

**COLLEGE OF COMPUTING AND INFORMATION SCIENCES**

**DEPARTMENT OF COMPUTER SCIENCE**

**GROUP 15**

|  |  |  |
| --- | --- | --- |
| **NAME** | **REG. NO** | **STUDENT N0.** |
| Mugisha Balam | 23/U/11585/EVE | 2300711585 |
| Namanya Rinah | 23/U/14355/EVE | 2300714355 |
| Musasizi Junior | 23/U/12116/PS | 2300712116 |
| Kalyango Mark Maurice | 23/U/08815/PS | 2300708815 |
| Munyakazi Natasha Aaliyah | 23/U/0867 | 2300700867 |

**DISTRIBUTED LIGHT INTENSITY MONITORING SYSTEM PROJECT REPORT**

**Abstract**

This project demonstrates the implementation of a Distributed Light Intensity Monitoring System using Arduino microcontrollers. The system measures ambient light intensity at multiple locations and transmits the data to a central hub for analysis and display. The application is designed for environmental monitoring in use cases such as optimizing lighting in greenhouses, adjusting artificial lighting in buildings, and monitoring light pollution levels. The system incorporates functionalities like ADC, GPIOs, timers, interrupts, EEPROM storage, and serial communication to ensure efficient and reliable data collection and transmission.

**Objective**

The main objective of this project is to:

1. Measure ambient light intensity at three distinct locations using Arduino-based sensor nodes.
2. Transmit the readings to a central Arduino hub for processing and display.
3. Optimize light monitoring for environmental and energy-saving applications.
4. Utilize core Arduino functionalities, including ADC, GPIO, timers, interrupts, EEPROM, and serial communication, to build a robust and scalable system.

**Project Requirements**

**Hardware Components**

1. **Arduino Uno/Nano (x4)**: Used as the central hub and light sensor nodes.
2. **LDR or Photodiodes (x3)**: For measuring ambient light intensity.
3. **10kΩ Resistors (x3)**: To create voltage divider circuits for the LDRs.
4. **Push Buttons (x4)**: For toggling between day/night modes or resetting data.
5. **LCD Display with I2C Module**: To display light intensity readings at the central hub.
6. **LEDs (x3)**: To indicate successful data transmission from sensor nodes.
7. **Debouncing Capacitors (0.1 µF)**: To stabilize button input.
8. **Wires, Breadboards, and Power Supplies**: For circuit setup.

**Software Tools**

1. **Arduino IDE**: For coding and programming the boards.
2. **Serial Monitor**: For testing and debugging serial communication.

**System Design**

**Functional Architecture**

1. **Central Hub (Arduino Board 1)**:
   * Receives data from sensor nodes using serial communication.
   * Displays the data on an LCD screen.
   * Logs readings to a computer for further analysis.
2. **Sensor Nodes (Arduino Boards 2, 3, and 4)**:
   * Measure light intensity using LDRs or photodiodes.
   * Transmit readings to the central hub at periodic intervals.
   * Use a button to toggle between sampling modes (e.g., day and night).

## ****Implementation****

### ****Step-by-Step Procedure****

#### **Hardware Setup**

1. Connect the LDRs to analog input pins of Arduino Boards 2, 3, and 4 using a voltage divider circuit.
2. Attach LEDs to digital GPIO pins on each sensor node to indicate successful data transmission.
3. Wire push buttons with debouncing capacitors to digital pins of each sensor node.
4. Connect an LCD (16x2) to the central hub (Arduino Board 1) using I2C for data display.
5. Link the central hub and sensor nodes via serial communication (TX/RX lines).

#### **Software Implementation**

**Sensor Nodes (Boards 2, 3, 4)**:

* Read light intensity using **ADC** from the LDR.
* Transmit the data to the central hub using **serial communication**.
* Indicate successful transmission using **GPIO-controlled LEDs**.
* Use **interrupts** for real-time data transmission and **timers** for periodic sampling.

**Central Hub (Board 1)**:

* Collect data from sensor nodes via serial communication.
* Display readings on the LCD using I2C.
* Store configuration settings and thresholds in **EEPROM** for persistent memory.

## ****Core Functionalities****

1. **ADC (Analog to Digital Conversion)**: Converts the analog light sensor output to a digital value (0-1023).
2. **GPIO**: Controls LEDs to indicate data transmission success.
3. **Interrupts**: Ensures immediate response to new readings for efficient data transmission.
4. **Timers**: Sets intervals for periodic light sampling.
5. **Serial Communication**: Transmits data from sensor nodes to the central hub.
6. **EEPROM**: Saves user-defined thresholds for light intensity and other settings.

## ****Results****

The system was tested in various lighting conditions, and the following results were observed:

1. Accurate light intensity readings were transmitted from all sensor nodes to the central hub.
2. The LCD at the central hub displayed real-time data successfully.
3. Switch debouncing effectively eliminated false triggers when toggling settings.

### ****Applications****

1. **Greenhouses**: Optimizes sunlight exposure for plant growth.
2. **Buildings**: Saves energy by adjusting artificial lighting based on ambient light.
3. **Urban Planning**: Measures light pollution or street lighting needs.

## ****Conclusion****

The Distributed Light Intensity Monitoring System effectively demonstrated the ability to monitor light levels across multiple locations and transmit the data to a central hub for analysis and display. The project successfully integrated multiple Arduino functionalities, including ADC, GPIO, interrupts, timers, EEPROM, and serial communication.