EC6020 - MINI PROJECT PROPOSAL

INTELLIGENT DAM SAFETY SYSTEM FOR REAL-TIME WATER LEVEL FORECASTING AND AUTOMATED GATE CONTROL

GROUP NO: 3

RENUJAN J.	2022/E/065
ANOJINTH K.	2022/E/076
KILSHAN P.	2022/E/109
ANTONYRAJ T.	2022/E/160
UTHAYARAJ V.	2022/E/166

INTRODUCTION

The global challenges of dam safety and flood risk management will continue to be serious issues due to the drastic changes in performance of dams when rapid elevation increases occur in reservoirs due to extreme weather events, which require timely and accurate operational decisions being made in response to these rapid changes.

This issue has been illustrated by the recent flood disaster in Sri Lanka , which occurred as a result of delayed actions taken to prevent rapid water elevations, which had adverse social and infrastructure consequences. Therefore, incidents such as this demonstrate the need for a proactive, automated, and reliable dam management solution.

The Smart Dam Safety System will provide a solution by integrating real-time forecasting of water levels, an early warning system, and an automated gate control system into one compact embedded device. This Smart Dam Safety System is designed to be utilized for the continuous monitoring of reservoir conditions and provide a forecast of future water level trends, allowing for the automatic adjustment of gate openings to the required height based on the predicted risk of overflow. Therefore, the Smart Dam Safety System does not require human operation, as it provides a fully automated solution. It offers a cost-effective and permanent solution to the challenges presented by extreme weather events and enhances dam safety while reducing reliance on human intervention, providing a scalable and commercially viable solution that can be utilized across all regions and economic conditions.

PROBLEM STATEMENT

Dam safety and flood risk management systems currently in use exhibit several limitations when evaluated in terms of cost, scalability, intelligence, and long-term deployability:

- Lack of real-time monitoring, forecasting, and early warning in one solution.
- There is no affordable, permanent system that can automatically control dam gates to prevent overflow disasters.
- Manual monitoring and control cause delays during sudden water level changes.
- Existing advanced systems are expensive and difficult to deploy widely.

PROPOSED SOLUTION

The proposed system addresses the identified problems through the following features:

- Early warning generation based on forecasted and real-time risk levels
- Continuous real-time monitoring of reservoir water levels.
- Predictive water level forecasting to identify potential overflow conditions in advance.
- Automatic dam gate control to adjust gate opening height without manual intervention.
- A cost-effective, embedded system design suitable for permanent deployment at dams.
- Live data transmission for remote monitoring by authorities and field workers.
- A web-based interface for real-time visualization of water levels, alerts, and gate status.

DESIGN OVERVIEW

- **System Architecture**

  The system consists of a microcontroller-based embedded unit that interfaces with sensors and actuators, along with a software-based user interface.

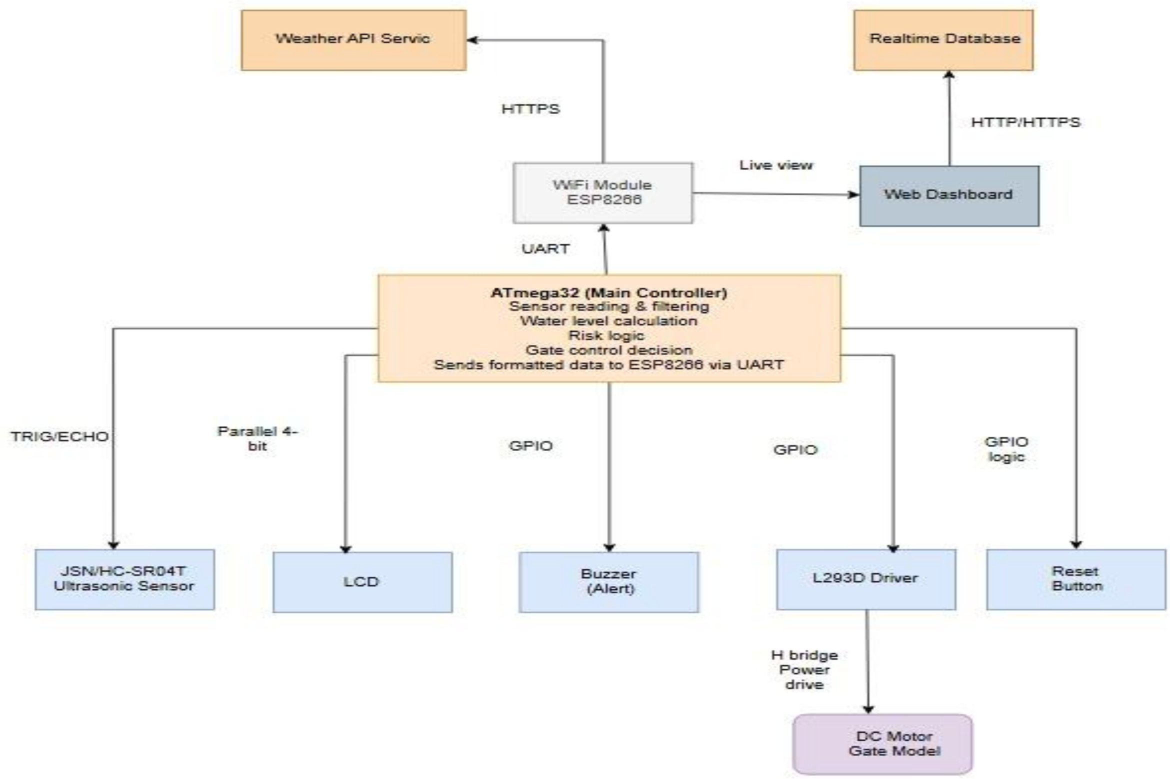


FIGURE 1: SYSTEM ARCHITECTUR

Hardware Interaction

- The ultrasonic sensor physically measures the distance to the water surface.
- The ATmega32 microcontroller connects all hardware components and receives raw signals from sensors.
- The LCD shows the current water level and risk status near the dam.
- The buzzer gives an audible alert when the water level becomes high.
- The L293D motor driver controls the power supplied to the motor.
- The DC motor represents the dam gate opening and closing mechanism.
- The ESP8266 WiFi module provides wireless connectivity to the internet.
- The reset button allows manual restarting of the system during faults.

Software Interaction

- The embedded software in ATmega32 reads data from the ultrasonic sensor.
- It filters the readings and calculates the water level in cm and percentage.
- The software compares the level with thresholds to decide Normal, Alert, or Danger.
- Based on the decision, the software:
  - Updates the LCD
  - Turns the buzzer ON or OFF
  - Sends control signals to the motor driver
- The ATmega32 formats the data and sends it to the ESP8266 via UART.
- The ESP8266 software uploads live data to a realtime database using HTTP/HTTPS.
- The web dashboard software displays live water level, risk status, and gate state.
- The ESP8266 also retrieves real-time weather data from a Weather API to improve risk prediction.

Communication Protocols

1. Ultrasonic Sensor ↔ ATmega32
   Protocol Used: Trigger–Echo Pulse Timing (Digital I/O)
   - Used for continuous water-level measurement using trigger and echo pulses

2. LCD ↔ ATmega32
   Protocol Used: Parallel Communication (4-bit mode)
   - Used to display water level and risk status locally

3. Buzzer ↔ ATmega32
   Protocol Used: Digital Output (GPIO control)
   - Used to generate audible alerts during warning conditions

4. Reset Button ↔ ATmega32
   Protocol Used: Digital Input (Active-Low Reset Line)
   - Used to reset the system through hardware control

5. L293D Motor Driver ↔ ATmega32
   Protocol Used: GPIO Logic Control
   - Used to control motor direction and enable signals

6. L293D Motor Driver ↔ DC Motor
   Protocol Used: H-Bridge Power Drive Interface
   - Used for bidirectional motor operation

7. ATmega32 ↔ ESP8266 WiFi Module
   Protocol Used: UART (Serial Communication)
   - Used for serial data transfer between controller and WiFi module

8. ESP8266 ↔ Realtime Database , Web Dashboard
   Protocol Used: HTTP/HTTPS (REST) over WiFi with JSON
   - Used for uploading live system data to the web platform

9. ESP8266 ↔ Weather API Service
   Protocol Used: HTTPS (REST API) with JSON
   - Used to retrieve real-time weather information

- ### Circuit Design

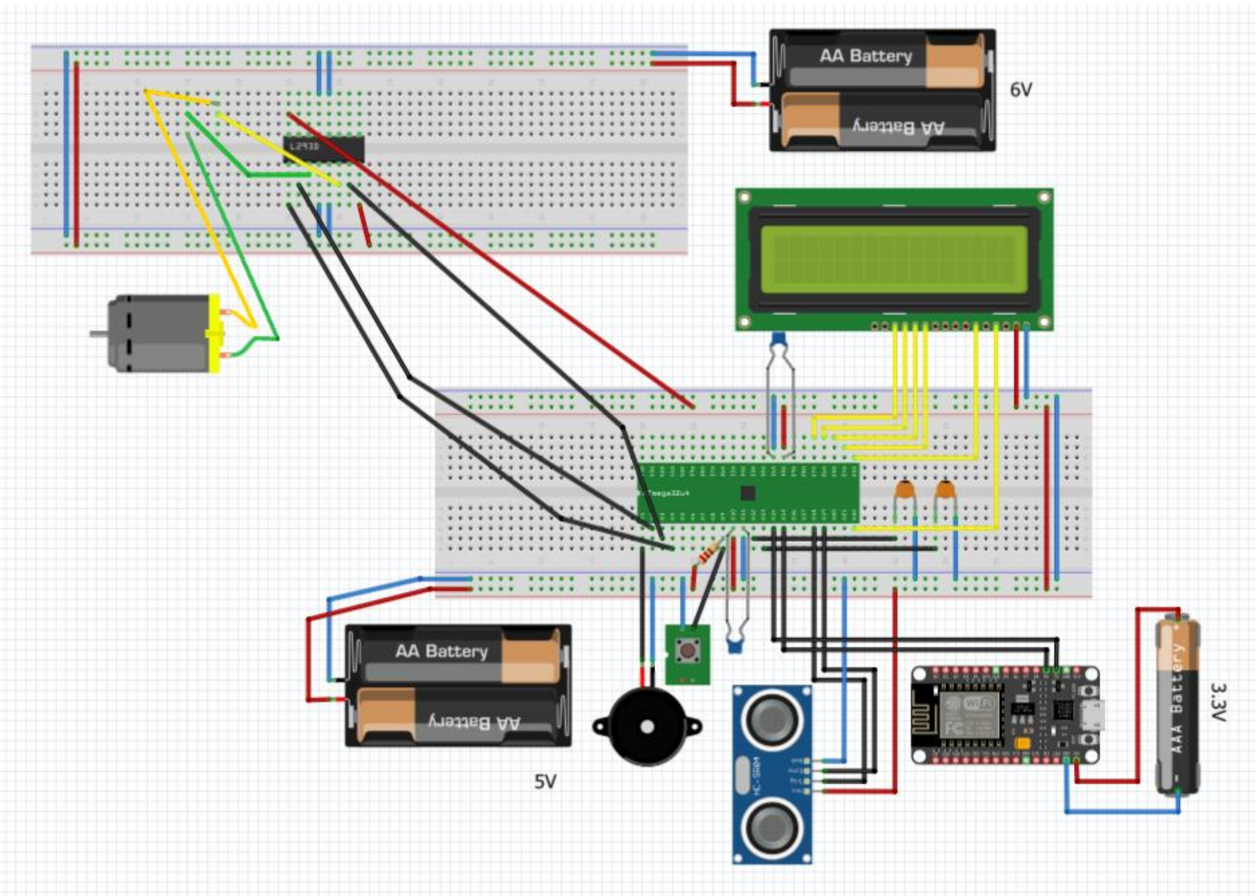


FIGURE 2: CIRCUIT DESIGN

TECHNOLOGIES TO BE USED

- ### Hardware Components
  - ATmega32 DIP40
  - Capacitors
  - Ultrasonic sensor (HC-SR04)
  - Buzzer
  - Wifi module (ESP8266)
  - Battery
  - Push button Switch
  - Resister
  - Bread board
  - 16x2 LCD module
  - Jumper wires
  - DC motor
  - L293D driver

- **Software Technologies**
  - Embedded C++ – Used to program the ATmega32 microcontroller for sensor reading, risk calculation, and device control.
  - ESP8266 Firmware – Used to handle WiFi communication and data transmission to the web.
  - HTTP/HTTPS (REST API) – Used for sending data to the realtime database and accessing weather data.
  - JSON Format – Used to structure sensor and system data for web communication.
  - Firebase Realtime Database – Used to store and update live system data.
  - HTML, CSS, JavaScript – Used to develop the web-based dashboard for live monitoring.
  - Weather API – Used to obtain real-time and forecast weather data.
  - Github – for version control.

UNIQUENESS OF THE PROPOSED SYSTEM

The proposed system stands out due to the following innovative aspects:

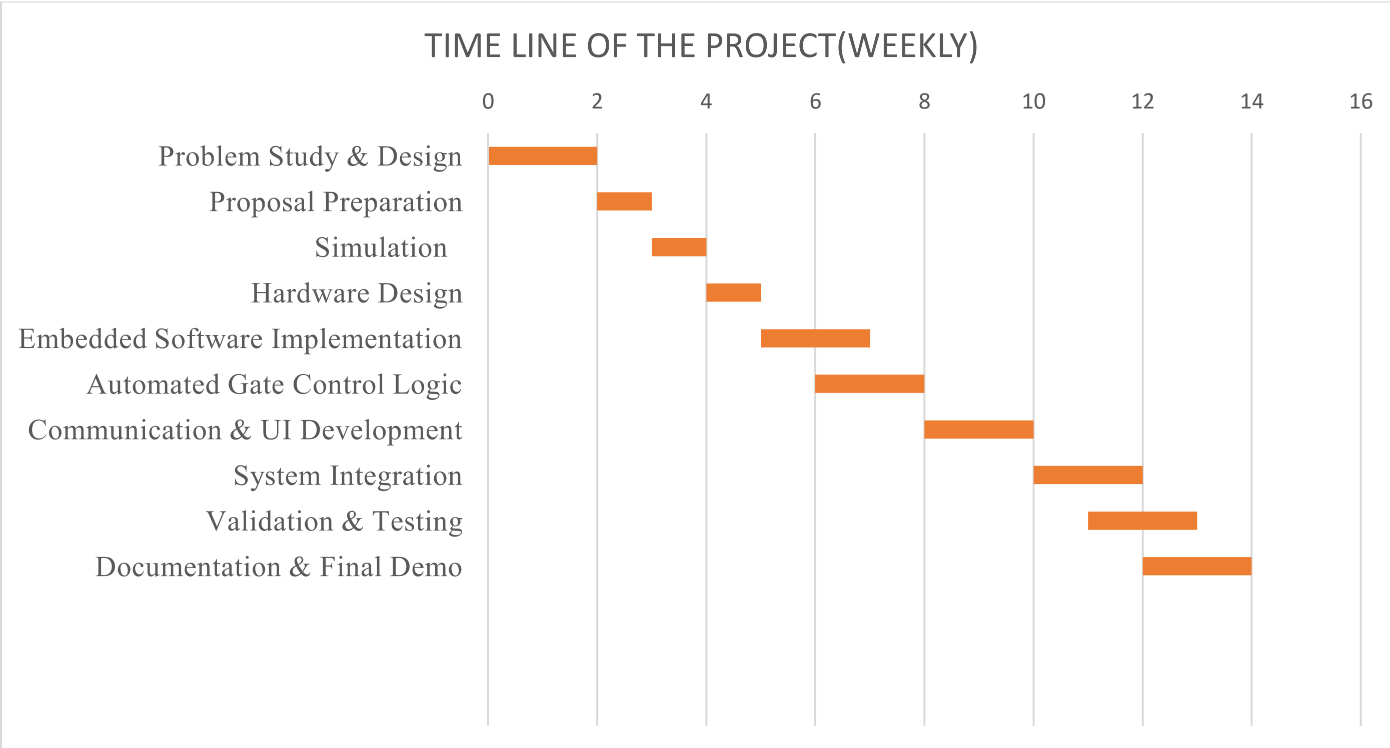
- Integrates real-time water level monitoring, forecasting, early warning, and automated gate control in one system.
- Enables proactive dam safety by predicting risks and acting before overflow occurs.
- Reduces manual intervention through automatic gate operation.
- Combines embedded hardware, control logic, and a user interface into a single solution.
- Designed to be cost-effective and scalable for practical deployment.

BUDGET ESTIMATION

Table 1: Budget Estimation

Component	Est. Price (LKR)
Capacitors (100nF - 2, 22pF - 2)	30.00
ATmega32 DIP40	900.00
Buzzer	50.00
Ultrasonic sensor (HC-SR04)	240.00
Resisters	320.00
16x2 LCD module	300.00
Stripboard	240.00
L293D driver	100.00
DC motor(1)	20.00
Push button Switch	130.00
Wife module (ESP8266)	270.00
Total Estimated Cost	2600.00

PROJECT TIMELINE



GITHUB REPOSITORY

The project is maintained in a public GitHub repository containing:

- All source code
- Circuit diagrams
- Documentation and README file

Repository Link:
https://github.com/JeyakumarRenujan/Dam_Safety_System.git

CONTRIBUTION

2022/E/065: Wireless Communication & Web Application

2022/E/076: Hardware Design & Circuit Implementation

2022/E/109: System Architecture & Planning

2022/E/160: Testing & Documentation

2022/E/166: Embedded Software Development