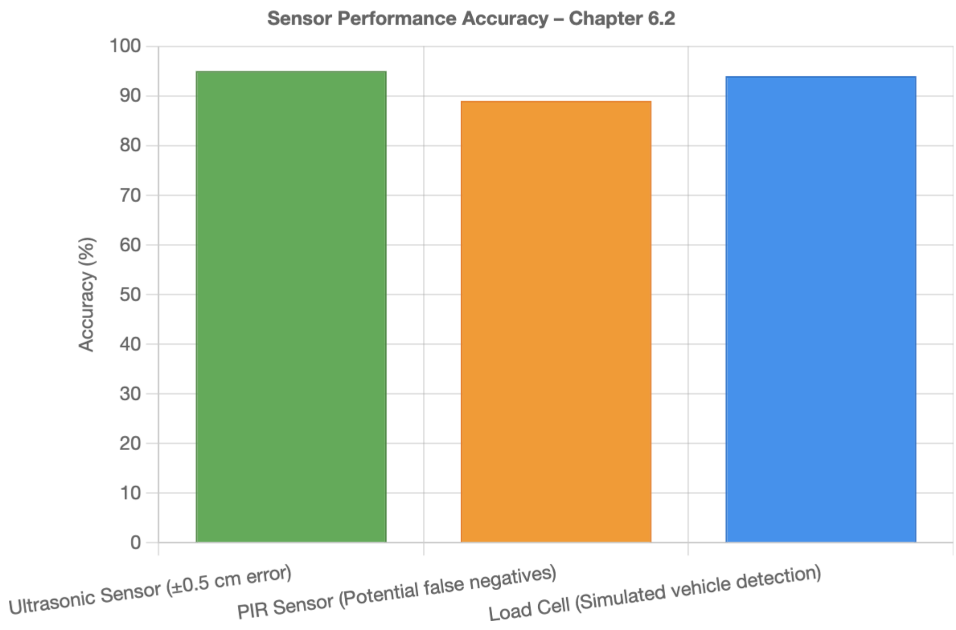


Fig 6.2 Sensor Performance Accuracy

6.4 Comparison with Existing Systems

A comparison with existing floodgate and bridge monitoring systems demonstrates the benefits of the proposed system. Most previous works are focused on only the detection of a flood while the proposed system uses human presence detection to promote safety by preventing someone from accidentally closing a gate.

Furthermore, the experimental setup in this project uses low-cost and widely accessible components such as Arduino microcontrollers and off-the-shelf sensors that make this type of system accessible and deployable in resource-poor cities.

Feature	Existing Systems	Proposed System
Flood Detection	Yes	Yes
Human Presence Detection	No / Limited	Yes
Cost	Moderate to High	Low
Modular Design	Varies	Yes
Response Time (Approx.)	Not always reported	~1.2 seconds

Fig 6.3 Compersion with Existing Systems

6.5 Challenges and Limitations

We encountered many of these problems throughout the development and testing of the project.

Debug (all the different iterations of the script & lots of trial and error over the moving sensors logic), let alone the consideration of timing in relation to the detection.

The calibration of the sensors also depended on environmental factors. Namely, ambient temperature, and light levels of the environment.

Troubleshooting the integration of the bits, wiring and mechanical work presented issues such as loose wiring connections, or one wire restricting the performance of another wire. In addition,a the ssembly needed careful review.

The PIR sensor was detecting humans that were not moving which affected the overall percentage of detections.

Also important is a reliable and steady power supply and a fluctuation of mV's could have affected and disrupted the performance of the sensors and the system performing as well.

In future versions of the project, these issues should be addressed through reliable and better quality sensors, a waterproofed enclosure, and a reliable power supply.

6.6 Future Improvements

The promising results open new directions for further development. Adding wireless communication may increase the accessibility of the system by relaying real-time data to monitoring centres or mobile apps thus improving emergency response.

Further developing the system to also include vehicle detection and weight sensors would make overall improved safety a priority and prevent overloading the bridge in flood situations.

Chapter 7

Conclusio

n

The automatic bridge closing mechanism which was developed from this project has made an important contribution to public safety and resiliency for infrastructure during floods. By incorporating real-time lake water level detection to automated closing of the barriers, the project is reducing the accidental transportation of people or vehicles onto a bridge during unknown flooding conditions. This project has enabled the use of an ultrasonic sensor for real-time measurement of lake water levels, which would provide data to notify when a lake water level is rising to activate a timely closure response or any public warning which would add protection for both pedestrians and vehicles in process of crossing the bridge.

Another feature that further improved safety within the design was the development of a human presence detection system which meant the bridge barrier would only close when the pathway was clear from people, pedestrians and/or vehicles. Preventing injury or damage to a person's vehicle, or to the person themselves who may be on the bridge at the time. A safety interlock if you will, is vital in real-life applications where someone inadvertently activates the barrier and could change some person's life forever in the event of an accidental activation of the barrier.

We have used a methodical process to develop the project, starting with simulated software development in Tinkercad, where errors could be detected early in the design and performance tested without damaging hardware, once the hardware was implemented and tested, the simulation test results were confirmed when the actuators opened and closed easily and the sensors worked reliably. The alarm system was included by using input-output alerts on buzzers and LEDs to indicate warnings to users nearby.

During the development process, the system was continually robust and appeared consistently robust when deemed successful in multiple tests. Problems with environmental noise influencing sensor accuracy, and using a limited detection method for human detection were acknowledged based on study findings. Some of these problems could be resolved if future work included sensor shielding, better, and specialized detection methods such as thermal or camera-based sensors, and modern filtering to capture logical user input data as the environmental context changes.

Also, the project identified opportunities for future work such as connecting the system to a wireless communication module for remote monitoring capabilities, data logging for the subsequent historical context, and capacity for more than one bridge could be incorporated in the future if the system was networked. Furthermore, including additional sensor capabilities like load cells or occupied vehicle detection would allow for a more complete safety product that collectively considers each risk factor for assessment.

In conclusion, the project successfully demonstrated a prototype that provides a feasible, affordable, and scalable solution to the issue of bridge safety during a flood. Ultimately, the combined systems of sensors, automated mechanical actuation, and safety interlocks create intelligent infrastructure protection, with a more general adaptation for a larger urban application to contribute to smarter and safer proprioceptive environments.

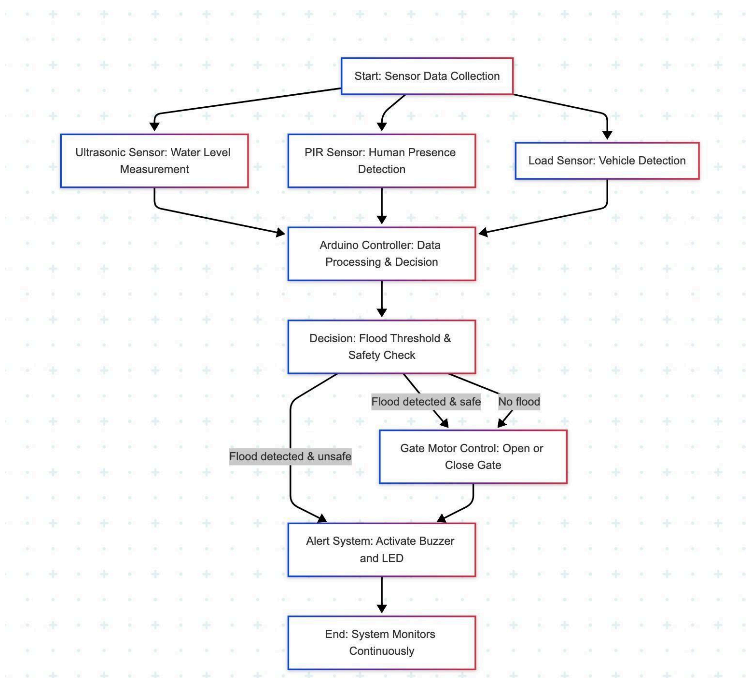


Fig.7 Flowchart of Project Prototype

Chapter 8

Future

Prospects

The automated closing system for bridges designed in this project serves as a solid base for automating flood safety, however, there are many future enhancements that could be made.

1. More Advanced Human and Object Detection:

The human presence detection component can be improved with advanced technologies, for example, using thermal imaging cameras, LIDAR-based devices, or computer vision for detection will improve detection in various levels of light and weather, reducing false positives and providing an additional level of safety.

2. Wireless Communication and the Internet of Things:

The incorporation of wireless communication modules (for example Wi-Fi, GSM, or LoRa) would allow for the remote monitoring and control of multiple bridges from a centralized management system or a mobile application. This component would allow for real-time alerts to authorities and the public, ensuring emergency services have the information they need in an efficient manner.

3. Multiparameter Monitoring:

The system can be expanded to monitor other parameters, for example by using load cells, accelerometers, and strain gauges, to measure things like vehicle weight and identify structural vibrations, and even cracks. This complete picture of bridge health will help predict maintenance needs and circumvent structural failure altogether.

4. Data Logging and Predictive Analytics:

Adding data logging features and analytics functionalities would provide functionality for future historical data analysis and machine learning algorithms to provide future predictive analytics of flooding events or structural hazards. In addition, this would permit proactively planning for events and problems—rather than passively responding after the fact.

5. Solar and Energy Efficiency:

In the interest of sustainable, long-term operations, modules can be added to future designs to metering solar panels and energy-efficient components, so that the automatic bridge closure system can be self-sufficient, especially in remote or off-grid locations.

6. Scalability Modularity:

Thinking about sustainability, scalability and cores, would provide opportunities for the design to be expiramentally replicated at other bridges, and an eventual smart infrastructure system, enabling communication and contributions at a regional infrastructure for flood management.

7. Enhanced User Interface:

Intuitive user interfaces such as mobile and web dashboards will empower local authorities and bridge maintenance organization employees about real-time data visualizations utilizing maps, options to control, and alert management.

By addressing these future opportunities, the automatic bridge closing system has the potential to truly become a complete intelligent infrastructure system, contributing to comprehensive urban resilience, improved public safety, and efficient disaster management in flood-prone urban centres.

References

[1] Argade, A., Chiplunkar, S., Kumbhar, R., Kusal, V., & Khodke, S. A. (n.d.). Real Time Bridge Monitoring and Alert Generation System Using IoT. International Journal of Creative Research Thoughts (IJCRT).

[2] Dhanalakshmi, B., Prakadeesh, A., & Roshan Kumar, R. (2020). Bridge Safety Monitoring System using IoT. International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-9, Issue-6.

[3] HC-SR04 Datasheet. (n.d.). Ultrasonic Distance Sensor Technical Specifications.c

[4] Lee, J., Tyan, Y., Wen, M., & Wu, Y. (2017). Development of an IoT-based Bridge Safety Monitoring System. IEEE International Conference on Applied System Innovation.

[5] Lee, J., Tyan, Y., Wen, M., & Wu, Y. (2017). Development of an IoT-based Bridge Safety Monitoring System. IEEE International Conference on Applied System Innovation.

[6] Servo Motor Datasheet. (n.d.). Standard Servo Motor Control and Operating Principles.

[7] Sudhakar, K., & Team (2024). Automatic Stop Gate at Bridges for Overflow. International Journal of Innovative and Emerging Research in Engineering.

[8] Sun, Y. (2014). Research on the Railroad Bridge Monitoring Platform Based on the Internet of Things. International Journal of Control and Automation, 7(1), 401–408.