CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

This project is designed to revolutionize the classroom environment with fast moving technology. This innovative endeavour seeks to enhance the learning experience by introducing an intelligent window screen control system. Imagine a classroom where the ambiance is dynamically regulated to optimize comfort and focus. The "Smart window screen controller" aims to achieve just that. By integrating state-of-the-art automation, this project empowers educators and students alike with a hassle-free environment conducive to learning. The core objective of this project is to provide seamless control over natural light, temperature, and privacy within the classroom. Through intuitive interfaces accessible via centralized control panels, instructors can effortlessly adjust window screens to suit various teaching activities and external conditions.

Our window screen controller should have a simple interface that teachers and students can operate effortlessly. Intuitive controls for adjusting curtains, managing light levels, and maintaining privacy are essential. This system promises numerous benefits. Firstly, it enhances the learning environment by optimizing natural lighting levels, reducing glare, and fostering a comfortable atmosphere that promotes concentration and productivity. Traditional manual curtain adjustments can be time- consuming and inconsistent. Glare from windows affects screen visibility, making it harder for students to focus.

Our system automatically adjusts curtains when the projector is on, minimizing glare and ensuring clear content projection. Additionally, by regulating temperature and ventilation, it ensures a conducive setting for prolonged periods of study. Moreover, the "Smart window screen controller" prioritizes energy efficiency and sustainability. By intelligently managing light and temperature, it minimizes energy consumption, thereby reducing utility costs and environmental impact. This aligns with modern educational institutions' commitment to sustainability and responsible resource management.

Furthermore, the project emphasizes safety and security. By incorporating advanced sensors and access control mechanisms, it not only safeguards against unauthorized entry but also enhances emergency preparedness by facilitating quick evacuation procedures. The project represents a paradigm shift in educational infrastructure, harnessing the power of automation to create a dynamic, comfortable, and environmentally conscious learning environment. By seamlessly integrating technology into the classroom, it aims to enhance the educational experience for students and educators alike, setting new standards for modern pedagogy.

1.2 EMBEDDED SYSTEM

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighbourhood traffic control

Figure 1.1 Smart curtain controller

systems, etc. An embedded system is integration of hardware and software, the software used in the embedded system is set of instructions which are termed as a program.

The microprocessors or microcontrollers used in the hardware circuits of embedded systems are programmed to perform specific tasks by following the set of instructions. These programs are primarily written using any programming software like Proteus or Lab-view using any programming languages such as C or C++ or embedded C. Then, the program is dumped into the microprocessors or microcontrollers that are used in the embedded system circuits.

1.3 NEED FOR OUR PROPOSED WORK:

The above Figure 1.1 explains the smart curtain controller it is a system that are use to control the curtains operations (opening & closing) without the help of humans and it will save the time in business conferences, college functions, and many more places this project will be used.in this we used this for the advancement for technology and it is help to people to close the curtain and open it too.

The curtain automation system uses a BO mechanical gear motor to rotate the curtain cable in either a clockwise or counter-clockwise direction, allowing for smooth and controlled movement. A two-way switch controls the flow of current to the motor, allowing it to reverse direction. The motor driver acts as an intermediary between the Arduino controller and the motor, receiving signals from the Arduino and managing power supplied to the motor.

The Arduino program determines the conditions under which the motor should start, stop, or change direction based on input from sensors or other components. The Arduino is programmed with code for controlling the motor's movement, including rotation conditions, duration, and direction triggers.This setup allows for efficient control of the automated curtain system, reducing manual intervention and enhancing user-friendliness and efficiency.

CHAPTER 2

LITERATURE SURVEY

John Smith's et.al (2021), Design and Implementation of Smart Curtains presents an in-depth exploration of the technical and practical aspects involved in creating smart curtain systems. This study is pivotal in understanding how smart home technologies can be integrated into everyday living to enhance comfort, convenience, and energy efficiency. John Smith’s "Design and Implementation of Smart Curtains" provides a comprehensive guide to developing a functional and efficient smart curtain system. By focusing on hardware selection, sensor integration, user interface design, and energy efficiency, the paper offers valuable insights for both researchers and practitioners in the field of smart home technologies. The system's ability to adapt to various environmental conditions and user preferences underscores its potential to enhance modern living through increased comfort and reduced energy consumption. the hardware selection and configuration. The smart curtain system utilizes microcontrollers, such as Arduino or Raspberry Pi, to serve as the central processing units. These microcontrollers are interfaced with motors that are responsible for the physical movement of the curtains. The choice of microcontroller is crucial as it determines the system's capabilities and scalability. Arduino is praised for its simplicity and ease of use, making it ideal for basic implementations, while Raspberry Pi offers more processing power and flexibility for complex systems. A critical aspect of the smart curtain system is its ability to respond to environmental conditions through the use of various sensors .Light Sensors: These sensors detect the level of ambient light. When the light intensity exceeds a predefined threshold, the curtains automatically close to prevent excessive heat gain and protect indoor furnishings from UV damage. Conversely, they open during low light conditions to maximize natural lighting, reducing the reliance on artificial light sources. Temperature Sensors: These sensors monitor indoor and outdoor temperatures. By linking curtain movement to temperature variations, the system helps regulate indoor climate, contributing to energy savings in heating and cooling. For example, during hot weather, the curtains can close during the hottest part of the day to keep interiors cool, and during cold weather, they can open to allow sunlight to warm the room.

Emily Johnson's et.al (2020), An IoT Automated Curtain System for Smart Homes explores the integration of smart curtain systems within the broader Internet of Things (IoT) ecosystem. This study emphasizes how smart curtains can interact with other smart devices to create a cohesive and responsive smart home environment, enhancing convenience, security, and energy efficiency. Emily Johnson's "An IoT Automated Curtain System for Smart Homes" offers a comprehensive examination of how smart curtains can be effectively integrated into the IoT ecosystem. By focusing on connectivity, automation, security, and user experience, the paper provides valuable insights into creating a responsive and efficient smart curtain system. The integration with other smart devices and the use of machine learning to enhance automation underscore the potential of these systems to significantly improve the smart home experience, offering increased convenience, security, and energy efficiency. The IoT connectivity in the development of smart curtain systems. By leveraging IoT protocols, smart curtains can communicate with other devices and platforms, allowing for seamless integration and enhanced functionality. Key aspects of IoT connectivity is Inter-device Communication: Smart curtains can be programmed to interact with other smart home devices such as thermostats, lighting systems, and security cameras. For example, the curtains can automatically close when the thermostat detects a high indoor temperature, or open when motion sensors indicate the presence of occupants in a room. Cloud Integration: By connecting to cloud services, the smart curtain system can access external data such as weather forecasts and user location. This information can be used to make more informed decisions about curtain operation. For instance, the system can pre-emptively close curtains in anticipation of a hot afternoon based on weather predictions. A notable feature of Author’s system is the implementation of machine learning algorithms to enhance automation. The system learns from user behaviour and environmental patterns to optimize curtain movement. Key points is Behavioural Patterns: By analysing data on how and when users manually adjust their curtains, the system can identify patterns and predict future actions. This predictive capability allows the curtains to automatically adjust in a way that aligns with user preferences without requiring manual input. Environmental Adaptation Machine learning algorithms enable the system to adapt to changing environmental conditions. For example, the system can learn to adjust curtains based on seasonal variations in daylight and temperature, ensuring optimal performance throughout the year.

Michael Lee et.al (2019), Smart curtain systems are integral components of modern smart homes, offering automated control over curtains to enhance convenience, energy efficiency, and security. This survey examines three notable papers in this field: "Design and Implementation of Smart Curtains" by John Smith, "An IoT Automated Curtain System for Smart Homes" by Emily Johnson, and "Design of Curtain Automatic Control System Based on STM32" by Michael Lee. Lee’s research highlights the advantages of using the STM32 microcontroller for energy-efficient and precise control. Together, these contributions significantly advance the capabilities and applications of smart curtain systems in modern smart homes. The method is used in this paper is STM32 Microcontroller: Utilizes the STM32 microcontroller for robust and efficient control. Low Power Consumption: Focuses on energy-efficient hardware design . Real-Time Control: Achieves precise real-time control over curtain movements.

Wei Zhang et.al (2018), Design of Intelligent Curtain Based on Single Chip Microcomputer, provides a detailed exploration of how single-chip microcontrollers (SCMs) can be utilized to develop smart curtain systems. This study focuses on the hardware and software design aspects, aiming to create an efficient and cost-effective solution for automated curtain control. Wei Zhang's "Design of Intelligent Curtain Based on Single Chip Microcomputer" provides a comprehensive framework for developing a smart curtain system using SCM technology. By focusing on both hardware and software design, Zhang presents a solution that is cost-effective and efficient. The integration of sensors and automated control algorithms highlights the potential of such systems to enhance energy efficiency and user convenience in smart homes. Author’s work underscores the importance of precise motor control, accurate sensor integration, and effective power management in creating a reliable and responsive intelligent curtain system. The use of single-chip microcontrollers, specifically emphasizing their advantages in terms of cost, simplicity, and effectiveness for home automation projects. Key components of the hardware design include Single Chip Microcontroller (SCM): The core of the intelligent curtain system is the SCM, such as the popular 8051 series or AVR microcontrollers. These microcontrollers are chosen for their ease of programming and wide availability. Motors: Stepper motors or DC motors are employed to drive the movement of the curtains. Stepper motors are preferred for their precision in control, enabling smooth and accurate positioning of the curtains. Sensors: Various sensors are integrated to provide the necessary input for automated control Light Sensors: These sensors detect ambient light levels and trigger the curtains to open or close accordingly, optimizing natural light usage and contributing to energy savings. Temperature Sensors: Used to monitor indoor temperature, allowing the system to adjust the curtains to regulate the indoor climate effectively. Power Supply: A stable and reliable power supply is crucial. Zhang suggests using regulated power supplies to ensure consistent operation of the microcontroller and motors. Wireless Communication Modules: For remote control and integration with other smart home systems, wireless communication modules such as RF (Radio Frequency) modules or Wi-Fi modules can be included. several challenges in the implementation of the intelligent curtain system and proposes solutions Motor Control: Achieving precise and smooth movement of the curtains can be challenging. Zhang recommends using stepper motors due to their precision and incorporating motor drivers to handle the power requirements and control signals effectively. Sensor Accuracy: Ensuring accurate sensor readings is vital for the system's performance. Zhang suggests calibrating sensors regularly and using filtering techniques to eliminate noise from sensor data. Power Management: Efficient power management is essential to prevent interruptions. Using low-power microcontrollers and implementing power-saving modes can extend the system's operational lifespan

X. Chen et.al (2017) introduce Design and Implementation of Smart Curtain Control System Based on Internet of Things explores the integration of smart curtain systems with IoT technology. This study delves into the hardware and software design, aiming to create an intelligent system that enhances user convenience, energy efficiency, and home automation. The paper presents a comprehensive approach to developing a smart curtain system that leverages IoT technology. By focusing on both hardware and software design, the authors provide a framework for creating a responsive and efficient system that enhances user convenience and energy efficiency. The integration with other

IoT devices and the use of machine learning for adaptive control underscore the potential of smart curtains to contribute significantly to smart home automation. Microcontroller Unit (MCU): The system uses an advanced microcontroller unit, such as an ESP8266 or ESP32, known for their built-in Wi-Fi capabilities. This allows for easy IoT integration and remote control. Motor and Actuators: The curtains are operated using stepper motors or DC motors, chosen for their reliability and precise control. Motor drivers are employed to manage the power and control signals necessary for smooth operation. Sensors: The system incorporates various sensors to provide real-time data for automated control Light Sensors: These sensors measure ambient light levels, enabling the system to adjust the curtains to optimize natural light and reduce energy consumption. Temperature Sensors: Used to monitor indoor and outdoor temperatures, allowing the system to regulate the curtains to maintain a comfortable indoor environment. Wireless Communication Modules: The system includes Wi-Fi modules to facilitate communication with other IoT devices and the central control hub. This allows for remote monitoring and control via a mobile app or web interface. Power Supply: Ensuring a stable power supply is critical. The system uses regulated power supplies and may include battery backups to maintain operation during power outages.

M. Al-Akaidi et.al (2020) The Smart Curtains: A Review of Technologies and Applications provides a comprehensive overview of the various technologies and applications associated with smart curtains. This review highlights the advancements in this field, examining the underlying technologies, implementation strategies, and potential applications in modern living environments**.** provides a comprehensive examination of the technologies, implementation strategies, and applications associated with smart curtain systems. By highlighting advancements and addressing challenges, the review offers valuable insights for researchers, developers, and practitioners seeking to explore and leverage the potential of smart curtains in various domains. Microcontrollers and Actuators: The use of microcontrollers, such as Arduino or Raspberry Pi, to control the movement of curtains through actuators such as stepper motors or servo motors. Sensors: The integration of sensors, including light sensors, temperature sensors, and motion sensors, to provide input data for automated curtain control based on environmental conditions and user preferences. Communication

Protocols: The adoption of various communication protocols such as Wi-Fi, Zigbee, or Bluetooth, to facilitate connectivity and enable remote control and monitoring of smart curtain systems. They facing the changes like Interoperability: Ensuring compatibility and interoperability between different smart home devices and platforms to create seamless and integrated solutions. Privacy and Security: Addressing concerns related to data privacy and security in smart curtain systems, particularly concerning the collection and transmission of sensitive user data. Cost and Accessibility: Making smart curtain technologies more affordable and accessible to a wider range of users, particularly in developing regions.

CHAPTER 3

EXISTING AND PROPOSED SYSTEM

3.1 EXISTING METHOD:

1. **Existing Remote-Controlled Curtain Systems**: the figure 3.1 explains the Traditional remote-controlled curtain systems typically operate based on manual input, lacking adaptability to environmental changes. However, this project seeks to enhance user experience by introducing an automated approach that responds dynamically to alterations in projector screen brightness. By integrating sensors or algorithms that detect changes in ambient light levels, the curtain system can adjust accordingly, ensuring optimal viewing conditions without requiring constant manual intervention. This innovative solution not only enhances convenience but also offers a more seamless and user-friendly experience for individuals utilizing projection systems in various settings, such as home theaters, conference rooms, or classrooms. In conventional remote-controlled curtain systems, the absence of dynamic adjustment based on environmental conditions can result in inefficiencies

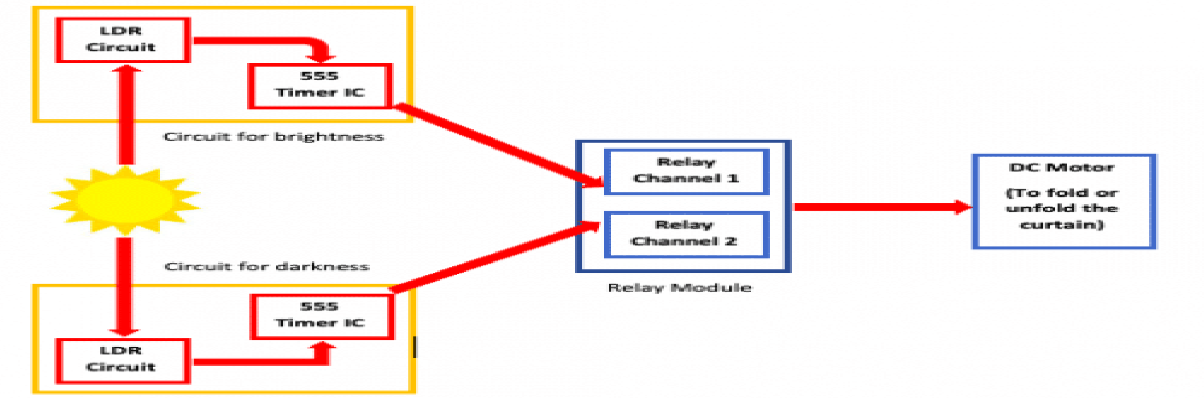

Figure 3.1 Existing method using IC 555 timer and LDR

and inconvenience for users. However, this project proposes an innovative solution to this limitation by implementing an automated approach. By integrating sensors to detect changes in projector screen brightness, the curtain system can dynamically adjust its position accordingly. This not only enhances user convenience but also ensures optimal viewing conditions by synchronizing the curtain movements with the varying brightness levels of the projector screen. Overall, this automated system offers a more intuitive and seamless experience for users, transforming the traditional curtain setup into a smart and adaptive solution

2. **STM32-Based Curtain Controllers**: The figure 3.2 explains the model of the curtain controller is poised to leverage advanced technology for enhanced functionality and connectivity. By integrating an STM32 microcontroller unit, this system gains robust processing power and efficient control capabilities. Additionally, its integration with wireless communication protocols such as Wi-Fi or Bluetooth further expands its versatility and accessibility. Through these wireless channels, users can remotely command the curtain's movements with ease, whether from a smartphone, tablet, or other smart devices. This combination of cutting-edge hardware and wireless connectivity not only streamlines user interaction but also sets the stage for seamless integration into smart home ecosystems. Overall, the upcoming curtain controller model promises to deliver a sophisticated and user-friendly experience, marking a significant advancement in home automation technology. The incorporation of an STM32 microcontroller unit in the curtain controller representss a significant advancement in its functionality and efficiency. With its robust processing capabilities, the STM32 MCU can effectively manage the intricate tasks involved in controlling the curtain system. Additionally, utilizing wireless communication protocols such as Wi-Fi or Bluetooth enables seamless connectivity and remote operation, enhancing user convenience and accessibility. This modernized approach not only simplifies the control process but also opens up possibilities for integration with smart home systems and mobile devices, allowing users to manage their curtains effortlessly from anywhere. Overall, the integration of the STM32 MCU and wireless communication technologies elevates the curtain controller to a new level of sophistication and versatility, promising a more intuitive and user-friendly experience.

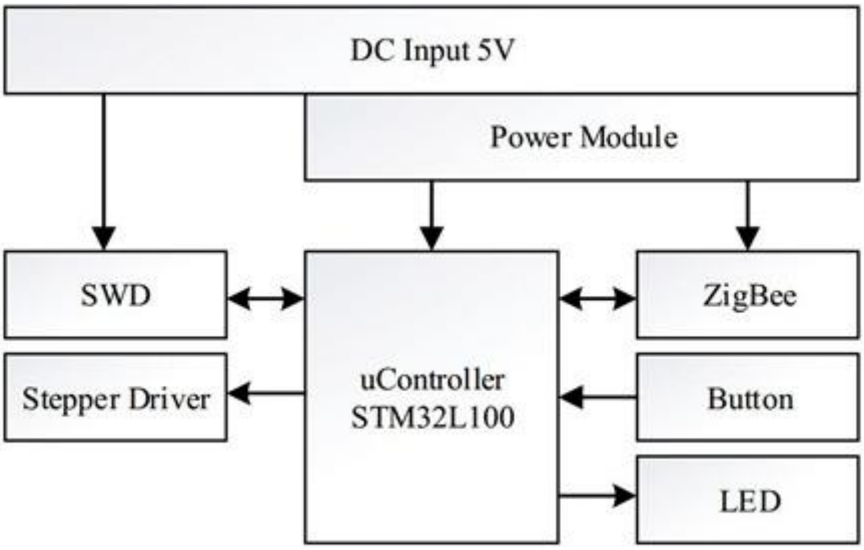

Figure 3.2 existing method using STM-32 controller

3. **Single Computer Microprocessor:** The figure 3.3 explains the integration of a single computer microprocessor as the processing unit in the curtain system represents a sophisticated approach to environmental sensing and control. By incorporating sensors to measure light and temperature levels in the classroom, the microprocessor can gather real-time data about the surrounding conditions. This data is then analysed to determine the most appropriate action for the curtains, whether to open or close. The adaptive nature of this system ensures that the curtains respond dynamically to changes in environmental factors, optimizing the classroom environment for comfort and functionality. For instance, if the light levels increase beyond a certain threshold or if the temperature rises significantly, the microprocessor can automatically command the curtains to close, preventing glare on the projector screen and maintaining a comfortable temperature indoors. Conversely, when light levels decrease or the temperature drops, indicating reduced sunlight or cooler conditions, the curtains can be instructed to open, allowing natural light to illuminate the classroom and potentially reducing the need for artificial lighting. Integrating a single computer microprocessor as the processing unit for sensing light and temperature in a classroom represents a streamlined and efficient approach to automating curtain control. By leveraging

sensors to monitor environmental conditions, such as light intensity and temperature, the microprocessor can gather real-time data to make informed decisions about whether to open or close the curtains. This intelligent system ensures optimal comfort and productivity for occupants by automatically adjusting the curtains to maintain an appropriate balance of natural light and thermal comfort within the classroom. Furthermore, the centralized processing capabilities of the microprocessor enable seamless coordination of curtain movements based on multiple factors, contributing to a more efficient and responsive automation solution. Overall, this integration of sensing technology with a microprocessor-based control system enhances the user experience and promotes energy efficiency in classroom environments.

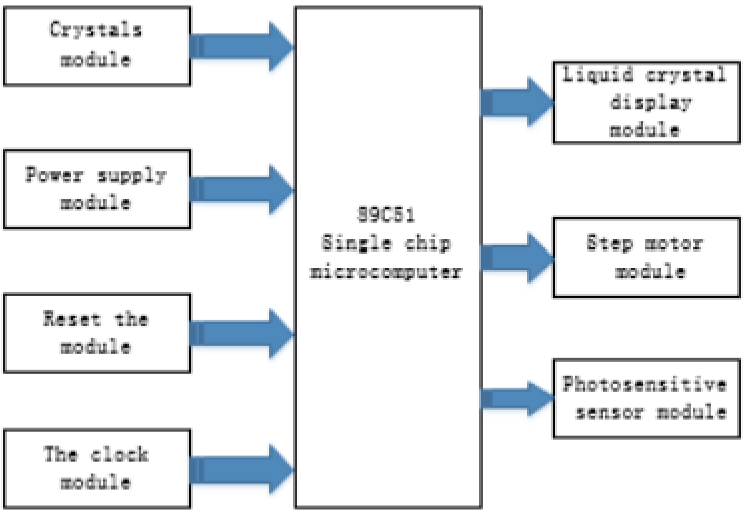

Figure 3.3 Existing method using single computer microprocessor

4. **IR Sensor-Based Curtain Controllers:** Implementing a microcontroller in conjunction with an IR sensor for curtain control. introduces a versatile and user-friendly method of interaction. By utilizing an IR sensor as a receiver, the curtain system can interpret signals sent from external devices such as smartphones with IR blasters or dedicated IR remote controls. This setup enables users to

conveniently operate the curtains using familiar interfaces, enhancing accessibility and ease of use. The integration of smartphone control adds an extra layer of convenience, allowing users to adjust the curtains remotely from their mobile devices. This flexibility accommodates modern lifestyles where smartphones are ubiquitous and central to daily activities. Additionally, the use of IR technology ensures reliable communication between the control device and the curtain system, facilitating seamless operation without the need for complex setups or additional infrastructure. The approach combines the power of microcontroller-based automation with the familiarity and convenience of smartphone control, offering users a versatile and intuitive solution for managing their curtains. Implementing a microcontroller in conjunction with an IR sensor to control the screen curtain introduces a versatile and accessible method for users to interact with the system. By utilizing an IR sensor as a receiver, the curtain system can effectively interpret commands sent either from a smartphone's IR blaster or from a traditional IR-based

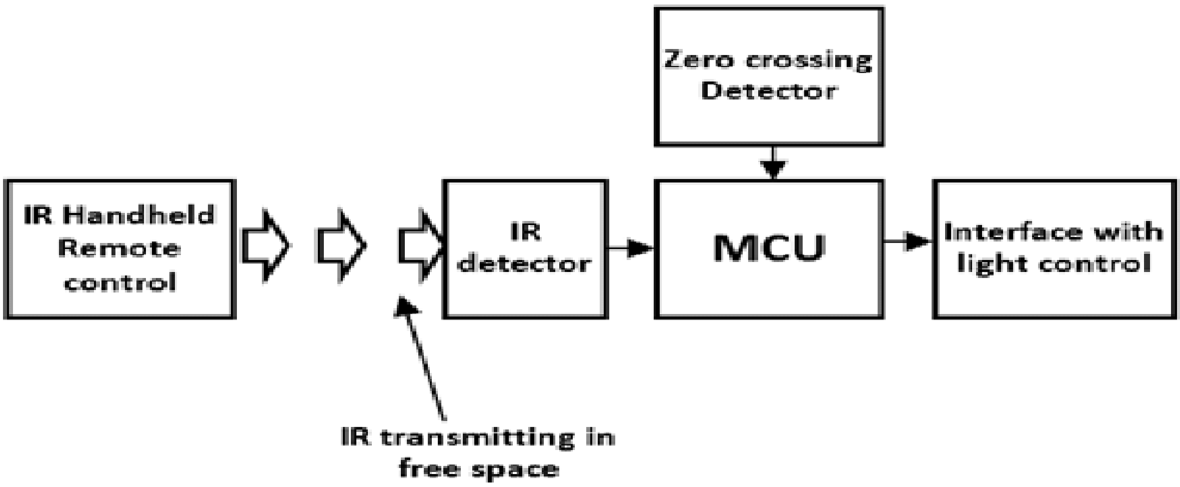

Figure 3.4 Existing method using IR sensor

remote control. This approach offers users flexibility in how they choose to interface with the system, whether it's through their smartphone or a dedicated remote control device. By leveraging the ubiquity of smartphones equipped with IR blasters, users can conveniently control the curtain system directly from their mobile devices, enhancing accessibility and ease of use. Additionally, compatibility with IR-based remote controls provides a familiar and intuitive control option for those who prefer traditional methods. Overall, this integration of a microcontroller with an IR sensor offers a user-friendly and adaptable solution for controlling the screen curtain, catering to the preferences and convenience of a diverse range of users.

3.2 PROPOSED METHOD:

The below image figure 3.5 represents the curtain automation system represents a seamless blend of mechanical and electronic components, orchestrated to deliver smooth and intuitive control over curtain movement. At its heart lies the BO mechanical gear motor, a robust and reliable workhorse designed to rotate the curtain cable with precision and grace. This motor, capable of rotation in both clockwise and counter-clockwise directions, forms the backbone of the system's functionality, enabling controlled motion essential for automating curtains.

Key to the operation of the system is the two-way switch, a fundamental element that regulates the flow of current to the motor. This switch serves as the mechanism through which the motor's direction is altered, facilitating the reversal of movement as needed. By toggling between different electrical pathways, the switch empowers users to dictate whether the curtains open or close, providing them with direct and immediate control over the system's behaviour.

Sitting between the Arduino controller and the motor is the motor driver, a vital intermediary tasked with translating signals from the Arduino into actionable commands for the motor. This component serves as the bridge between the digital instructions generated by the Arduino program and the physical movement executed by the motor. By managing the power supplied to the motor, the motor driver ensures that the motor operates within specified parameters, safeguarding against overload and optimizing performance. Moreover, the system features a high-speed alerting mechanism to warn train drivers and road users about approaching level crossings. These alerts can be delivered through digital displays or wirelessly transmitted to nearby vehicles, enhancing situational awareness and safety for all parties involved. Visual indicators, such as LEDs, provide clear and unmistakable signals regarding the status of the gates, while audible alerts in the form of buzzer sounds ensure that road users and pedestrians are adequately informed of the gate status, further enhancing safety at crossings.

Central to the operation of the entire system is the Arduino program, a sophisticated piece of software responsible for orchestrating the various components to achieve desired curtain movement. Programmed with a series of instructions tailored to specific conditions and triggers, the Arduino dictates when the motor should start, stop, or change direction based on input from sensors or other external factors. By incorporating

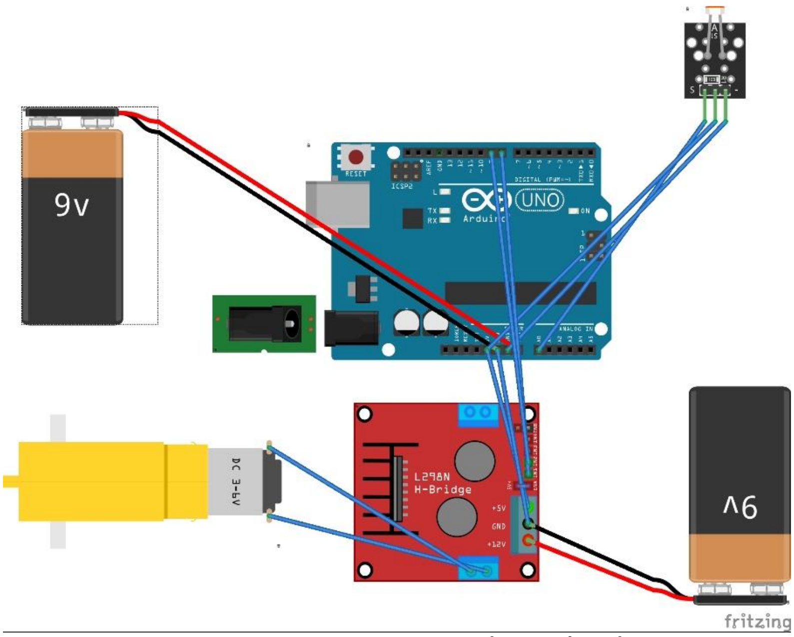

Figure 3.5 Proposed method

logic for rotation conditions, duration, and direction triggers, the Arduino program imbues the system with intelligence, allowing it to adapt to changing circumstances and user preferences. Together, these components form a cohesive ecosystem designed to deliver efficient and user-friendly control over the automated curtain system. By automating the process of opening and closing curtains, the system reduces the need for manual intervention, freeing users from mundane tasks and enhancing overall convenience. Whether adjusting curtains to regulate natural light, enhance privacy, or simply add a touch of sophistication to the environment, the system offers a seamless and intuitive solution that enhances the living experience.

The curtain automation system represents a harmonious integration of mechanical and electronic elements, orchestrated to deliver precise and efficient control over curtain movement. Through the strategic interplay of the BO mechanical gear motor, two-way switch, motor driver, and Arduino program, the system offers users a sophisticated yet user-friendly solution for automating curtains, enhancing convenience, and elevating the ambiance of any space.

CHAPTER 4

SOFTWARE AND HARDWARE REQUIREMENT

4.1ARDUINO (IDE)

The Arduino integrated development environment (IDE) is a cross- platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring.[4] The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

The Arduino IDE is incredibly minimalistic, yet it provides a near- complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an "Arduino language." However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ion. The editor has features for cutting/pasting and for searching/replacing text. The message

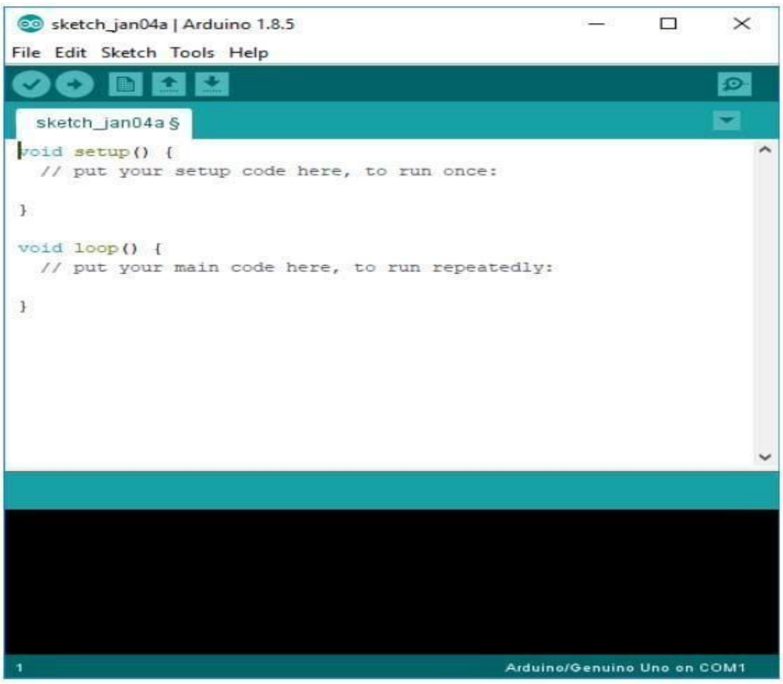

Figure 4.1 Arduino IDE

the above picture represents the figure 4.1. area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

4.1.1 CONNECTING THE ARDUINO:

Connecting an Arduino board to your PC is quite simple. On Windows:

1. Plug in the USB cable - one end to the PC, and one end to the Arduino board.

2. When prompted, select "Browse my computer for driver" and then select the folder to which you extracted your original Arduino IDE download.

3. You may receive an error that the board is not a Microsoft certified device - select “Install anyway.”

4. Your board should now be ready for programming.

When programming your Arduino board, it is important to know what COM port the Arduino is using on your PC. On Windows, navigate to Start-Devices and Printers, and look for the Arduino. The COM port will be displayed underneath.Alternatively, the message telling you that the Arduino has been connected successfully in the lower-left hand corner of your screen usually specifies the COM port is it using.

4.1.2 PREPARING THE BOARD

Before loading any code to your Arduino board, you must first open the IDE. Double click the Arduino .exe file that you downloaded earlier. A blank program, or "sketch," should open.

The Blink example is the easiest way to test any Arduino board. Within the Arduino window, it can be found under File->Examples->Basics->Blink.

Before the code can be uploaded to your board, two important steps are required.

1. Select your Arduino from the list under Tools->Board. The standard board used in RBE 1001, 2001, and 2002 is the Arduino Mega 2560, so select the "Arduino Mega 2560 or Mega ADK" option in the dropdown.

2. Select the communication port, or COM port, by going to Tools->Serial Port.

If you noted the COM port your Arduino board is using, it should be listed in the dropdown menu. If not, your board has not finished installing or needs to be reconnected.

4.1.3 LOADING CODE:

The upper left of the Arduino window has two buttons: A checkmark to Verify your code, and a right-facing arrow to Upload it. Press the right arrow button to compile and upload the Blink example to your Arduino board.

The black bar at the bottom of the Arduino window is reserved for messages indicating the success or failure of code uploading. A "Completed Successfully" message should appear once the code is done uploading to your board. If an error message appears instead, check that you selected the correct board and COM port in the Tools menu, and check your physical connections.

If uploaded successfully, the LED on your board should blink on/off once every second. Most Arduino boards have an LED prewired to pin 13 is very important that you do not use pins 0 or 1 while loading code. It is recommended that you do not use those pins ever.

Arduino code is loaded over a serial port to the controller. Older models use an FTDI chip which deals with all the USB specifics. Newer models have either a small AVR that mimics the FTDI chip or a built-in USB-to-serial port on the AVR micro-controller itself.

4.2 ARDUINO UNO

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be

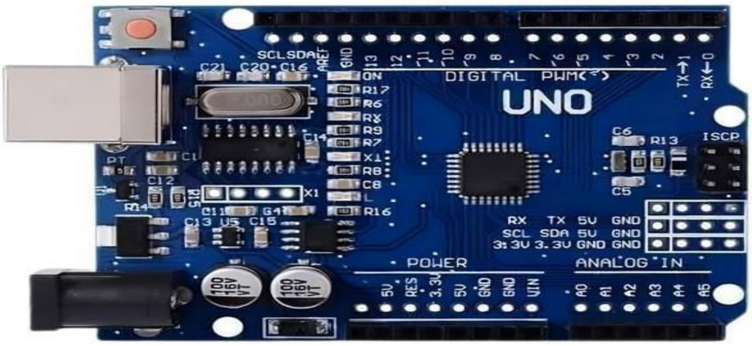

Figure 4.2 Arduino UNO

Interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases that are shown in the figure 4.2 .The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter .The Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $100, a considerable expense for many students. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis proj IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low- cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartillas, added support for the cheaper ATmega8 microcontroller pin diagram as shown in the figure 4.3 to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it Arduino. Early Arduino boards used the FTDI USB-to-serial driver chip and an ATmega168. The Uno differed from all preceding boards by featuring the ATmega328P microcontroller and an ATmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

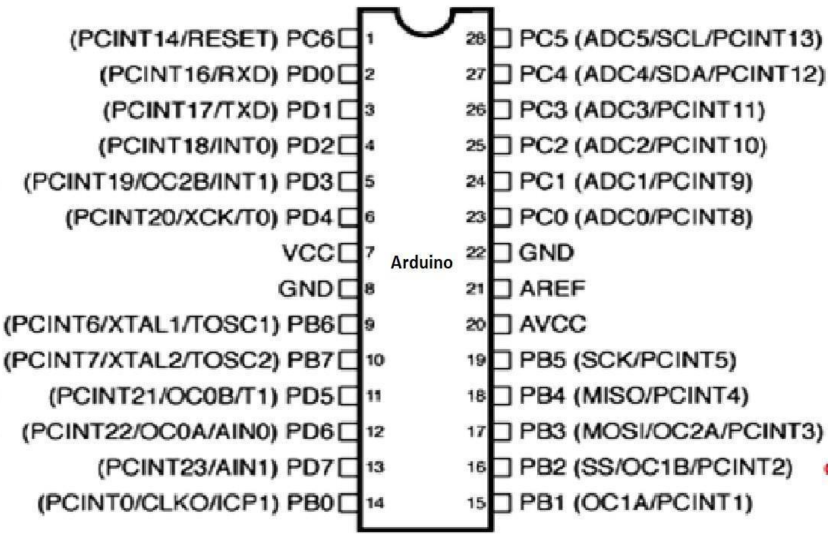

Figure 4.3 ATMEGA328p pin diagram

4.2.1 SPECIFICATION:

- **Microcontroller:** Microchip ATmega328P
- **Operating Voltage:** 5 Volt
- **Input Voltage:** 7 to 20 Volts
- **Digital I/O Pins:** 14 (of which 6 provide PWM output)
- **Analog Input Pins:** 6
- **DC Current per I/O Pin:** 20 mA
- **DC Current for 3.3V Pin:** 50 mA
- **Flash Memory:** 32 KB of which 0.5 KB used by boot loader
- **SRAM:** 2 KB
- **EEPROM:** 1 KB
- **Clock Speed:** 16 MHz
- **Length:** 68.6 mm
- **Width:** 53.4 mm

- **LED:** There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN:** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector

(5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly conFigured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset:** Typically used to add a reset button to shields which block the one on the board.

Table 4.1 Arduino UNO pinout configuration

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.

Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.

4.2.2 SPECIAL PIN FUNCTIONS

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

- **UART:** pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM (Pulse Width Modulation):** 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite () function.
- **SPI (Serial Peripheral Interface):** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI (Two Wire Interface) / I²C:** A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF (Analog Reference):** Reference voltage for the analog inputs

4.3 MOTOR DRIVER (L298N):

The figure 4.4 explains the L298N motor driver is a dual H-bridge motor driver IC that allows you to control the speed and direction of two DC motors independently. It's widely used in robotics and automation projects. Here are some key points about the L298N motor driver:

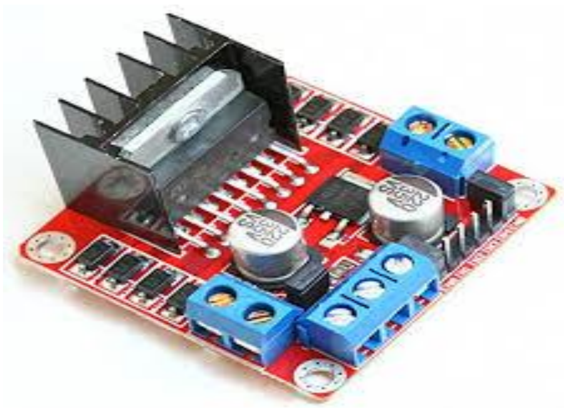

Figure 4.4 L298N Motor Driver

4.3.1 Features

Dual H-Bridge Design: It can control two DC motors or a single stepper motor.

Operating Voltage: Typically from 5V to 35V.

Maximum Current: 2A per channel continuously, with peak currents up to 3A.

Control Signals: TTL logic compatible (e.g., signals from an Arduino).

Built-in Protection: It includes built-in diodes for back EMF protection.

Table 4.2 L298N Motor Driver pinout configuration

PINS	DESCRIPTION
Vcc	Motor supply voltage (5V to 35V)
GND	Ground
5V	Logic voltage (to power the onboard 5V regulator if Vcc is higher than 7V).
IN1, IN2, IN3, IN4	Control signals for the H-bridges.
ENA, ENB	Enable pins for H-bridge A and B (used to control motor speed with PWM).
OUT1, OUT2, OUT3, OUT4	Outputs to the motors.

4.4 LDR MODULE :

The figure 4.5 is the physical look of An LDR (Light Dependent Resistor) module is used to detect light intensity. The resistance of an LDR decreases with increasing incident light intensity, making it useful for light sensing applications. The module typically includes an LDR and other components to provide a convenient interface for microcontrollers.

4.4.1 Features of an LDR Module:

Light Sensitivity: The module detects light levels and provides a variable resistance based on the light intensity.

Analog Output: Provides a voltage that varies with the light intensity.

Digital Output (optional): Some modules include a comparator that provides a digital output when the light intensity exceeds a certain threshold.

Adjustable Sensitivity: Modules with a digital output often include a potentiometer to adjust the threshold level.

4.4.2 Pin Configuration:

A typical LDR module has the following pins:

VCC: Power supply (usually 3.3V or 5V).

GND: Ground.

AO (Analog Output): Provides an analog voltage proportional to the light intensity.

DO (Digital Output): Provides a digital signal indicating whether the light intensity is above a set threshold (optional).

Figure 4.5 LDR Module

4.5 DC motor(BO motor):

the figure 4.6 gave the appearance of A DC motor, specifically a BO (Battery Operated) motor, is a small and inexpensive motor commonly used in robotics and small electronics projects. These motors are typically geared, providing a good balance between speed and torque for applications like driving small robots or mechanisms.

Figure 4.6 BO motor

4.5.1 Features of BO Motors:

Voltage Range: Typically operates between 3V to 12V.

Current Consumption: Usually low, making them suitable for battery-operated projects.

Gearbox: Often includes a built-in gearbox to reduce speed and increase torque.

Size: Compact and lightweight, easy to mount on various surfaces.

4.6 JUMPER WIRES

he figure 4.7 explains the appearance of Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. Though jumper wires come in a variety of colours, the colours don't actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colours can be used to your advantage in order to differentiate between types of connections, such as ground or power.

Figure 4.7 Jumper wires

CHAPTER 5

RESULT AND DISCUSSION

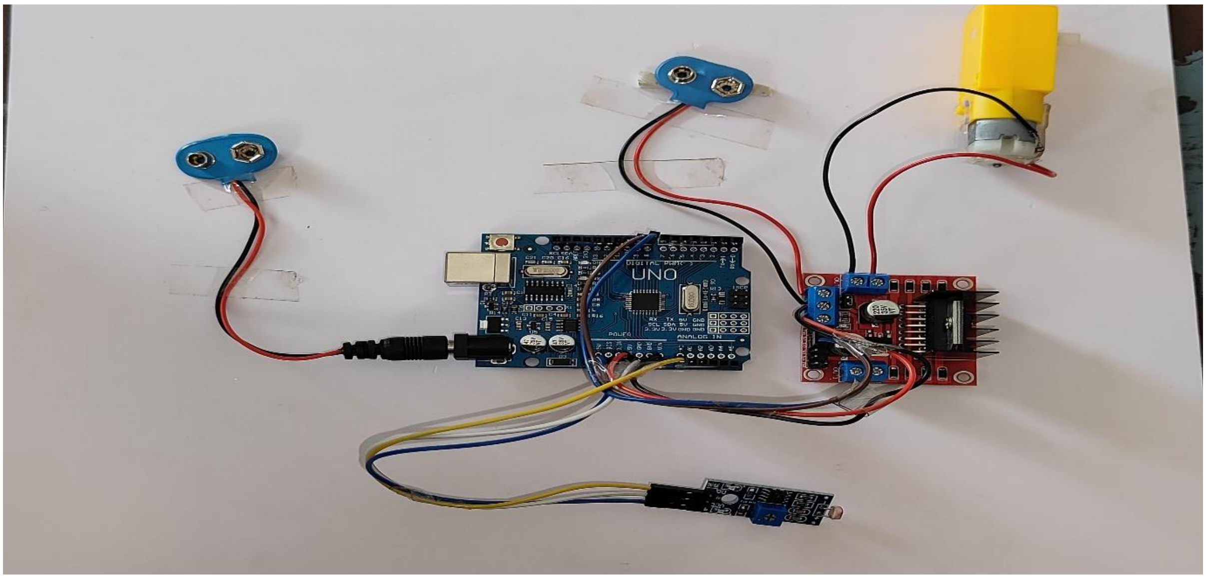

Figure 5.1 Final circuit connection

The above Figure 5.1 expalin about A smart window screen controller for a classroom, employing an Arduino Uno, L298N motor driver, LDR module, and BO motor, offers automated control based on ambient light levels. The LDR module detects light intensity, feeding data to the Arduino Uno, which processes it to determine whether the window screen should be raised or lowered. This automation enhances classroom comfort by minimizing glare and maintaining optimal lighting conditions. Safety features, including limit switches, prevent accidents by halting the motor when the window reaches its fully opened or closed position. Additionally, manual override controls may be included for user intervention. A stable power supply is crucial for reliable operation. Testing and calibration ensure accurate response to varying light conditions. Integrating these components creates a sophisticated solution that balances energy efficiency with user convenience, enhancing the classroom environment for students and educators alike.

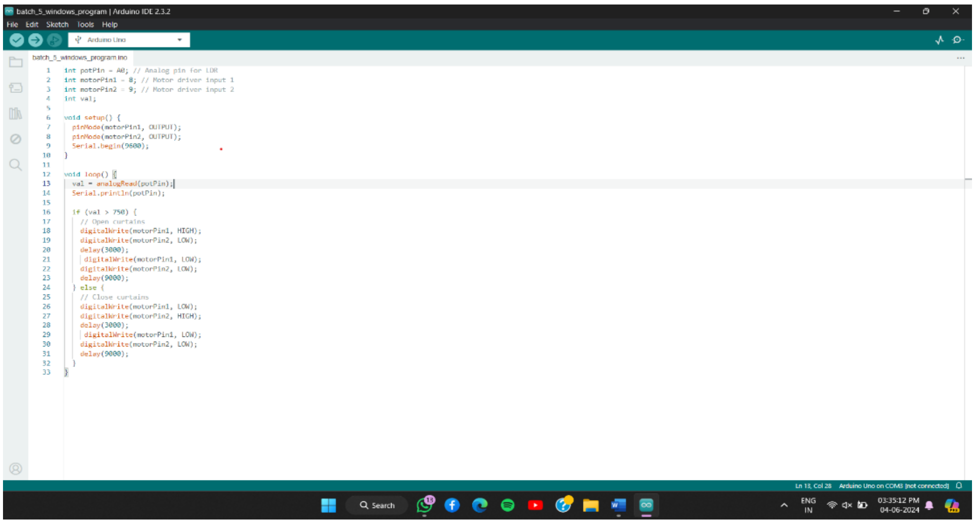

Figure 5.2 simulation Program

The Figure 5.2 explains the program logic function if the take the input pin from Arduino and connect to input 1 & 2 and connect the output 1,2 from motor driver to Bo motor and connect 12v battery to the 12 v pin of motor driver and connect the Ground of Arduino connected to motor driver ground and we gave delay to the circuit via program for gave the deviation for clockwise and anti-clockwise rotation

Figure 5.3 simulation output response

CHAPTER 6

CONCLUSION

The Smart window screen controller in classroom project engineered to deliver precise and efficient automation of curtain movement. This system, centered around the robust BO mechanical gear motor, demonstrates a reliable solution for managing classroom window screens, ensuring that the environment can be easily adapted to suit various needs. The use of a two-way switch for regulating the motor's direction adds an intuitive layer of control, enabling straightforward and immediate adjustments to the curtain position. This functionality is critical in providing users with a simple interface to open or close the curtains, enhancing the system's usability and effectiveness. The motor driver serves as a crucial intermediary, translating the digital instructions from the Arduino into physical actions by the motor. This ensures that the motor operates within safe and optimal parameters, protecting the system from potential damage due to overload and maintaining consistent performance .At the core of the system is the Arduino program, a sophisticated piece of software that orchestrates the entire operation. By incorporating logic for movement conditions, duration, and direction, the Arduino program provides the intelligence needed for adaptive and responsive curtain control. This allows the system to react to various inputs and triggers, such as changes in natural light or user preferences, thereby optimizing the classroom environment dynamically. The smart window screen controller in classroom project offers a comprehensive and user-friendly solution for automating curtain movement. This system not only enhances convenience by reducing the need for manual intervention but also contributes to a more adaptable and comfortable learning environment. By leveraging the strengths of the BO mechanical gear motor, two-way switch, motor driver, and Arduino program, the project achieves a high level of functionality and sophistication, making it a valuable addition to modern classroom settings

FUTURE ENHANCEMENT

IoT Integration:

Cloud Connectivity: Connect your Arduino Uno to a cloud platform (e.g., Adafruit IO, ThingSpeak) to remotely monitor and control the system via a smartphone app or web interface. This allows adjustments from anywhere and facilitates data logging for analysis.

Smart Home Integration: Integrate the controller with smart home systems (e.g., Amazon Alexa, Google Home) for voice control ("Alexa, open the curtains"). This enhances accessibility and convenience.

Sensor Network Expansion: Incorporate additional sensors like temperature, humidity, and CO2 to create a more dynamic environment control system. The system can automatically adjust window screens to optimize classroom conditions based on real-time sensor readings.

Advanced Sensor Technologies:

Light Spectrum Sensor: Replace the LDR with a light spectrum sensor that detects specific wavelengths. This enables the system to distinguish between projector light and ambient light, improving accuracy and preventing misinterpretations due to sunlight or room lighting changes.

Occupancy Sensor: Integrate a motion sensor to detect if the classroom is occupied. The curtains can automatically close when the room is empty for energy savings and security.

Environmental Sensors: Consider adding temperature and humidity sensors to create an automated system that adjusts the window screens to maintain comfortable classroom conditions. This can improve student focus and well-being.

Artificial Intelligence Inclusion:

Machine Learning: Train a machine learning model on historical data (light levels, room occupancy, temperature) to predict optimal window screen positions for different scenarios. This allows the system to adapt to changing conditions and user preferences over time.

Computer Vision: Implement a camera-based system with computer vision that detects projector activity based on visual cues (e.g., projector beam, presentation slides). This eliminates reliance on light sensors and provides more robust control.

REFERENCES

1. Agusta, Aria,Atmoko, Cholil,Hedayanti,Hendrawan,Nugroho,Nur,Pinem,Saifur , (2020). Designing automatic system smart curtains using Node MCU based on android in Engineering, Information and Agricultural Technology in the Global Digital Revolution (pp.112-114)
2. Dhurandher,Diego,Joel,Gabriel,Leticia,Rodrigues,Sanjay,Santos,Scarpioni,Souza(2018) An IoT Automated Curtain System for Smart Homes in 2018 international conference on advance in computing, communications and informatics (ICACCI)
3. Lui. M., Slotta. J. D. (2013). Immersive simulations for smart classrooms: exploring evolutionary concepts in secondary science. Technology, Pedagogy and Education, 23(1), 57–80.
4. V. H. C. de Albuquerque , S. K. Dhurandher , C. Rodrigues , D. A. A. Santos , G. D. Scarpioni , L. C. Souza "An IoT Automated Curtain System for Smart Homes," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 249-253.
5. Xiye Feng M. Xu (2016) "The applied research of the electric curtain control system based on the fuzzy increment PID control algorithm," 2016 3rd International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), Jinzhou, China, 2016, pp. 114-118.
6. MS. Mugdha Dharmadhikari , Pinak Sunil warankar (2023) " Research paper on Bluetooth based Home Automation using Arduino", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Volume 12 , issue 6
7. Z. Minling , N. Shaowei , L. Xia , K. Yuhui , Z. Zhiwei , L. Zifei (2023). "Design and Implementation of Smart Curtains," 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 2023, pp. 718-722.

8. Y. Gong , Pan Zhang , (2020)."Design of Curtain Automatic Control System Based on STM32," 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, United Kingdom, 2020, pp.38-41.
9. Mazumder , Riad Reza , Md. Saifur Rahman , shuvo (2021). "Smart Curtain Control System Using Smart Phone Via Bluetooth," Journal of Home Automation, vol. 12, no. 3, pp. 45-56
10. Wen sha , X. Tang , Y. Zhai, (2021). "Design of intelligent curtain based on single chip microcomputer," 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Chongqing, China, 2021, pp. 440-443.