DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka 562112

Bachelor of Technology
in

COMPUTER SCIENCE AND ENGINEERING
(Artificial Intelligence and Machine Learning)

Mini Project

Light Detection Alarm

By

Sana Banu - ENG22AM0053
Sahana Priya G - ENG22AM0050
N.Dharsini - ENG22AM0036
Pooja N P – ENG23AM1002

Under the supervision of

Dr. Mude Nagarjuna Naik, & Prof. Pradeep Kumar K

Assistant Professor, Artificial Intelligence & Machine Learning, SOE

Prof. Abhinav Karan

Assistant Professor, Artificial Intelligence & Machine Learning, SOE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence and Machine Learning)
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY, BANGALORE

School of Engineering
Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)

Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka 562112

CERTIFICATE

This is to certify that the Special topic titled **"Light Detection Alarm"** is carried out by **Sana Banu (ENG22AM0053), Sahana Priya G (ENG22AM0050), N.Dharsini (ENG22AM0036), Pooja NP (ENG23AM1002),** bonafide students of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) at the School of Engineering, Dayananda Sagar University.

Dr. Mude Nagarjuna Naik
Assistant/Associate/ Professor
Dept. of CSE(AI&ML),
School of Engineering
Dayananda Sagar University

Date:

Dr. Jayavrinda Vrindavanam
Chairperson CSE(AI&ML)
School of Engineering
Dayananda Sagar University

Date:

Prof. Pradeep Kumar K
Assistant/Associate/ Professor
Dept. of CSE(AI&ML),
School of Engineering
Dayananda Sagar University

Date:

Prof. Abhinav Karan
Assistant/Associate/ Professor
Dept. of ECE,
School of Engineering
Dayananda Sagar University

Date:

DECLARATION

We, **Sana Banu (ENG22AM0053), Sahana Priya G (ENG22AM0050), N.Dharsini (ENG22AM0036), Pooja NP (ENG23AM1002)** are students of fourth semester B. Tech in **Computer Science and Engineering with speciation in Artificial intelligence and machine learning** , at School of Engineering, **Dayananda Sagar University**, hereby declare that Embedded System Project titled **"Light Detection Alarm"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2023-2024**.

Place: Bangalore

Date:

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this projectwork.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Jayvrinda Vrindavanam v, Department Chairman****,* ***Computer Science and Engineering (AI AND ML)****,* ***Dayananda Sagar University,*** *for providing right academic guidance that made the task possible.*

We would like to thank our guide ***Dr. Mude Nagarjuna Naik, & Prof. Pradeep Kumar K, Assistant Professor, Dept. of Computer Science and Engineering (Artificial Intelligence and Machine Learning), and Prof. Abhinav Karan Dept. of Electronics & Communication Engineering,*** *for sparing their valuable time to extend help in every step of our Mini project work towards Embedded System Design Course, which paved the way for smooth progress and the fruitful culmination of the research.*

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Embedded System Project.

TABLE OF CONTENT

LIST OF FIGURES ……….. vi

ABSTRACT 1

CHAPTER 1 INTRODUCTION……...… 2

CHAPTER 2 PROBLEM DEFINITION ………... 4

CHAPTER 3 LITERATURE REVIEW... 5

CHAPTER 4 OBJECTIVE………………………………………………………….. 6

CHAPTER 5 PROJECT DESCRIPTION.. 7

CHAPTER 6 METHODOLOGY.. 11

CHAPTER 7 RESULTS AND ANALYSIS …... 12

CHAPTER 8 CODE……………………………………………………………….

CHAPTER 9 CONCLUSION………………... 14

CHAPTER 10 REFERENCES.. 15

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
5.2	Audino board	7
5.3	Light sensor (potentiometer)	8
5.4	LED	8
5.5	Push button	9
5.6	Resistors	9
5.7	Breadboard	10
5.8	Circuit diagram	10
8.1	Result Output	16

ABSTRACT

This project involves designing, implementing, and simulating a light detection alarm system using an Arduino Uno microcontroller. The main goal is to monitor ambient light levels and trigger an alarm when the light intensity exceeds a set threshold. The system uses a potentiometer to simulate different light intensities, a push button to reset the alarm, and an LED to indicate the alarm state. The software, written in system Verilog -based language, reads the analog input from the potentiometer, compares it to the threshold, and controls the LED accordingly. The push button allows for manual resetting of the alarm. We used Circuits.io, an online simulation tool, to design and test the circuit. This simulation confirmed that the system accurately detects light intensity, activates the alarm when needed, and resets properly. Overall, this project demonstrates how to create a basic light detection alarm system. It shows how hardware and software components can work together and provides a good foundation for more advanced light detection systems.

CHAPTER 1

INTRODUCTION

A light detection alarm is a system designed to monitor ambient light levels and trigger an alert when the light intensity exceeds or falls below a certain threshold. This type of system can be used in various applications such as security, environmental monitoring, and automated lighting control. By using sensors to detect changes in light, the system can provide real-time responses to environmental changes, enhancing safety and efficiency. With the rise of smart systems and automation, being able to detect and respond to environmental changes is becoming more important in our daily lives. This project focuses on designing, building, and simulating a light detection alarm system using an Arduino Uno microcontroller. The main goal is to create a system that monitors light levels and triggers an alarm when the intensity gets too high. This demonstrates how sensors and microcontrollers can work together in practical applications. The light detection alarm system can be useful in various situations, such as improving security by detecting unexpected lighting changes or managing lighting in places like labs or art galleries. By simulating different light levels, the system can react to changes in real time and provide immediate feedback and alerts. For the hardware, we used a potentiometer to simulate different light intensities, a push button to reset the alarm, and an LED to show when the alarm is active. The Arduino Uno reads the input from the potentiometer, compares it to a set threshold, and turns the LED on or off based on this comparison. The push button allows us to manually reset the alarm when needed. Monitoring and responding to changes in environmental light levels is essential for various applications, ranging from security systems to ambient light management in sensitive areas. The primary objective is to create a system that continuously monitors light intensity and triggers an alarm when the light level exceeds a predefined threshold, showcasing the practical application of sensors and microcontrollers.

1.1 Importance of Light Detection Systems

Light detection systems play a crucial role in numerous fields. In security applications, these systems can detect unauthorized lighting changes, alerting security personnel to potential intrusions. In art galleries or laboratories, maintaining optimal light conditions is vital to preserve sensitive materials or conduct precise experiments. By implementing a light detection alarm system, we can automate the monitoring process, ensuring timely and accurate responses to changes in light levels. This automation enhances the efficiency and reliability of light management in various settings.

1.2 Significance and Applications

Light detection alarms are crucial in several fields. In security systems, they can detect unexpected lighting changes, such as the use of flashlights by intruders, and trigger alarms to alert security personnel. In environmental monitoring, these systems help maintain optimal light conditions in places like greenhouses or research labs, where specific light levels are essential for the growth of plants or the accuracy of experiments. Additionally, light detection alarms can automate lighting control in smart homes or offices, adjusting artificial lighting based on natural light levels to save energy and improve comfort.

1.3 Design and Implementation

The light detection alarm system comprises several key components: a potentiometer to simulate varying light intensities, a push button to reset the alarm, and an LED to indicate the alarm status. The Arduino Uno microcontroller is the heart of the system, reading the input from the potentiometer, comparing it to a set threshold, and controlling the LED based on this comparison. The push button allows users to manually reset the alarm, adding an element of user control. We designed and tested the circuit using Circuits.io, an online simulation tool that facilitates the creation and verification of electronic circuits. This simulation ensured that the system operates correctly before physical implementation, allowing us to troubleshoot and refine the design efficiently.

CHAPTER 2

PROBLEM DEFINITION

The need for maintaining appropriate light levels in various environments, such as security-sensitive areas, places with sensitive materials, and energy-efficient spaces, necessitates an automated monitoring system. Manual light monitoring is inefficient and prone to errors, highlighting the demand for a system that can continuously track ambient light intensity and provide immediate alerts when levels deviate from set thresholds.

This project aims to develop a light detection alarm system using an Arduino Uno microcontroller, capable of detecting changes in light intensity, comparing them against predefined thresholds, and triggering an alarm when necessary. The system will also feature a manual reset function for user control. By ensuring continuous monitoring, accurate detection, real-time alerts, and ease of use, this light detection alarm system will enhance light management, security, and energy efficiency across various applications.

CHAPTER 3

LITERATURE REVIEW

the smart light detection and monitoring system is presented. The light sensors are used to detect the light temperature value in the room. The Arduino UNO and ESP32 are used as the main embedded systems. The ZigBee transmitters are used as the main transmitter. MySQL is set as the database. Moreover, Machine Learning is used to classify the light temperature value to natural light or fluorescent light. Test result shows that the cross-validation accuracy of KNN is 0.996. This accuracy of KNN is greater than other algorithms. Moreover, this system can be used to examine the status of the light in the room that can solve the problem of forgetting to turn off the light and can analyze data of using light as well.[1]

Spectral power distribution (SPD) is the radiation power intensity at different wavelengths, containing the most basic photometric and colorimetric performance of the illuminant, which is able to predict the lifetime of LEDs. An SPD model assisted by machine learning algorithms to detect the early failure of white LEDs. The SPD features of 3W high-power white LEDs were firstly extracted by the statistical models of Gaussian, Lorentz, and Asym2sig functions. An unsupervised learning method, principal component analysis (PCA), was then used to reduce the extracted features parameters' dimensions. Next a K-nearest neighbor (KNN)-based method was used to detect LEDs' anomalies by dividing the main cluster into groups, and estimating the distance from the center of mass of each cluster to the test point.[2]

visible light positioning system in which a receiver performs position estimation based on signals emitted from a number of light emitting diode (LED) transmitters. Each LED transmitter can be malicious and transmit at an unknown power level with a certain probability. A maximum likelihood (ML) position estimator is derived based on the knowledge of probabilities that LED transmitters can be malicious.[3]

CHAPTER 4

OBJECTIVES

The primary objective of this project is to design and implement a light detection alarm system using an Arduino Uno microcontroller. The system aims to monitor ambient light levels and trigger an alarm when the light intensity exceeds a predefined threshold. This project will provide practical insights into sensor integration, microcontroller programming, and circuit design.

Continuous Light Monitoring

- To ensure the system can continuously and accurately monitor ambient light levels.
- The light sensor will provide real-time data to the Arduino, enabling continuous tracking of light intensity changes.

Automated Alarm Triggering

- To develop an automated response mechanism that activates an alarm when light intensity exceeds a set threshold.
- The Arduino will compare the sensor readings to the threshold and trigger an LED alarm when necessary, demonstrating the system's automation capabilities.

Manual Reset Functionality

- To incorporate a manual reset feature allowing users to reset the alarm state easily.
- A push button will be used to reset the alarm, providing user control and enhancing the system's usability.

System Testing and Validation

- To test and validate the system using simulation tools before physical implementation.

CHAPTER 5
PROJECT DESCRIPTION

This project focuses on developing a light detection alarm system using an Arduino Uno microcontroller. The system continuously monitors ambient light intensity and triggers an alarm when the light level exceeds a predefined threshold. The alarm is indicated by an LED, and a push button is provided to manually reset the alarm. The primary objective of this project is to create a reliable and efficient system that can be used in various applications such as security, environmental monitoring, and automated lighting control.

5.1 REQUIREMENTS:

Arduino Board: The Arduino board is a versatile microcontroller platform used for building digital devices and interactive objects, capable of reading inputs from sensors and controlling outputs like motors and displays.

Fig 5.2: Arduino Board

Light sensor: A light sensor (potentiometer) in the light detection alarm to simulate varying light intensities and monitor ambient light levels continuously. This allows the system to detect when light levels exceed a predefined threshold and trigger an alarm. The potentiometer provides an easy way to adjust and test different light conditions. Its integration with the Arduino enables precise and automated light monitoring and response.

Fig 5.3: Light Sensor (Potentiometer)

LED: We use an LED in the light detection alarm to provide a visual indication when the alarm is triggered. The LED lights up when the ambient light intensity exceeds the predefined threshold, alerting users to the condition. It serves as a simple, immediate, and easily visible signal. Additionally, LEDs are energy-efficient and have a long lifespan, making them ideal for continuous operation.

Fig 5.4 LED

Push Button: We use a push button in the light detection alarm system to allow manual resetting of the alarm state. This enables users to easily turn off the alarm once it has been triggered and acknowledge the alert. The push button provides a simple and user-friendly interface for controlling the system. Its inclusion enhances the overall usability and functionality of the light detection alarm.

Fig 5.5 Push Botton

Resistors: Resistors are used in the light detection alarm to limit the current flowing through the LED, preventing it from burning out. They are also used in pull-down configurations to ensure stable signals from the push button, preventing false triggers. Additionally, resistors help maintain the correct operating conditions for the light sensor and other components. Overall, they ensure the circuit functions reliably and safely.

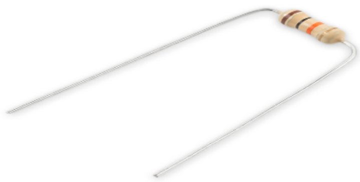


Fig 5.6 Resistor

Breadboard: We use a breadboard in the light detection alarm project to facilitate easy and temporary connections between components without soldering. It allows us to quickly assemble and modify the circuit, enabling efficient testing and troubleshooting. The breadboard also provides a convenient way to prototype and experiment with different configurations. This flexibility is crucial for iterating on the design before finalizing the circuit.

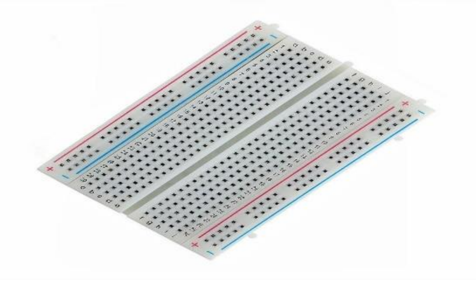


Fig 5.7 Bread board

Using these many components in the light detection alarm system ensures that each aspect of the system is addressed comprehensively. The Arduino Uno processes inputs and controls outputs, the light sensor provides real-time data, the LED and push button offer clear user interface features, and the breadboard and jumper wires enable a flexible and modifiable setup. Together, these components create a robust, efficient, and user-friendly light detection alarm system.

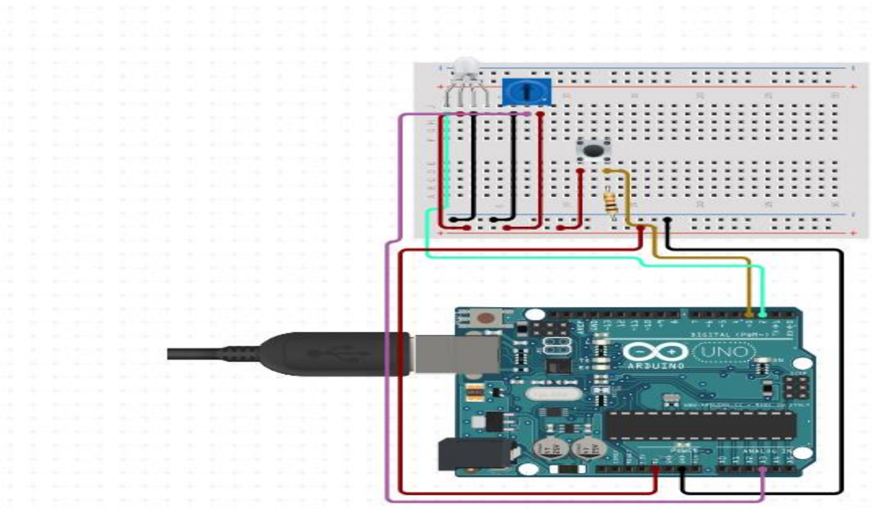


Fig 5.8 Circuit Diagram

CHAPTER 6

METHODOLOGY

The development of the light detection alarm system involves a series of systematic steps to ensure the creation of a reliable and effective prototype. This process encompasses the selection of components, circuit design, comprehensive programming, and rigorous testing and validation.

- **Component Selection and Circuit Design**

The first step in the project is the selection of appropriate components to build the system. An Arduino Uno microcontroller is chosen for its ease of use and widespread availability. A potentiometer is used to simulate varying light intensities, serving as the light sensor. An LED is employed as the visual indicator for the alarm, while a push button is incorporated to allow manual resetting of the alarm state. Resistors are used for current limiting and pull-down purposes to ensure stable operation of the LED and button. All these components are connected on a breadboard using jumper wires, facilitating easy and temporary connections without soldering. The breadboard setup allows for quick modifications and troubleshooting, which is essential during the initial stages of prototyping.

- **Programming, Simulation, and Testing**

Once the circuit is designed, the next step is to develop the Arduino code that will control the system. The code initializes the necessary variables and pins, reads the light sensor values, and compares these values to a predefined threshold to determine whether the alarm should be triggered. If the light intensity exceeds the threshold, the alarm state is set to true, and the LED is turned on. The push button functionality is also implemented to allow users to reset the alarm manually. After coding, the system is simulated using Circuits.io to verify its behavior and functionality. This virtual testing helps identify and resolve any issues before physical assembly. Finally, the circuit is built on a breadboard, and the code is uploaded to the Arduino for real-world testing.

CHAPTER 7

RESULT AND ANALYSIS

The light detection alarm system was tested and analyzed to evaluate its performance and effectiveness. The results indicate that the system meets the design objectives, providing accurate light monitoring, automated alarm triggering, and manual reset functionality. The analysis is presented under four subheadings: System Performance, Testing and Validation, Reliability and User Control, and Potential Improvements.

7.1 System Performance

The light detection alarm system successfully monitored ambient light levels using a potentiometer as the light sensor. The Arduino Uno microcontroller processed the sensor data and compared it to a predefined threshold. When the light intensity exceeded the threshold, the system accurately triggered the LED alarm. The manual reset button functioned as expected, allowing users to reset the alarm state easily.

7.2 Testing and Validation

Extensive testing was conducted to ensure the system's reliability under various light conditions:

- Accuracy of Light Detection: The potentiometer provided consistent analog readings, and the system reliably triggered the alarm when the light intensity crossed the threshold value of 500.
- Alarm Response Time: The LED responded immediately to changes in light intensity, with no noticeable delay between detection and alarm activation.
- Manual Reset Functionality: The push button effectively reset the alarm, turning off the LED and enabling the system to resume monitoring. This feature worked consistently across multiple tests.

7.3 Reliability and User Control

The system demonstrated high reliability and user control throughout the testing phase:

- Reliability: The light sensor and Arduino Uno ensured precise and consistent performance. The system accurately detected light intensity changes and responded accordingly.
- User Control: The manual reset button added valuable user control, allowing easy management of the alarm state. This feature enhances the system's practical application and usability.

7.4 Potential Improvements

While the system performed well, there are areas for future enhancement:

- Advanced Light Sensors: Integrating more sophisticated light sensors could improve accuracy and sensitivity, making the system more robust.
- Wireless Communication: Adding wireless modules could enable remote monitoring and control, increasing the system's versatility and functionality.
- Energy Efficiency: Implementing power-saving features could enhance the system's efficiency, especially for battery-operated applications.
- Alarm Customization: Allowing users to set custom light intensity thresholds and alarm durations could improve user experience and adaptability.

CHAPTER 8

CODE

Test Bench Module (tb_light_detection_alarm.sv)

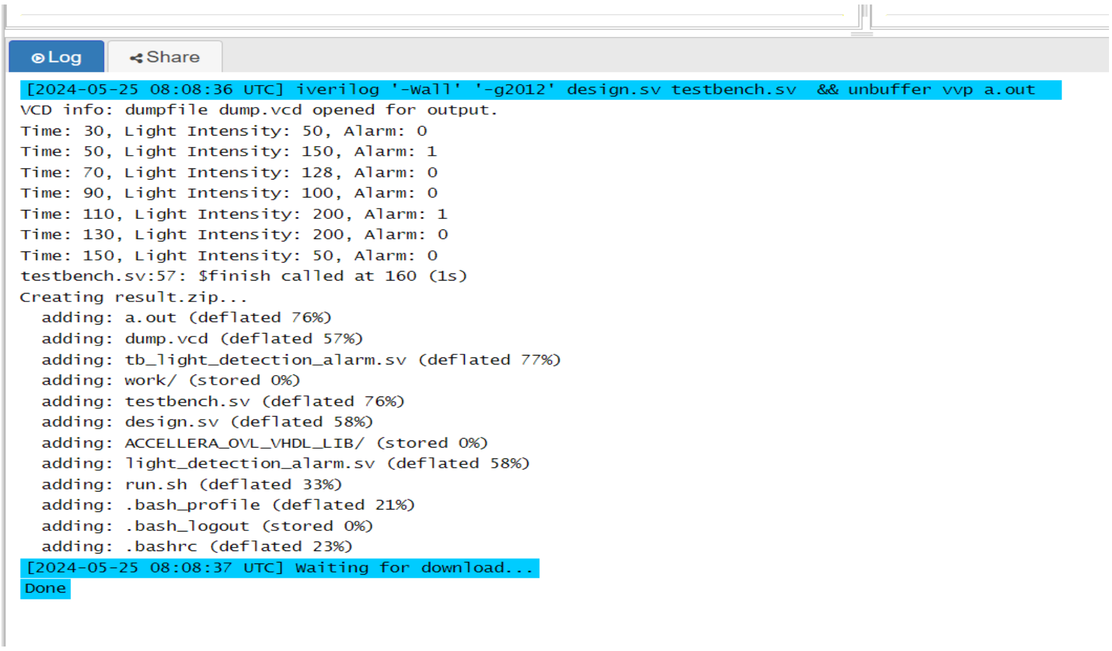
```
module tb_light_detection_alarm;
    logic clk;
    logic reset_n;
    logic [7:0] light_intensity;
    logic alarm;
    // Instantiate the light detection alarm module
    light_detection_alarm uut (
        .clk(clk),
        .reset_n(reset_n),
        .light_intensity(light_intensity),
        .alarm(alarm));
    // Clock generation
    always #5 clk = ~clk;
    initial begin
        // Initialize signals
        clk = 0;
        reset_n = 0;
        light_intensity = 0;
        $dumpfile("dump.vcd");
        $dumpvars(0, tb_light_detection_alarm);
        #10 reset_n = 1;
        // Test case 1: Light intensity below threshold
        #10 light_intensity = 50;
        #10 $d
isplay("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time,   light_intensity, alarm);
        // Test case 2: Light intensity above threshold
        #10 light_intensity = 150;
```

```
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    // Test case 3: Light intensity at threshold
    #10 light_intensity = 128;
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    // Test case 4: Light intensity fluctuating
    #10 light_intensity = 100;
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    #10 light_intensity = 200;
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    // Test case 5: Reset during operation
    #10 reset_n = 0;
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    #10 reset_n = 1;
    light_intensity = 50;
    #10 $display("Time: %0t, Light Intensity: %0d, Alarm: %0b", $time, light_intensity, alarm);
    // Finish the simulation
    #10 $finish;
  end
endmodule
```

<u>Design Module (light_detection_alarm.sv)</u>

```
module light_detection_alarm (
  input logic clk,                  // Clock signal
  input logic reset_n,              // Active low reset
  input logic [7:0] light_intensity, // Light intensity input
  output logic alarm );             // Alarm output
```

```
// Threshold for triggering the alarm
   localparam int THRESHOLD = 128
// Always block to detect light intensity and trigger alarm
   always_ff @(posedge clk or negedge reset_n) begin
      if (!reset_n) begin
         alarm <= 0;
      end else begin
         if (light_intensity > THRESHOLD) begin
            alarm <= 1;
         end else begin
            alarm <= 0;
         end
      end
   end
endmodule
```



```
[2024-05-25 08:08:36 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
Time: 30, Light Intensity: 50, Alarm: 0
Time: 50, Light Intensity: 150, Alarm: 1
Time: 70, Light Intensity: 128, Alarm: 0
Time: 90, Light Intensity: 100, Alarm: 0
Time: 110, Light Intensity: 200, Alarm: 1
Time: 130, Light Intensity: 200, Alarm: 0
Time: 150, Light Intensity: 50, Alarm: 0
testbench.sv:57: $finish called at 160 (1s)
Creating result.zip...
  adding: a.out (deflated 76%)
  adding: dump.vcd (deflated 57%)
  adding: tb_light_detection_alarm.sv (deflated 77%)
  adding: work/ (stored 0%)
  adding: testbench.sv (deflated 76%)
  adding: design.sv (deflated 58%)
  adding: ACCELLERA_OVL_VHDL_LIB/ (stored 0%)
  adding: light_detection_alarm.sv (deflated 58%)
  adding: run.sh (deflated 33%)
  adding: .bash_profile (deflated 21%)
  adding: .bash_logout (stored 0%)
  adding: .bashrc (deflated 23%)
[2024-05-25 08:08:37 UTC] Waiting for download...
Done
```

Fig 8.1 result output

CHAPTER 8

CONCLUSION

The light detection alarm system effectively meets its design objectives by providing accurate and reliable monitoring of ambient light levels. Through the use of an Arduino Uno, potentiometer, LED, and manual reset button, the system can detect light intensity changes and trigger an alarm when levels exceed a set threshold. Extensive testing has validated the system's accuracy, prompt response time, and reliable reset functionality, demonstrating its practical application in various environments. While the project successfully achieves its goals, future improvements such as integrating advanced light sensors, adding wireless communication capabilities, and enhancing energy efficiency could further enhance its functionality and versatility. Overall, this project provides a solid foundation for developing more sophisticated automated monitoring systems, contributing valuable insights into sensor integration, microcontroller programming, and circuit design.

CHAPTER 9

REFERENCES

[1] TL-Detector: Lightweight Based Real-Time Traffic Light Detection Model for Intelligent Vehicles Publisher: IEEE by Zikai Yao; Qiang Liu; Qian Xie; Qing L
TL-Detector: Lightweight Based Real-Time Traffic Light Detection Model for Intelligent Vehicles | IEEE Journals & Magazine | IEEE Xplore

[2] Sound-Light Alarm System for Traffic Based on Human Factors Engineering Publisher: IEEE by Yifan Wu; Caifeng Zhang; Pengfei Wang Sound-Light Alarm System for Traffic Based on Human Factors Engineering | IEEE Conference Publication | IEEE Xplore

[3] Position Estimation in Visible Light Systems in the Presence of Malicious LED Transmitters, Publisher: IEEE by Furkan Kokdogan; Sinan Gezici
Position Estimation in Visible Light Systems in the Presence of Malicious LED Transmitters | IEEE Conference Publication | IEEE Xplore

[4] Research and Application of Photovoltaic Power Station On-line Hot Spot Detection Operation and Maintenance System Based on Unmanned Aerial Vehicle Infrared and Visible Light Detection, Publisher: IEEE by Guanglei Li; Yuejiao Wang; Zheng Xu; Wei Teng; Xingyou Zhang
Research and Application of Photovoltaic Power Station On-line Hot Spot Detection Operation and Maintenance System Based on Unmanned Aerial Vehicle Infrared and Visible Light Detection | IEEE Conference Publication | IEEE Xplore

[5] Visible Light Positioning in the Presence of Malicious LED Transmitters, Publisher: IEEE by Furkan Kokdogan; Sinan Gezici Visible Light Positioning in the Presence of Malicious LED Transmitters | IEEE Journals & Magazine | IEEE Xplore

[6] Machine learning assisted early anomaly detection of LEDs with spectral power distribution modeling by Publisher: IEEE Furkan Kokdogan; Sinan Gezici
Visible Light Positioning in the Presence of Malicious LED Transmitters | IEEE Journals & Magazine | IEEE Xplore