JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA-62

B. TECH SEMESTER V

ODD SEM 2025

COMPUTER ORGANISATION AND ARCHITECHTURE LAB PROJECT REPORT

[15B17CI373]

TITLE:

Distance-Based Multi-Modal Safety Alert System

ENROLLMENT NO.	23803006	23803014	23803020
NAME	ASMI VAISH	LOVISH KUMAR	KALPNA MISHRA

SUBMITTED TO:

Dr. Naveen Chauhan

INDEX

S. No.	Chapter/Section
1.	Abstract of the Project
2.	Introduction
3.	Literature Review
4.	Scope of the Project
5.	Tools and Technologies Used
	5.1. Hardware Components
	5.2. Software & Programming Tools
6.	Design / Block Diagram of the Project
	6.1. Block Diagram (Concept Flow)
	6.2. Explanation of Design Flow
7.	Implementation Details
8.	Methodology / Working Principle
9.	Hardware & Software Requirements
	9.1. Hardware Requirements
	9.2. Software Requirements
	9.3. Arduino IDE Code
10.	Working Flow & Circuit Explanation
11.	Testing / Output Results
	11.1. Summary of Final Test Results
12.	Applications
13.	Future Scope
14.	Conclusion
15.	References

Abstract of the Project

This project focuses on developing a smart obstacle detection and alert system using an ultrasonic sensor and an Arduino UNO. The primary aim of the model is to continuously measure the distance of any approaching object and accordingly generate safety alerts using LEDs and a buzzer.

The ultrasonic sensor emits high-frequency waves which strike an object and return back as an echo, and based on the time taken for the echo to return, the Arduino calculates the distance. According to the calculated value, the system indicates three different alert levels – green LED when the distance is safe, yellow LED when the object comes moderately close, and red LED along with a buzzer when the object enters the danger zone.

This makes the system reliable for avoiding collisions and improving safety. The project is simple to implement, low cost and highly useful for applications like parking assistance, robotics, industrial safety and visually impaired support.
With further improvement such as LCD display, IoT integration or servo-based scanning, the project can be expanded into a more advanced intelligent safety device.

Introduction

Rapid development in embedded systems and automation has increased the demand for intelligent safety solutions. In daily life as well as in industrial environments, collision with obstacles often causes damage, delays and safety risks. Machines and humans cannot always respond instantly to unexpected objects in front of them, and therefore a system that detects obstacles automatically and alerts the user becomes very useful. Our project is an attempt in this direction, where an Arduino UNO along with an ultrasonic sensor is used to measure distance in real-time and provide alert signals whenever something comes close.

The idea behind the project is simple yet effective. The ultrasonic sensor works like the eyes of the system by continuously sending sound pulses and waiting for their reflection. As soon as the waves hit an object, they return back as an echo and this time difference helps in finding out the exact distance. Based on this reading, the Arduino controls LEDs and a buzzer to indicate how close the object is. This makes the project suitable for safety applications like parking, robotics and blind assistance.

The reason for selecting this concept is that it is low-cost, useful and easy to build even for beginners. No complex hardware is needed and the entire model can be assembled on a breadboard. This system not only reduces chances of accidents but also improves awareness, especially in areas where visibility is low or movement space is narrow.

The project can further be expanded into advanced versions by adding display modules, wireless connectivity or motorised scanning. Overall, this project forms a foundation for developing smart, automatic and efficient obstacle detection systems.

Literature Review

Obstacle detection and proximity alert technologies have been researched and used for many years, especially in the fields of automation, robotics and intelligent vehicle systems. Earlier safety systems mostly depended on infrared sensors or physical contact switches, which were not very accurate in measuring distance and sometimes failed when the object had an uneven surface. With time, ultrasonic-based sensing gained popularity due to its non-contact measurement ability and reliability in different light conditions. Research studies show that ultrasonic waves are effective in detecting solid objects and can measure distance by calculating the time taken by pulse echo. This technique is already used in car parking sensors, smart robotics and industrial automation lines.

Several academic projects similar to ours have been developed in past years, focusing on short-range object detection. Many of them used microcontrollers like Arduino, Raspberry Pi or PIC to process sensor data. However, most projects focused only on displaying distance and did not provide multi-stage alert indication as implemented in our system. In an engineering journal report, ultrasonic sensors were used for automatic reverse parking in cars, where the buzzer intensity increased with closeness of the obstacle. Another paper described a blind-navigation stick designed with vibrational feedback instead of LEDs. Compared to those, our project uses a simpler yet more user-understandable three-level LED indication system.

By reviewing multiple previous works, we understood gaps such as lack of adaptive colour alerts, limited range classification and absence of easy expansion. Our system attempts to refine these aspects by making the design modular, more interactive and suitable for multiple environments.

Scope of the Project

The scope of this project is quite broad because the system is designed in a modular and expandable manner. At its core, the obstacle detection unit can sense nearby objects and notify users through LED signals and a buzzer. This basic functionality itself has a wide range of real-time applications, especially in areas where human attention, visibility, and reaction time are limited. Since the project does not rely on complex components and operates on a very low voltage level, it is safe to deploy in indoor as well as outdoor environments with minimal modification.

The system can be used in automobiles for reverse parking assistance, where lack of judgment in small spaces often leads to scratches or minor accidents. Similarly, it can be integrated with mobile robots and automated vehicles for collision avoidance. In industries and warehouses, the same circuit can prevent forklifts or moving robots from hitting objects or workers. The design is small enough to be attached to helmets, assistive walking devices, or even blind-support tools, where it could act as a real-time movement guide.

The scope is not only limited to hardware application — the project can also be extended at software and communication levels. For example, a display can be added to show exact distance readings, or IoT modules can be introduced for remote monitoring and alert tracking. A servo motor can be paired with the sensor to create a 180-degree scanning radar, enabling wider detection. The same device may also be developed into a fully smart safety product by integrating machine learning-based prediction of movement patterns or Bluetooth-based smartphone alerts.

In conclusion, the scope of this project extends from simple laboratory demonstration to real-world usage in vehicles, robotics, automation industries, and personal safety equipment. Since the system is low-cost, easy to assemble, consumes less power, and can be upgraded anytime, it holds a strong potential to grow into more advanced commercial-grade safety technology.

Tools and Technologies Used

This project is developed using basic but powerful electronics and programming tools that work together to create a functional sensing and alert mechanism. Each component in the system plays an important role, and understanding their purpose helps in both better implementation and future upgrades. The technologies used are selected keeping in mind simplicity, low power consumption, availability in the market, and ease of understanding for beginners in embedded systems.

Hardware Components

1. **Arduino UNO**
    - Acts as the main controller of the project.
    - Reads signals from the ultrasonic sensor, processes distance values, and triggers LED/buzzer responses.
    - Easy to program using USB interface and supports multiple sensors simultaneously.
2. **Ultrasonic Sensor (HC-SR04)**
    - Used for real-time distance measurement.
    - Emits ultrasonic waves that reflect back after hitting an object.
    - Provides accurate readings within 2 cm to 400 cm range.
3. **LED Indicators (Red, Yellow, Green)**
    - Display visual output based on distance level.
    - Green → Safe distance
Yellow → Medium distance
Red → Danger proximity
4. **Buzzer**
    - Generates audible alert when danger level is reached.
    - Helps in warnings where visual attention may not be enough.
5. **Mini Breadboard**
    - Used for circuit assembly without soldering.
    - Makes testing and component rearrangement easier.
6. **220Ω Resistors**
    - Placed in series with LEDs to prevent excessive current flow.
    - Ensures safety of LEDs and prolongs their lifespan.
7. **Jumper Wires**
    - Connect different components over the breadboard.
    - Provide flexible wiring for power and data transmission.

Software & Programming Tools

1. **Arduino IDE**
    - Used to write, compile, and upload the code into the Arduino board.
    - Supports serial monitoring for debugging and testing sensor values.
2. **C/C++ Programming Language**
    - Core logic for distance calculation and output control is written in ArduinoC.
    - Simple syntax and easy integration with embedded systems.
3. **Serial Monitor (Optional for Debugging)**
    - Helps observe real-time distance values during testing.
    - Useful for verifying sensor response and adjusting thresholds.

Design / Block Diagram of the Project

The design of this project is planned in a simple but effective way so that distance measurement and alert generation work smoothly without delay. The entire system is divided into two main parts — **Input Unit** (Ultrasonic Sensor) and **Output Unit** (LED Indicators + Buzzer). The Arduino UNO acts as the central processor that receives input, makes decisions, and controls the output signals. The block diagram below gives a clear overview of how data flows inside the system.

Block Diagram (Concept Flow)

```
        ULTRASONIC SENSOR (HC-SR04)
              |
         Measures Real-Time Distance
              |
              ▼
           ARDUINO UNO
      (Processes Distance & Compares Levels)
   ┌──────────────────┬──────────────────┐
   ▼                  ▼                  ▼
GREEN LED        YELLOW LED        RED LED + BUZZER
 Safe Zone       Warning Zone        Danger Zone
```

Explanation of Design Flow

1. **Ultrasonic Sensor as Input Source**
   The HC-SR04 continuously emits ultrasonic waves and waits for their reflection. The time taken for the echo to return helps determine how far the object is. Faster echo means the object is closer.

2. **Arduino UNO as Decision Controller**
   The measured time is converted into distance through programmed logic. The Arduino compares this value with pre-defined ranges and decides which output should activate.

3. **LEDs for Visual Indication**
   - Green LED glows when the object is far enough and there is no risk.
   - Yellow LED turns ON when distance begins to fall into a moderate zone.
   - Red LED lights up when object becomes dangerously close.

4. **Buzzer for Audio Alert**
   The buzzer creates sound only in danger conditions where immediate attention is required. This helps alert even when the user is not looking at the LEDs.

Why This Design is Effective?

- Very low complexity → easy to understand and troubleshoot.
- Real-time response → suitable for moving vehicles or robots.
- Can be extended by adding display, multiple sensors or servo scanning.

Implementation Details

The implementation of this project was carried out step-by-step to ensure that every component worked properly before integrating everything together. The setup was first tested individually for sensor response, LED blinking, and buzzer activation. After confirming basic functionality, the complete circuit was assembled on a mini breadboard and connected to Arduino UNO for final execution.

The implementation involved both hardware construction and software programming, making it a complete embedded project.
The Ultrasonic Sensor was connected with the Arduino using four pins — VCC, Trig, Echo, and GND. The Trig pin was used to send a sound pulse, while the Echo pin received the reflected wave and returned timing information back to the Arduino. LEDs were connected using 220Ω resistors to prevent overheating, and the buzzer was attached to a digital pin so that it could be controlled through code. Jumper wires made all plug-and-play connections easier to modify and rearrange during testing.

Once the wiring was complete, Arduino IDE was used to upload the code. Inside the program, a trigger pulse of 10 microseconds was sent repeatedly and the echo duration was measured. This duration was then converted into distance using the speed of sound formula. Based on the measured distance, the program turned ON the appropriate LED, and in critical conditions, activated the buzzer. Multiple trials were conducted with objects like hand, book and plastic bottle to confirm detection reliability.
During implementation, some small improvements were done such as adjusting threshold values for better response and positioning the sensor at an ideal straight angle for maximum accuracy. After assembly and coding, the system started functioning exactly as planned — detecting objects, measuring distance and alerting instantly. This confirms that the implementation was successful and the model was ready for practical demonstration.

Methodology / Working Principle

The working principle of this project is based on the reflection of ultrasonic sound waves. The ultrasonic sensor continuously sends out high-frequency pulses through its transmitter. Whenever these pulses hit an object, they reflect back and return to the receiver section of the sensor. The time taken for the pulse to travel forward and then return back is measured by the Arduino. Because sound has a fixed speed in air, the distance between the sensor and the object can be calculated using a simple mathematical formula. This process happens multiple times per second, which allows our system to detect obstacles in real-time.

Inside the program, the Arduino sends a short trigger pulse of 10 microseconds to the Trig pin of the sensor. After receiving this pulse, the sensor emits ultrasonic waves at around 40 kHz. The Echo pin stays HIGH until the reflected wave is detected. The duration of this HIGH time represents the round-trip travel time of the sound wave. The Arduino reads this duration and converts it into distance using the speed of sound (approximately 343 m/s) as reference. The distance formula used in the project is:

Distance (cm) = (Time × Speed of Sound) / 2

The division by 2 is necessary because the sound wave travels twice — once towards the obstacle and once back to the sensor. Once the distance is calculated, conditional statements inside the code compare this value with predefined ranges. If the distance is more than 25 cm, the object is considered safe and the green LED lights up. When the object moves closer, within 10 cm to 25 cm range, the system enters the warning state and the yellow LED glows. If the distance falls below 10 cm, it is treated as a danger condition and the red LED glows along with the buzzer to alert the user immediately.

The entire process repeats continuously inside the loop of Arduino, which means the system never stops monitoring. As soon as the object changes position, the sensor detects it instantly and updates the output without delay. This continuous real-time scanning makes the system highly useful for moving robots, vehicles and blind navigation support. The methodology is simple, effective and consumes very low power, making the system suitable for long-term deployment.

Hardware & Software Requirements

For successful development and execution of this project, both hardware components and software tools were required. Each part had a specific role in making the system functional and efficient. The project was carefully designed using easily available components so that it remains low-cost, beginner-friendly and replicable for academic as well as practical use.

Hardware Requirements

1. **Arduino UNO (Microcontroller Board)**
   The heart of the entire circuit. It receives sensor data, processes it and controls output devices like LEDs and the buzzer. The UNO version was chosen because it is simple to program, has multiple digital/analog pins, and works directly over USB power without external circuitry.
2. **HC-SR04 Ultrasonic Sensor**
   This sensor was used to measure distance by sending ultrasonic waves and capturing their reflections. Its range of about 2–400 cm makes it suitable for short-distance detection used in this project. It has separate TRIG and ECHO pins for transmitting and receiving signals.
3. **LED Indicators (Red, Yellow, Green)**
   Three LEDs were used to visually represent the distance status.
   - Green indicates safe distance
   - Yellow represents moderate or warning distance
   - Red along with buzzer indicates danger or very close obstacle
4. **Buzzer**
   Provides audible alert to warn the user when obstacle enters danger zone. Useful especially when visual attention might not be possible.
5. **Mini Breadboard**
   A solderless board used for connecting components temporarily during testing. This allows easy adjustments without permanent wiring.
6. **220Ω Resistors**
   Resistors were connected in series with LEDs to limit current flow. They prevent LEDs from burning out and maintain safe operating levels.
7. **Jumper Wires**
   All circuit connections were made with male-to-male jumper wires. Their flexibility allowed easy rearrangement and troubleshooting.

Software Requirements

1. **Arduino IDE**
2. **C/C++ Programming**
3. **USB Drivers (Auto Installed)**

Arduino IDE Code:

```
// Project: Ultrasonic Obstacle Detection and Alert System
// Description: Measures distance to an obstacle using HC-SR04 and provides
//              tiered alerts via LEDs and a buzzer based on distance thresholds.
// -------- PIN SETUP --------
const int trigPin = 8;
const int echoPin = 9;
// LEDs
const int redLED = 2;    // too close (<= 10 cm)
const int yellowLED = 5; // medium (10 cm to 25 cm)
const int greenLED = 4;  // safe (> 25 cm)
// Buzzer
const int buzzerPin = 3;
void setup() {
  Serial.begin(9600);
  Serial.println("Ultrasonic Alert System Initialized.");
  // Set sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  // Set output pins
  pinMode(redLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  // Initial state: turn everything off
  digitalWrite(redLED, LOW);
```

```
  digitalWrite(yellowLED, LOW);
  digitalWrite(greenLED, LOW);
  digitalWrite(buzzerPin, LOW);
}
// Function to calculate distance in centimeters (cm)
long getDistance() {
  // Clear the trigPin by setting it LOW for a moment
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Send a 10 microsecond pulse to trigger the sensor
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Measure the duration (Time-of-Flight) of the echo pulse
  // Timeout added (50000 µs) for stability if no object is found
  long duration = pulseIn(echoPin, HIGH, 50000);

  // Check if a timeout occurred (duration is 0 if pulseIn times out)
  if (duration == 0) return -1;

  // Calculate the distance: Time * Speed_of_Sound / 2
  // Speed of sound is approx 0.034 cm/µs
  long distance = duration * 0.034 / 2;
  return distance;
}
void loop() {
  long d = getDistance();
```

```
Serial.print("Distance: ");

Serial.print(d);

Serial.println(" cm");


// Reset all outputs before checking the new distance

digitalWrite(redLED, LOW);

digitalWrite(yellowLED, LOW);

digitalWrite(greenLED, LOW);

digitalWrite(buzzerPin, LOW);


// If reading failed (d < 0), skip the logic for this cycle

if (d < 0) {

  Serial.println("Reading failed (Timeout).");

  return;

}

// --- Alert Logic ---

if (d <= 10) {

  // DANGER ZONE (<= 10 cm) -> Red + Buzzer

  digitalWrite(redLED, HIGH);

  digitalWrite(buzzerPin, HIGH);

  Serial.println("ALERT: DANGER - CRITICAL PROXIMITY!");

} else if (d > 10 && d <= 25) {

  // WARNING ZONE (10 cm to 25 cm) -> Yellow

  digitalWrite(yellowLED, HIGH);

  Serial.println("Warning: Caution Range.");
```

```
  } else {
    // SAFE ZONE (> 25 cm) -> Green
    digitalWrite(greenLED, HIGH);
    Serial.println("Safe: Clear Path.");
  }
  // Delay before the next reading
  delay(200);
}
```

Working Flow & Circuit Explanation

The working flow of the project describes how data moves through the system from start to output. As soon as the Arduino receives power, it activates the ultrasonic sensor and begins continuous scanning of the surroundings. The Trig pin sends a short burst of sound waves, and the Echo pin measures how long it takes for the waves to return after striking an object. Based on this measured time, the Arduino calculates the distance using the programmed formula. Once the distance is known, the output module is activated according to the defined threshold values.

The circuit is designed in such a way that each component is connected to the Arduino in a systematic manner. The VCC and GND of the ultrasonic sensor are connected to the 5V and GND pins of the Arduino. The Trig pin is connected to a digital output pin, which allows the microcontroller to send trigger pulses, and the Echo pin is connected to an input pin where reflected signals are received and timed. Three LEDs representing Safe, Warning, and Danger are connected through 220Ω resistors to prevent current damage. These LEDs are linked to separate output pins so that they can be controlled individually based on distance readings. The buzzer is also connected to an output pin so that it only activates when the object is too close.

To understand the circuit more clearly, consider the complete flow below:

Power ON → Ultrasonic Trigger Pulse → Echo Received → Arduino Calculates

Testing / Output Results

After completing the assembly and coding of the project, various tests were conducted to ensure that the system worked accurately in different conditions. Testing was done using multiple objects including a hand, notebook, bottle, box and even a soft cushion to observe how the ultrasonic waves respond to different surfaces. The distance threshold values were checked repeatedly to confirm that the LEDs switched at the correct range and the buzzer activated only when the object was very near.

During initial testing, the distance readings were monitored using the Serial Monitor of Arduino IDE. This helped in adjusting the threshold levels more precisely. Once the values were set, the Serial Monitor was no longer required and the LED indicators alone were enough to understand system behavior. The project behaved consistently and responded instantly as soon as any object entered the sensing range.

The green LED remained active when no object was in the danger zone, while the yellow LED turned on when the object moved closer. When the object reached the critical short-range limit, the red LED and buzzer activated together, making it very clear that attention was required.

The final test results can be summarized as follows:

Distance Range	System Output	Observation
> 25 cm	Green LED ON	Safe zone, no danger detected
10–25 cm	Yellow LED ON	Object approaching, caution suggested
< 10 cm	Red LED + Buzzer ON	High-risk zone, immediate action needed

Applications

This project has a very wide range of applications because obstacle detection is required almost everywhere movement takes place. The system is simple, compact and works entirely on real-time sensing, making it suitable not only for small-scale academic use but also for future industrial and automation-based solutions. Since it alerts the user before an object comes too close, it can effectively reduce chances of accidents and damage.

One of the most common areas of application is **vehicle parking assistance**, where drivers often face difficulty while reversing in tight spaces or blind spots. By installing sensors at the rear side of a car, the system can warn the driver when they get too close to a wall or another vehicle. Similarly, it can be used in **autonomous robots and automated guided vehicles (AGVs)** to help them move safely without hitting obstacles.

The project can also support **visually impaired individuals** by helping them identify obstacles in front of them. Instead of LEDs, vibration motors or voice alerts can be used so that they can sense the surroundings without needing vision. In **smart homes**, this system may assist in automatic door safety, staircase alerts, pet monitoring, and security alarms. Another strong field of application is **industrial safety**, where heavy machines or moving belts often require alert systems to prevent accidents.

To summarize, the practical applications include:

- Vehicle parking and reversing alert
- Obstacle avoidance in robots and drones
- Blind assistance walking tools
- Warehouse and industrial automation safety
- Smart home security and automation devices
- Research and educational embedded projects

Because of its flexibility and low cost, the same circuit can be redesigned into different products depending on the need. This makes the project useful and adaptable for many real-world purposes.

Future Scope

Although the project is simple and fully functional in its current form, it holds a wide potential for future development. The basic working concept of obstacle detection can be enhanced with more advanced features to increase accuracy, usability and application areas. As technology grows, this system can be upgraded with additional modules and software improvements, making it suitable even for commercial use.

One of the most promising enhancements is the addition of a **servo motor to rotate the ultrasonic sensor**, allowing it to scan in multiple directions instead of focusing only on a straight line. This would convert the system into a small radar-like scanner capable of generating a 180° or 360° surrounding map. Another upgrade could include using an **LCD or OLED display** to show real-time distance readings on screen, which would make the system more user-friendly and informative.

For remote monitoring and automation, the project can be integrated with **IoT modules such as ESP8266 or ESP32** to send distance alerts wirelessly to a smartphone or computer. In critical applications like blind assistance or autonomous robots, **voice output or vibration feedback** can be added to ensure alerts are received without needing to see LEDs. Further enhancement may include implementing **machine learning algorithms** to predict movement patterns, identify frequently occurring obstacles, or even make automatic navigation decisions.

Future upgradable ideas include:

- Servo-based angular scanning & radar mapping
- IoT connectivity for wireless alerts and cloud logging
- Distance display on LCD/OLED screens
- Bluetooth or Wi-Fi based smartphone notification system
- Voice or haptic feedback for blind navigation
- AI-based object recognition and automated control actions

With these improvements, the project can evolve from a simple prototype to a sophisticated real-world safety device. The flexibility of Arduino and the modular nature of the circuit ensure that upgrades can be added without rebuilding the entire system. This gives the design strong long-term future potential.

Conclusion

The completion of this project provided a practical understanding of how sensors and microcontrollers work together to create intelligent systems. The ultrasonic-based obstacle detection module successfully measured distance in real time and classified it into different safety levels. The use of three LEDs and a buzzer made the alerts easy to understand, even for someone who sees the system for the first time. Green indicated a safe condition, yellow acted as a caution sign, and red combined with the buzzer gave an immediate danger warning. This simple yet effective design helped the model achieve its goal of avoiding collisions and improving safety awareness.

References

During the development of this project, various study materials, online resources and technical documents were referred to in order to understand the working principles of ultrasonic sensing, Arduino programming and real-time embedded system execution. These references helped in learning the distance calculation method, pin configuration, timing functions and correct way of interfacing modules.

Some of the most useful resources included the official Arduino documentation and multiple electronics project forums that explained real examples of ultrasonic distance measurement. YouTube tutorials and GitHub project codes were also helpful during debugging and understanding how threshold-based alert systems work in practical scenarios. Along with that, theoretical notes from classroom lectures and PDF manuals contributed to clearing basic concepts of microcontrollers and serial communication.

Below is a list of major references used throughout the project:

1. **Arduino Official Documentation** – www.arduino.cc
   Used for syntax, programming functions and board configuration guidelines.
2. **HC-SR04 Ultrasonic Sensor Datasheet**
   Helped in understanding trigger/echo timing, sensing range and technical operating values.
3. **Electronics Tutorials Websites**
   Basic circuit understanding, resistor use with LEDs and prototyping guides.
4. **YouTube Project Demonstrations & Tutorials**
   Useful during sensor testing, connection verification and output observation.
5. **GitHub and Arduino Community Forums**
   Reference for troubleshooting logic errors and improving code efficiency.
6. **Class Notes & Lab Guidance Material**
   Provided foundational understanding of microcontrollers and embedded interfacing.