



## Project Report

Submitted to,

✚ Md. Repon Islam,  
Lecturer,  
Department of CSE,  
Khulna University Of Engineering &  
Technology

✚ Md. Badiuzzaman Shuvo,  
Lecturer,  
Department of CSE,  
Khulna University Of Engineering &  
Technology

## Night Security Light System

A SIMPLE PROJECT TO DETECT MOTION AT NIGHT

CSE 3104 | Peripherals and Interfacing Laboratory | Date: 08 June 2023

Submitted By-

Biduyt Das

Doniel Tripura

Roll: 1907118

Roll: 1907121

## Objectives:

The objective of a night security light with Arduino can enhance security and visibility during nighttime by automatically turning on a light when it detects movement in low light. This can serve as a restrictive to potential trespassers. The specific objectives for a night security light with Arduino includes:

1. Motion Detection: to detect any movement or presence within the monitored area. If motion is detected, the light should remain on for a configurable duration.
2. Action: to trigger an action over the detection of movement in the preferred light condition.
3. User Interface: Provide a user interface, such as a keypad or buttons, to allow control over the situation when to detect or not.

## Introduction:

A night security light with Arduino can enhance security and visibility during nighttime by automatically turning on a light when it detects low light conditions. This can serve as a restrictive to potential trespassers and provide lighting for improved surveillance.

The system includes a motion sensor (PIR - Passive Infrared Sensor) to detect any movement or presence within the monitored area. If motion is detected, the light should remain on for a configurable duration. Here we used cardboard to limit the range PIR will monitor.

Here's the main features of this project:

- the lamp turns on when it's dark AND movement is detected;
- when movement is detected the lamp stays on for 5 seconds;
- when the lamp is ON and detects movement, it starts counting 5 seconds again;
- when there's light, the lamp is turned off, even when motion is detected.

## Apparatus Required:

- ❖ 1 Arduino UNO
- ❖ 1 PIR Motion Sensor
- ❖ 1 LDR Module
- ❖ 1 Relay module
- ❖ 1 LED Bulb(5 watt)
- ❖ 1 Breadboard
- ❖ 1 Bluetooth Module
- ❖ Jumper wires
- ❖ 1 AC male socket
- ❖ 1 AC wire
- ❖ 1 bulb holder

## Project Hardware design and principle of components:

### 1. **Arduino Uno:**

The Arduino Uno is a microcontroller board that serves as the brain of the project. It receives input from various sensors, processes the data, and controls the output devices accordingly. The Arduino Uno runs the code that defines the logic and behaviour of the night security light.



*Figure 1: arduino uno*

### 2. **PIR Sensor (Passive Infrared Sensor):**

The PIR sensor detects infrared radiation emitted by objects in its field of view. It has two pyroelectric sensors that detect changes in the infrared energy patterns. When a warm object moves within the sensor's range, it triggers a signal indicating motion.



*Figure 2: pir sensor*

3. **Bluetooth Module:**

The Bluetooth module allows wireless communication between the Arduino Uno and a smartphone. It enables you to remotely control the security light and adjust its settings using our “Night Light”.



*Figure 3: bluetooth module*

4. **LDR Module (Light Dependent Resistor):**

The LDR module consists of a light-dependent resistor that changes its resistance based on the ambient light level. It allows the Arduino to measure the intensity of light in the surroundings. This information is used to determine whether the security light needs to be activated or not based on the ambient light conditions.



*Figure 4: ldr module*

5. **LED AC Bulb:**

The LED AC bulb is the primary output device that provides illumination when the security light is triggered. It is an energy-efficient light source that can be controlled by the Arduino through a relay module.



*Figure 5: led ac bulb*

6. **Relay Module:**

The relay module is used to control the LED AC bulb using the Arduino. It acts as a switch that can handle the high voltage and current requirements of the AC bulb. When the Arduino sends a signal to the relay module, it activates the relay, allowing the power to flow to the LED AC bulb and turn it on.



*Figure 6: relay module*

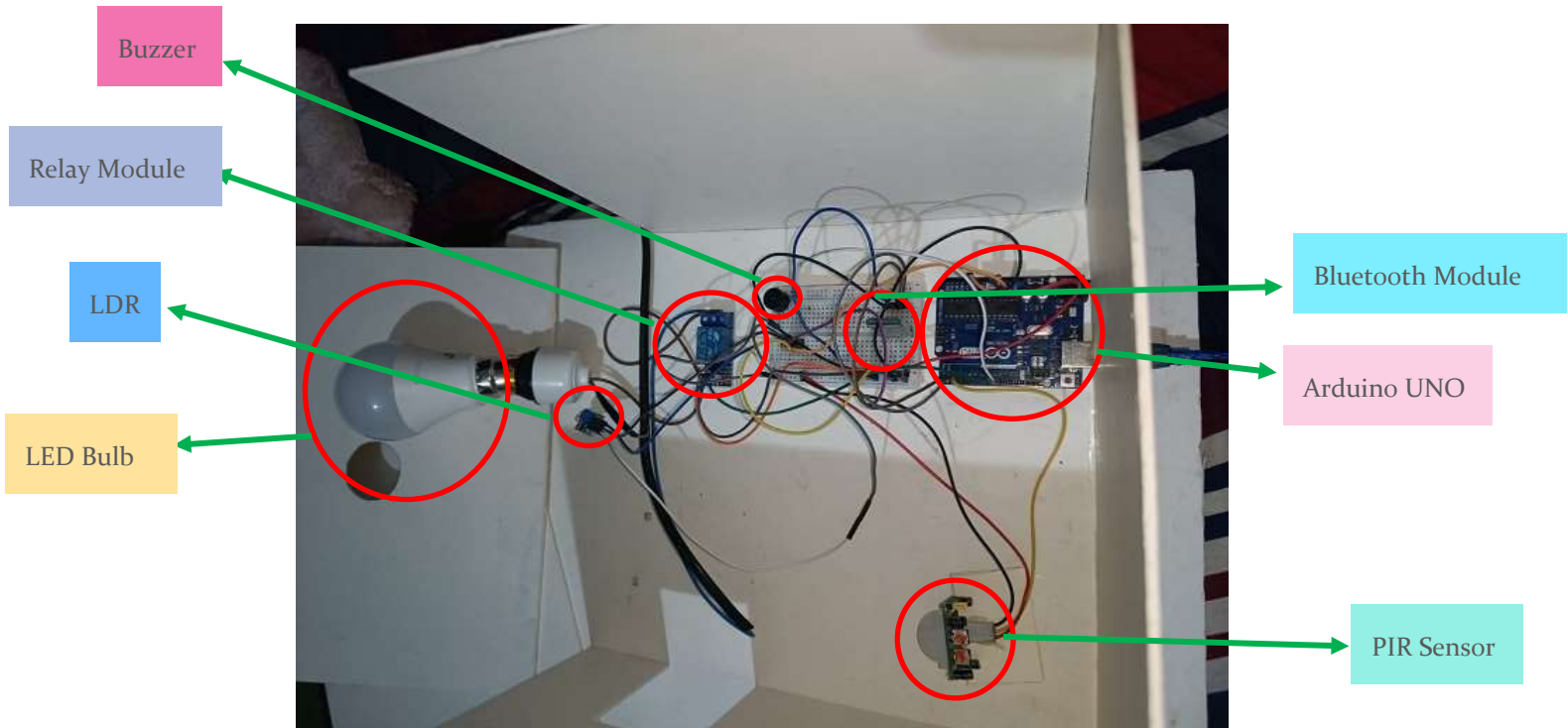
7. **Buzzer:**

The buzzer is an audio output device that emits sound when certain conditions are met. In the night security light project, the buzzer is used to provide audible notifications or alarms when motion is detected or specific events occur.



*Figure 7: buzzer*

By combining these components and their principles of operation, we created a night security light system that detects motion using the PIR sensor, adjusts lighting based on ambient light using the LDR module, communicates wirelessly using the Bluetooth module, and controls the output devices (LED AC bulb and buzzer) through the Arduino using the relay module.

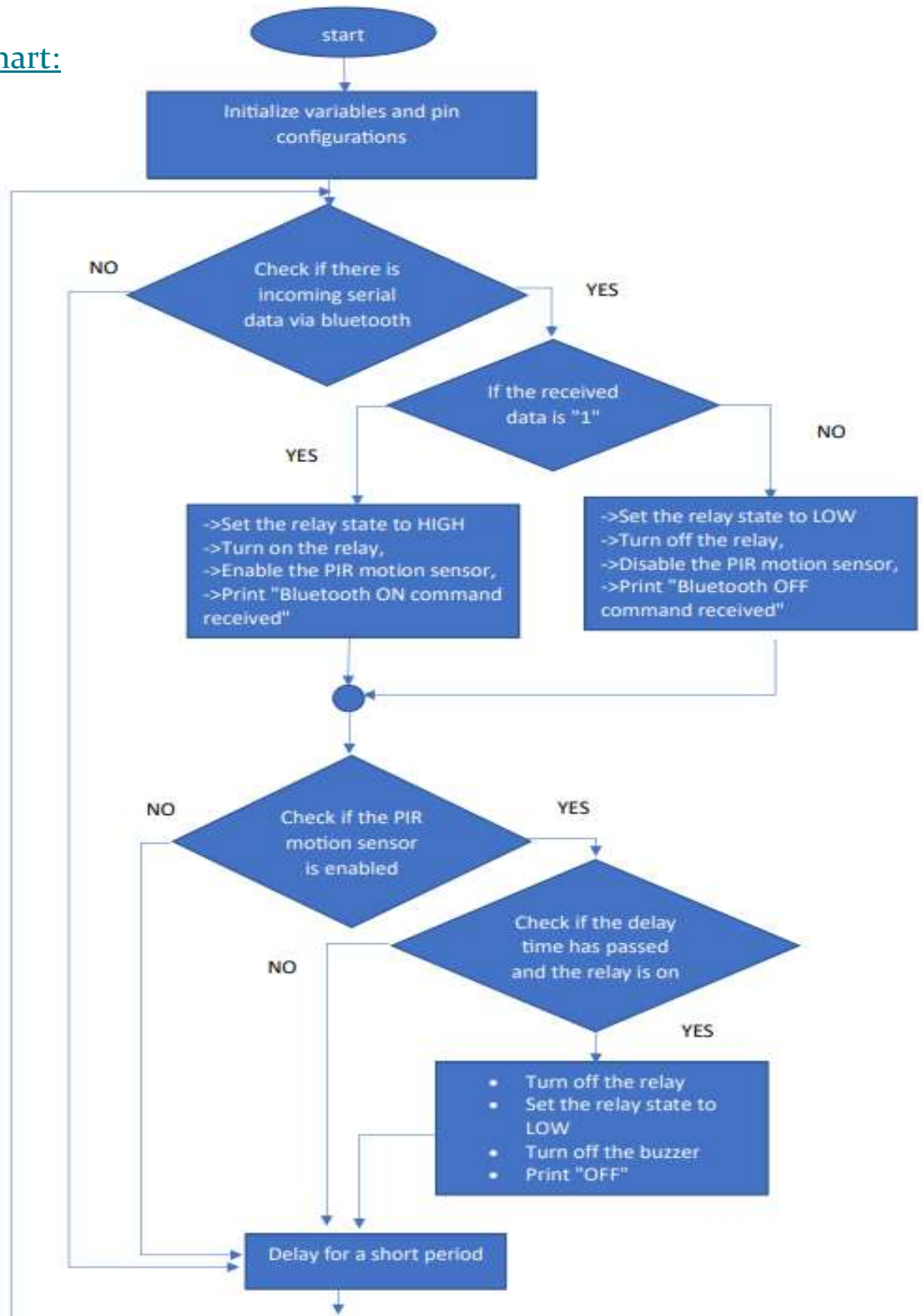


### **NIGHT SECURITY LIGHT**



Fig: Front View, Top view, side view

## Flow chart:



## Software Design(Pseudocode):

```
// Define pin numbers
relayPin = 8 // Relay pin is connected to digital pin 8
pirPin = 2 // PIR motion sensor pin is connected to digital pin 2
ldrPin = A0 // LDR pin is connected to analog pin A0
buzzerPin = 9 // Buzzer pin is connected to digital pin 9

// Initialize variables
relayState = LOW // Relay state is initially LOW (off)
pirEnabled = true // PIR motion sensor is initially enabled
ldrThreshold = 300 // Threshold value to compare with LDR reading
lastMotionTime = 0 // Stores the time of the last detected motion
motionTimeout = 5000 // Motion timeout duration in milliseconds (5 seconds)

setup():
    // Set pin modes
    pinMode(relayPin, OUTPUT) // Set relay pin as output
    pinMode(pirPin, INPUT) // Set PIR motion sensor pin as input
    pinMode(buzzerPin, OUTPUT) // Set buzzer pin as output
    attachInterrupt(digitalPinToInterrupt(pirPin), detectMotion, RISING) // Attach
interrupt to PIR motion sensor pin
    Serial.begin(9600) // Initialize serial communication

loop():
    checkBluetoothCommands() // Check for incoming Bluetooth commands

    if (pirEnabled):
        checkMotionTimeout() // Check if motion timeout has occurred

    delay(50) // Delay to avoid excessive loop iterations

checkBluetoothCommands():
    if (Serial.available()): // Check if there is incoming serial data
        receivedData = Serial.readString() // Read the received data
        receivedData.trim() // Remove leading/trailing whitespaces
        if (receivedData == "1"):
            turnOnLights() // Turn on the lights
            Serial.println("Bluetooth ON command received")
        else if (receivedData == "0"):
            turnOffLights() // Turn off the lights
            Serial.println("Bluetooth OFF command received")
```



```

checkMotionTimeout():
    if ((millis() - lastMotionTime) >= motionTimeout and relayState == HIGH):
        turnOffLights() // Turn off the lights if motion timeout has occurred

turnOnLights():
    if (relayState == LOW): // Check if the lights are currently off
        digitalWrite(relayPin, LOW) // Turn on the relay to activate the lights
        digitalWrite(buzzerPin, HIGH) // Turn on the buzzer if required
        relayState = HIGH // Update the relay state to indicate lights are on
        lastMotionTime = millis() // Update the last motion time
        Serial.println("Lights ON")

turnOffLights():
    digitalWrite(relayPin, HIGH) // Turn off the relay to deactivate the lights
    digitalWrite(buzzerPin, LOW) // Turn off the buzzer
    relayState = LOW // Update the relay state to indicate lights are off
    Serial.println("Lights OFF")

detectMotion():
    if (pirEnabled): // Check if the PIR motion sensor is enabled
        Serial.println("Motion")
        ldrReading = analogRead(ldrPin) // Read the LDR value
        if (ldrReading > ldrThreshold): // Compare LDR reading with threshold
            turnOnLights() // Turn on the lights if threshold is exceeded

```

## Project Execution:

The video of the project is [Here](#)





## Discussion:

The night security light project combines various components and techniques to create an effective and versatile security lighting system. The project utilizes the principles of motion detections, ambient light sensing, wireless communication, and a microcontroller-based control to enhance nighttime security and provide convenient control options. The inclusion of PIR sensor enables the system to detect motion within range. By incorporating an LDR module, the system can measure the ambient light level in its surroundings. The integration of Bluetooth module enables users to remotely control the system. The Arduino serves our projects as central processing unit. It receives input from LDR module and PIR sensor, process the data, and controls the output data according to out input. The project utilizes an LED AC bulb and a buzzer as the primary output devices.

## Conclusion:

The night security light project combines motion detection, ambient light sensing, wireless communication, and a microcontroller-based control to create an effective and a versatile security lighting system. By incorporating these features, the project enhances nighttime security and provides convenient control options to users. The flexibility and customization options offered by the Arduino platform and the wireless communication capabilities add value and adaptability to the system. Overall, the project provides an efficient and user-friendly solution for nighttime security applications.

## Applications:

1. Home security
2. Outdoor security
3. Pathway Illumination
4. Energy conservation
5. Smart home integration
6. Public safety
7. Emergency lighting

## Bottleneck:

1. manual security system instead of automatic security system.
2. The delay function pauses the program for a specified amount of time, which can cause the program to become unresponsive during that delay.
3. Range is not that vast, that's why usable only for short range.

## Future Update:

1. automatic security system instead of manual security system.
2. Multiple pir sensor support.
3. Integration with home automation system.
4. Geofencing and GPS Integration.
5. Advanced motion detection Algorithm.

## References:

- 1.[Random Tutorials](#)
- 2.[Home-Arduino UNO](#)