

Uganda Christian University

Department of Computing

AUTOMATED CLASSROOM SYSTEM

|  |  |
| --- | --- |
| Student Name: | Magezi Richard Elijah |
| Course: BSIT |  |
| Instructor(s): | Mr. Elijah, Mr. Bashir |
| Date: | 19th March 2025 |

**Table of Contents**

[Chapter 1. Introduction 1](#_Toc193728490)

[**1.1** **Background:** 1](#_Toc193728491)

[**1.2** **Problem Statement:** 1](#_Toc193728492)

[**1.3** **Objectives:** 1](#_Toc193728493)

[Chapter 2. Proposed Solution 2](#_Toc193728494)

[**2.1** **Overview** 2](#_Toc193728495)

[**2.2** **Key Components and Technologies** 2](#_Toc193728496)

[Chapter 3. Design Process 3](#_Toc193728497)

[**3.1** **Concept Development** 3](#_Toc193728498)

[**3.2** **Component Selection** 3](#_Toc193728499)

[**3.3** **Programming** 4](#_Toc193728500)

[Chapter 4. Implementation and Testing 8](#_Toc193728501)

[**4.1** **Assembly Process** 8](#_Toc193728502)

[**4.2** **Testing Methods and Results** 8](#_Toc193728503)

[Chapter 5. Final Project Outcome 9](#_Toc193728504)

[**5.1** **Final Working Project** 9](#_Toc193728505)

[**5.2** **Images of the Completed Project** 9](#_Toc193728506)

[Chapter 6. Discussion and Analysis 10](#_Toc193728507)

[**6.1** **Evaluation** 10](#_Toc193728508)

[**6.2** **Potential Improvements** 10](#_Toc193728509)

[**6.3** **Limitations** 10](#_Toc193728510)

[Chapter 7. Conclusion 11](#_Toc193728511)

[**7.1** **Summary of Outcomes** 11](#_Toc193728512)

[**7.2** **Learning Experience** 11](#_Toc193728513)

# Introduction

## **Background:**

Modern classrooms have a manual management of controls for lighting, air circulation and equipment like projectors. While this has worked for the longest time, it leads to inefficiencies, equipment being left on and unnecessary energy consumption. As educational institutions continue to stress sustainability and forward technology, there is a growing need to automate management of classroom resources. An automated system which uses motion sensors to control lights fans, projectors and their screens can optimize energy usage, enhance classroom comfort and simplify operations therefore improving convenience.

## **Problem Statement:**

In a regular classroom setup, things like lights, fans and projectors are manually controlled leading to issues like the following;

1. Lights, fans and projectors are often left on both when classrooms are occupied and unoccupied, leading to **unnecessary energy consumption** and higher electricity costs.
2. Constant use of classroom equipment and devices without regard for usage patterns may lead to **premature wear and tear** thus increasing maintenance costs.
3. Inadequate control over classroom devices can affect the overall learning environment, such as improper lighting, insufficient ventilation, or difficulty accessing multimedia resources at the right time. This creates **inconvenience** in the classroom.

## **Objectives:**

1. Energy Efficiency: Automation reduces energy consumption thus lowering electricity costs.
2. Convenience: Removal manual control improves the overall classroom experience.

# Proposed Solution

## **Overview**

The Automated Classroom System is designed to automate and optimize the management of classroom devices such as lights, fans and projectors through the use of motion sensor to automatically control their operation based on occupancy. The system detects movement in the classroom to turn on devices and turns them off when a button is pressed and depending on which one is pressed. This increases energy efficiency and reduces operational and repair costs. Options for manual control and monitoring improve convenience, reduce energy waste and enhances sustainability within educational spaces.

## **Key Components and Technologies**

1. **Arduino Uno Microcontroller:** This will be used to process inputs from the motion sensor and send output signals to the connected devices (lights, fans, and projectors). This was used since it would be the simplest to work with and has a manageable power consumption for the size of the project.
2. **Passive Infrared Sensors:** These sensors detect infrared radiation from human bodies to determine whether the classroom is occupied or not. The sensors are low-cost, reliable, and easy to integrate into an automated system. PIR sensors where used since the other types of sensors such as the ultrasonic sensors refused to work with the system.
3. **Lights, Fan, and Projectors**: Controlled automatically based on motion detection. LEDs where used to represent the projectors and lights and a motor was used represent the fan.

# Design Process

## **Concept Development**

There was a brainstorming session in which multiply ideas both good and bad where thought out discussed. The ideas where heavily focused on implementing the principles of the internet of things and energy efficiency. The initial idea was to create an automated light system but a conclusion was drawn that it would to general.

The idea selection was based of the criteria of feasibility, time limit of the project development, the target group and simplicity leading to the final idea being the automated classroom system.

## **Component Selection**

|  |  |  |
| --- | --- | --- |
| **Component** | **Number of pieces** | **Purpose** |
| Arduino Uno | 1 | Acts as a microcontroller and is where code to manage the system is uploaded. |
| PIR motion sensor | 1 | Detects motion and sends signals to the Arduino to control the system |
| LED | 2 | One represents the lighting and another the projector. |
| Button | 2 | One button turns of the motor for the fan and LED for lighting and the other the LED representing the projector. |
| Transistor | 2 | These control the flow of electrical current between the Dc motors and the Arduino board. |
| Resistor | 5 | These also regulate current but are mostly used on LEDs and buttons. |
| Bread board | 2 | This acts as the circuit board to which all the other components are added. |
| DC motors | 2 | One DC motor represents the fan and the other acts as the motor to release and roll back the projector screen |
| Jumper Wires | 30 | To carry signals to and from the Arduino board, Ground components and provide the 5V power. |

## **Programming**

The code controls LEDs and motors based on input from a PIR motion sensor and two buttons. When motion is detected, LED 1 turns on, LED 2 turns off, and two motors are activated, with one moving forward and the other backward. A button press turns off the LEDs, and another button stops the motors. The system includes a timer to turn off one motor after 2 seconds and ensures LEDs stay on if no motion is detected. The motors continue running unless the stop button is pressed, halting motor movement.

The key algorithms and logic work like this. The code continuously checks the PIR sensor for motion, toggles LEDs and motor states based on button presses, and controls motor movement with a timer. When motion is detected, LEDs are toggled, and motors are activated in specific directions. If the stop button is pressed, motors stop, and the LEDs can be turned off with another button press. If no motion is detected, LEDs stay on for 40 seconds, and motors continue running unless stopped manually.

Defining pins for different components

// Define motor control pins

#define motor1Pin1  7 // IN1

#define motor1Pin2  6 // IN2

// Define LED pins

const int led1Pin = 12; // LED 1 (Use a different pin for the LED)

const int led2Pin = 13; // LED 2 (Use a different pin for the LED)

// Define PIR sensor pin

const int pirPin = 8; // PIR sensor output pin

// Define button pins

const int buttonPin = 9; // Button to manually turn off LEDs

const int stopButtonPin = 10; // Button to stop the motors

Setup for different pins

void setup() {

  // Start the Serial communication

  Serial.begin(9600);

  // Set motor control pins as outputs

  pinMode(motor1Pin1, OUTPUT);

  pinMode(motor1Pin2, OUTPUT);

  // Set LED pins as outputs

  pinMode(led1Pin, OUTPUT);

  pinMode(led2Pin, OUTPUT);

  // Set PIR sensor pin as input

  pinMode(pirPin, INPUT);

  // Set button pins as input

  pinMode(buttonPin, INPUT);

  pinMode(stopButtonPin, INPUT);

}

Void loop for decision making for the PIR Sensor, motors, LEDs and buttons

void loop() {

  // Read the PIR sensor

  int pirState = digitalRead(pirPin);

  // Check if the button is pressed to turn off LEDs

  buttonState = digitalRead(buttonPin);

  // Check if the stop button is pressed to stop the motors

  stopButtonState = digitalRead(stopButtonPin);

  // If the stop button is pressed, stop the motors

  if (stopButtonState == HIGH && motorsRunning) {

    // Stop the motors

    digitalWrite(motor1Pin1, HIGH);

    digitalWrite(motor1Pin2, HIGH);

    motorsRunning = false;

    Serial.println("Stop button pressed: Motors ON");

    delay(2000);

  }

  // If the button is pressed, turn off the LEDs

  if (buttonState == HIGH && ledsOn) {

    digitalWrite(led1Pin, LOW);

    digitalWrite(led2Pin, LOW);

    ledsOn = false;  // Update LED status to off

    Serial.println("Button pressed: LEDs OFF");

  }

  // If motion is detected and LEDs are not already on

  if (pirState == HIGH && !ledsOn) {

    Serial.println("Motion detected!");

    // Turn on LED 1

    digitalWrite(led1Pin, HIGH);

    Serial.println("LED 1 ON");

    // Turn off LED 2

    digitalWrite(led2Pin, HIGH);

    Serial.println("LED 2 OFF");

    // Start moving Motor 1 forward

    digitalWrite(motor1Pin1, HIGH);

    digitalWrite(motor1Pin2, LOW);

    Serial.println("Motor 1 moving forward");

    // Start moving Motor 2 backward

    digitalWrite(motor1Pin2, HIGH);

    digitalWrite(motor1Pin2, HIGH);

    Serial.println("Motor 2 moving backward");

    motorsRunning = true; // Motors are now running

    ledsOn = true; // LEDs are on

    motor1Pin2Timer = millis();  // Start timer for motor 2

    motor1Pin2On = true;

  }

  // If 2 seconds passed after motion detection, turn off motor 2

  if (motor1Pin2On && millis() - motor1Pin2Timer >= 2000) {

    digitalWrite(motor2Pin2, LOW);

    motor1Pin2On = false;

  }

  // If no motion detected, keep LEDs on if they've been turned on

  if (pirState == LOW && ledsOn) {

    Serial.println("No motion detected, but LEDs are still ON.");

    delay(40000);

  }

  // Motors keep running if the stop button is not pressed

  if (motorsRunning) {

    digitalWrite(motor1Pin1, HIGH);

    digitalWrite(motor1Pin2, LOW);

    Serial.println("Motors are running.");

  }

}

# Implementation and Testing

## **Assembly Process**

The first step was to gather all the necessary components and do further research on what other components where necessary and how they can best be inserted into the system. The components gathered included bread boards, resistors, LEDs, DC motors, Transistors, Jumper Wires and a few others.

The next step was setting up the components. Here a simulation was first drawn up in the Tinker Cad simulation software to make sure that the gathered components where right for the job and the system would have less failures and blow outs after this the system was set up according to the simulation.

The 5V port and grounding port where connected to the two topmost horizontal lines to create a 5V and grounding line.

Instead using a motor driver, transistors where used on the DC motors since their functions where no going to be that complicated. The motors where connected to the collector pins on the transistors, the base pin was grounded and the emitter pin was connected to a port on the motor. LEDs where used to represent the lighting and the projector, resistors where connected to these LEDs to manage the current. The PIR sensor was grounded, connected to the 5V line on the Breadboard and connected to a port on the Arduino board. Components like LEDs and buttons where grounded and then connected to Arduino ports to manage their functionality.

This whole system was then placed in a box that would simulate a classroom.

## **Testing Methods and Results**

The components where set up and tested in phases of first the LEDs, then the motors and lastly the PIR sensor this was mostly done to test for functionality and weed out faulty parts. Different example codes where used to test the different phases.

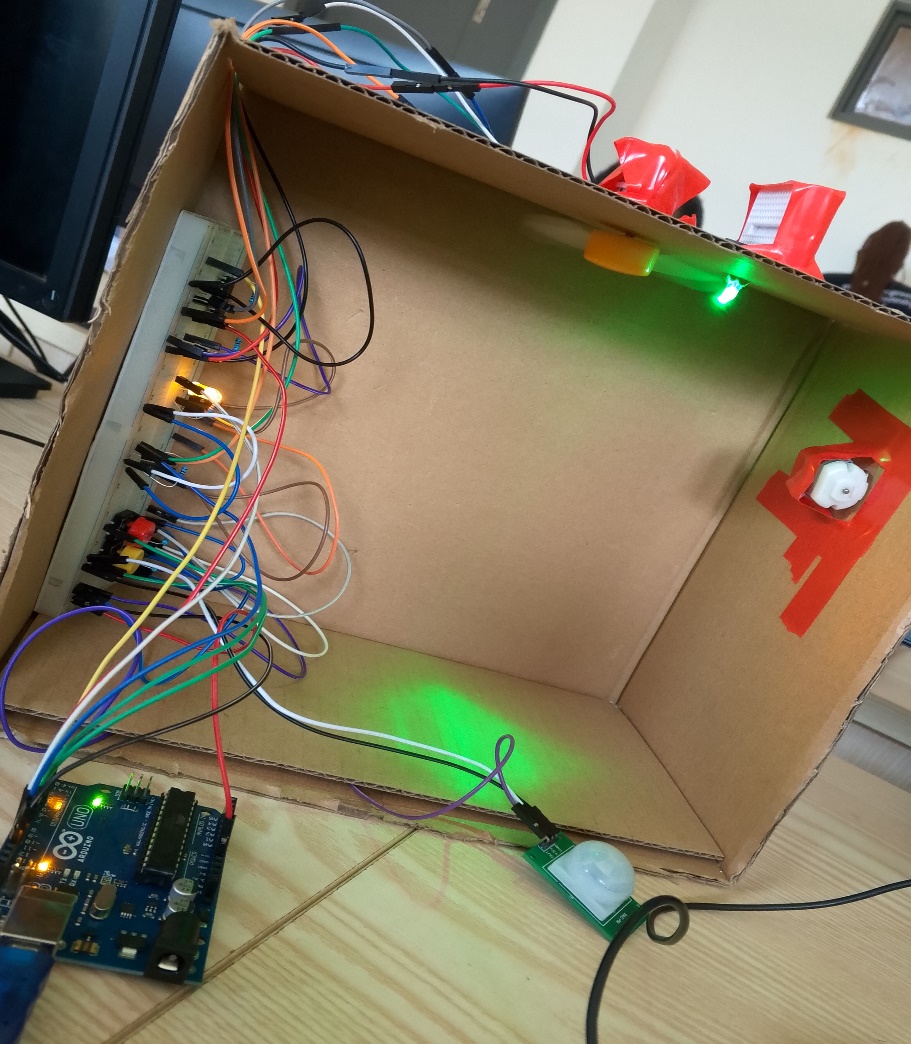
A multimeter was use determine the presence of and amount of current going through the system. This helped to decide if more resistors and power regulating components where to be used.

# Final Project Outcome

## **Final Working Project**

The Automated classroom system works when the PIR Sensor first detects motion a signal is sent to the Arduino. This signal is what the Arduino uses to turn on the motor representing the fan, the LED representing the lighting, the LED representing the project and the motor to push down the projector screen. These components will run until one button is pressed which will turn off the lighting LED and the motor representing the fan. Pressing the second button turns off the LED representing the projector. This is how the project will work.

## **Images of the Completed Project**



# Discussion and Analysis

## **Evaluation**

Overall the project from start to finish was a good challenge but on the final testing day it was a success.

## **Potential Improvements**

Use of an Arduino mega to provide more ports.

Addition of an LCD to show the states of different components.

Changing the LED for the lighting to actual bulbs.

Reducing the systems power consumption.

## **Limitations**

Lack of jumper wires and sometimes the available ones would be to short for their assigned purposes.

Limited understanding and time to create a more efficient system.

# Conclusion

## **Summary of Outcomes**

A deeper understanding of Arduino and embedded systems has been gained.

A world of possibilities on the project’s application has been opened up increasing the range in which I can apply the knowledge of embedded systems.

## **Learning Experience**

The experience has been quite challenging but enlightening as a deeper more profound understanding of how computers how they integrate into the world around has been developed.

This project has helped me understand how projects can be applied to real world situations and has shown the importance of embedded systems to all computing and engineering students.