INDEX

LIST OF FIGURES

ABSTRACT

The IoT Based Bank Security with Telegram project is a cutting-edge system designed to bolster security measures within banks by integrating IoT technology and the Telegram messaging platform. It employs a variety of hardware components such as lights, fans, lasers, sensors, and microcontrollers like the ESP32 and Arduino Uno to create a sophisticated security infrastructure. Through seamless integration with Telegram, bank administrators can remotely monitor and control security devices, receive alerts, and manage operations in real-time.

Environmental sensors like the DHT11 ensure optimal indoor conditions by regulating temperature and humidity levels through automated adjustments of HVAC systems. Laser-based perimeter detection using LDRs and laser diodes enhances security by detecting unauthorized entry attempts and triggering alarms or notifications. IR sensors facilitate visitor counting, providing insights into foot traffic patterns to optimize staffing and resource allocation. A touch sensor interface enables manual control of security devices for rapid response to security incidents or emergencies.

Overall, this project offers a comprehensive solution to address the evolving security needs of modern banking environments, empowering banks to enhance security, streamline operations, and deliver superior customer experiences.

CHAPTER-1

1.1 INTRODUCTION TO SMART BANKING USING IR AND LASER TECHNOLOGY

The modern banking sector faces numerous security challenges, ranging from physical breaches to cyber threats. As such, implementing robust security measures is imperative to safeguard assets, data, and personnel. Traditional security systems often lack the flexibility and efficiency required to address evolving security needs. With advancements in IoT technology, however, it is now possible to develop sophisticated security solutions that offer remote monitoring, control, and real-time notifications.

The IoT Based Bank Security with Telegram project aims to leverage IoT capabilities to enhance security in bank environments. By integrating various hardware components such as lights, fans, lasers, and sensors, along with the power of Telegram messaging, the system provides comprehensive security functionalities. Users can remotely control and monitor security components, receive instant notifications for security breaches, and automate certain security tasks based on environmental conditions.

This project serves as a demonstration of how IoT technology can be effectively utilized to create innovative and efficient security solutions. By combining hardware integration, sensor data processing, and cloud-based communication, the system offers a scalable and customizable approach to bank security. In the following sections, we will delve into the details of the project components, functionalities, implementation, and benefits.

1.2 OBJECTIVES

The overarching objectives of this project are multifaceted, encompassing both technical and operational dimensions. At its core, the project seeks to:

1. Design and implement an integrated IoT-based security ecosystem capable of seamlessly interfacing with the Telegram messaging platform to provide real-time monitoring, control, and alerting functionalities.

2. Develop a comprehensive environmental monitoring system leveraging DHT11 sensors to regulate ambient conditions within banking premises, thereby ensuring optimal comfort and asset protection.

3. Introduce dynamic fan speed modulation mechanisms driven by real-time temperature data analytics, enabling energy-efficient climate control while mitigating the risk of temperature-related disruptions.

4. Establish a robust laser-based security framework utilizing LDR sensors and laser diodes to detect and deter unauthorized access attempts, bolstering perimeter security and threat detection capabilities.

5. Implement an advanced visitor tracking mechanism harnessing IR sensors to accurately monitor occupancy levels, facilitate crowd management, and optimize resource allocation.

1.3 EXISTING SYSTEM

Traditional bank security systems typically consist of surveillance cameras, physical barriers, and manual security procedures. While these systems offer basic protection, they are often limited by blind spots, vulnerability to breaches, and the need for constant human monitoring. Surveillance cameras may fail to provide comprehensive coverage, physical barriers can be circumvented, and manual procedures are prone to errors and delays.

Limitations of traditional system are:

1. Lack of real-time monitoring: Traditional systems offer limited real-time monitoring capabilities, making it challenging to respond promptly to security threats.
2. Limited automation: Manual security checks and controls are time-consuming and inefficient, leading to potential security gaps.
3. Minimal remote accessibility: Traditional systems do not offer remote access for monitoring and control, limiting flexibility and responsiveness.

Overall, existing systems lack the sophistication and efficiency needed to address modern security challenges effectively.

1.4 PROPOSED SYSTEM

The proposed IoT-Based Bank Security with Telegram system represents a leap forward in security innovation, offering a holistic and integrated approach to safeguarding banking assets and ensuring operational continuity. By harnessing the power of IoT, Telegram integration, and advanced sensor technologies, the proposed system aims to deliver:

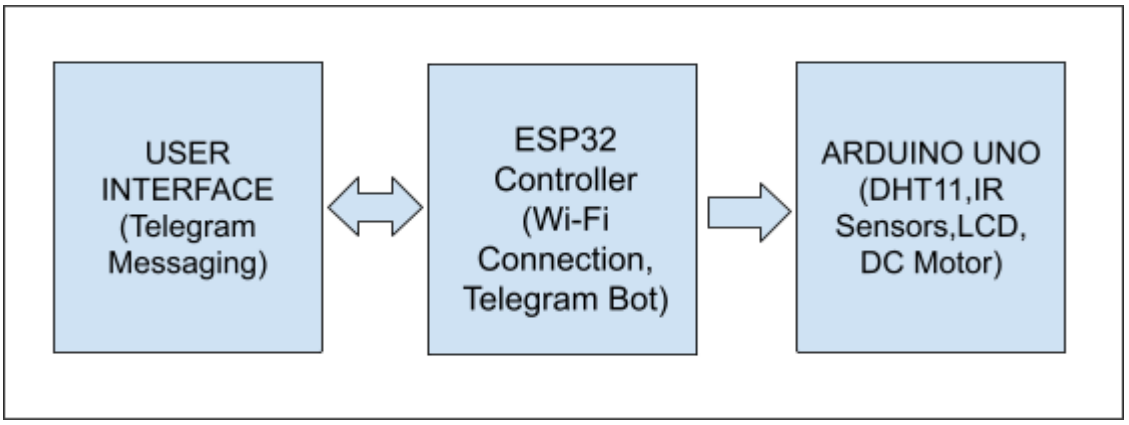


Fig.1.4.: Block Diagram of Proposed System

1. Remote Control via Telegram: Users can remotely monitor and control security parameters using Telegram commands, enabling convenient access from anywhere.

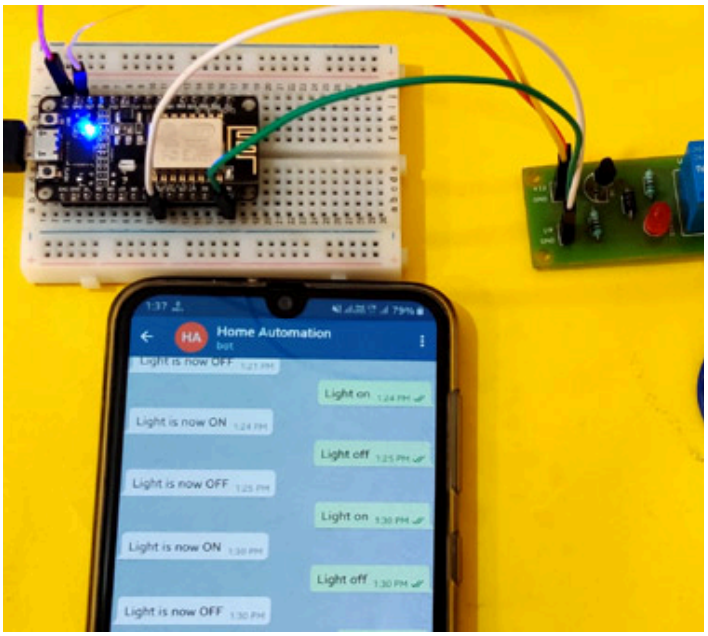


Fig.1.4.2 : ESP32 & Telegram Command Response

2. Advanced Sensor Integration: The system integrates IR sensors for visitor counting, LDR sensors for intrusion detection, and DHT11 sensors for temperature and humidity monitoring.

3. Real-time Alerts and Notifications: In the event of security breaches or environmental anomalies, the system triggers instant alerts and notifications via Telegram, ensuring swift action.
4. User-Friendly Interface: The system provides a user-friendly interface for interacting with security parameters, simplifying the monitoring and control process for users.
5. Enhanced Security Measures: By accurately counting visitors, detecting intrusions, and monitoring environmental conditions, the system enhances overall security measures within the bank premises.

By embracing innovation and leveraging the transformative potential of IoT technologies, the proposed system aims to revolutionize banking security, ushering in an era of enhanced resilience, efficiency, and peace of mind.

CHAPTER-2

2.1 LITERATURE REVIEW

The integration of Internet of Things (IoT) technology in various applications has revolutionized the way we interact with our surroundings, providing enhanced automation, monitoring, and control capabilities. In the domain of security systems, IoT-based solutions offer unprecedented levels of customization, remote access, and real-time monitoring, thereby enhancing the safety and security of residential, commercial, and industrial premises.

Several studies have explored the application of IoT in security systems, highlighting its potential to address the limitations of traditional security measures. One key aspect of IoT-based security systems is their ability to leverage interconnected sensors, actuators, and communication networks to detect, analyze, and respond to security threats in real-time.

Existing literature indicates that IoT-based security systems often employ a combination of sensors, such as infrared (IR) sensors, motion detectors, temperature sensors, and light sensors, to monitor different aspects of the environment. These sensors are integrated with microcontrollers or single-board computers, such as Arduino and ESP32, to process sensor data and trigger appropriate actions based on predefined rules or user commands.

In addition to sensor-based monitoring, IoT-based security systems frequently utilize wireless communication protocols, such as Wi-Fi, Bluetooth, or Zigbee, to facilitate remote access and control via smartphones, tablets, or web interfaces. This remote accessibility allows users to monitor their premises in real-time, receive instant notifications of security events, and remotely control security devices, such as cameras, alarms, and door locks.

Moreover, recent advancements in IoT technology have enabled the integration of artificial intelligence (AI) and machine learning algorithms into security systems, enhancing their ability to recognize patterns, detect anomalies, and adapt to changing environments. These intelligent systems can analyze large volumes of sensor data to identify potential security threats more accurately and reduce false alarms.

Furthermore, the emergence of cloud computing and edge computing technologies has facilitated the storage, processing, and analysis of sensor data on remote servers or edge devices, enabling scalable and efficient management of IoT-based security systems. Cloud-based solutions offer benefits such as data redundancy, scalability, and accessibility from anywhere with an internet connection, while edge computing enables faster response times and reduced reliance on centralized infrastructure.

Overall, the literature underscores the potential of IoT-based security systems to enhance safety, convenience, and peace of mind for homeowners, businesses, and organizations. By leveraging interconnected sensors, wireless communication, and intelligent algorithms, these systems can provide comprehensive security coverage, real-time monitoring, and remote control capabilities, thereby meeting the evolving needs of modern security applications.

CHAPTER-3

3.1 PROJECT DESCRIPTION

The IoT-Based Bank Security System is a comprehensive solution designed to enhance the security infrastructure of financial institutions using advanced IoT technology. The system integrates multiple hardware components and software functionalities to provide real-time monitoring, remote control, and proactive security measures.

At its core, the system comprises two main components: an ESP32-based controller for remote monitoring and control, and an Arduino Uno-based controller for sensor data acquisition and environmental monitoring. These controllers communicate with each other and with external devices using Wi-Fi and Telegram messaging protocols.

The ESP32 controller serves as the central hub of the system, facilitating communication with the user via Telegram commands. It connects to the internet through a Wi-Fi network and acts as a Telegram bot, receiving commands from authorized users and executing corresponding actions. Commands include turning lights, fans, and lasers on or off, as well as querying the status of various devices.

The Arduino Uno controller, on the other hand, handles sensor data acquisition and environmental monitoring tasks. It interfaces with IR sensors for visitor counting, LDR sensors for detecting intruders, and DHT11 sensors for measuring temperature and humidity levels. Sensor data is collected periodically and transmitted to the ESP32 controller for analysis and decision-making.

The system's key functionalities include remote control of lights, fans, and lasers; real-time monitoring of visitor count and environmental conditions; intrusion detection and alerting via Telegram messages; and automatic adjustment of fan speed based on room temperature. These features enable bank personnel to monitor and manage security parameters from anywhere, at any time, using their smartphones or other devices.

In terms of implementation, the system utilizes Arduino and ESP32 development boards, along with various sensors, actuators, and communication modules. Code written in Arduino IDE and the Arduino Telegram Bot library enables communication between controllers and Telegram messaging platform. Additionally, libraries such as LiquidCrystal_I2C and DHT11 are used for interfacing with LCD displays and DHT11 sensors, respectively.

Overall, the IoT-Based Bank Security System represents a significant advancement in bank security technology, offering enhanced capabilities, remote accessibility, and proactive security measures. By leveraging IoT technology, the system provides a scalable and versatile solution for ensuring the safety and integrity of bank premises and personnel.

3.2 BLOCK DIAGRAM

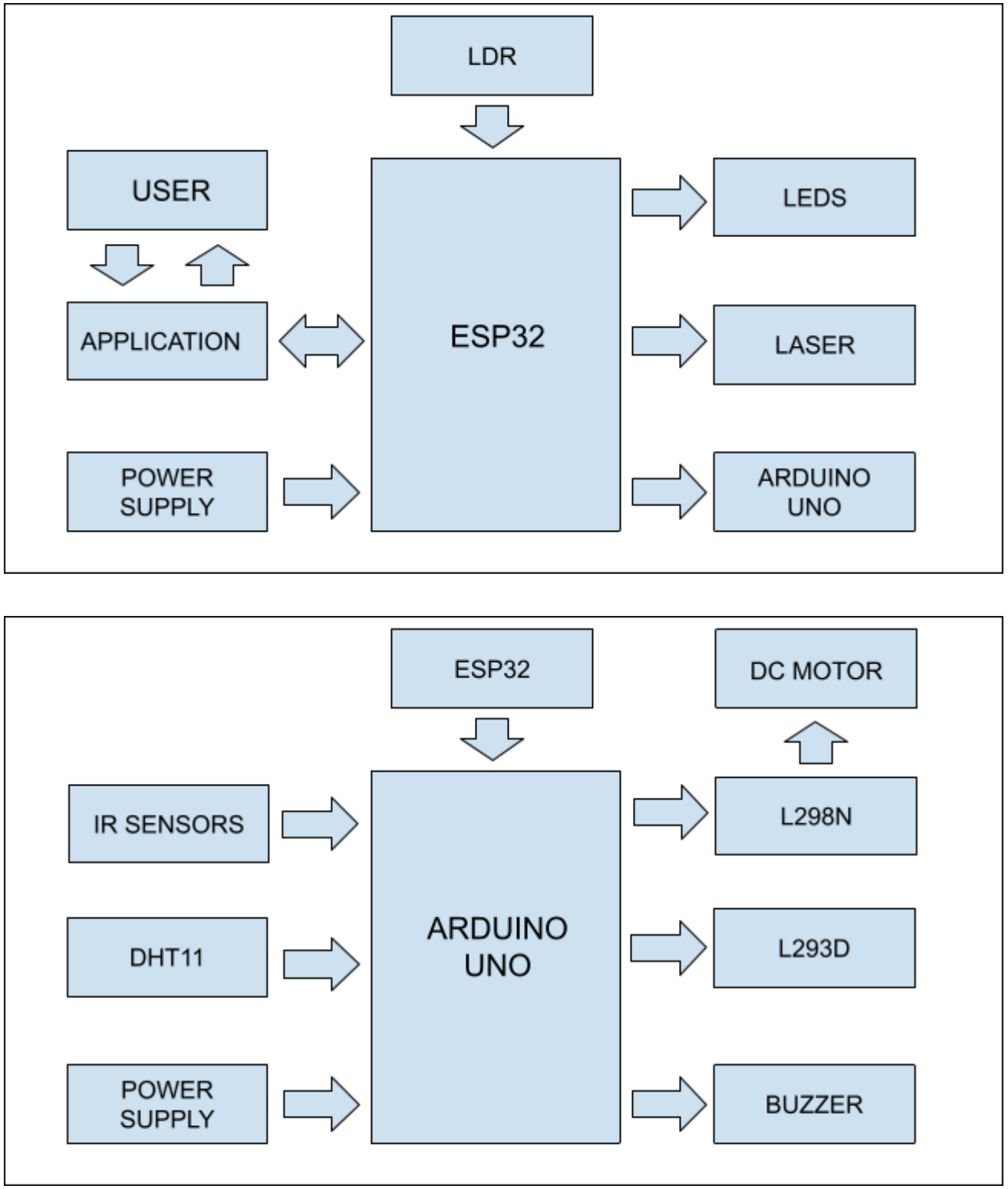


Fig. 3.2 : Block diagrams of Smart Banking using IR and Laser Technology

3.2.1 BLOCK DIAGRAM DESCRIPTION

The Block Diagram of this system consists of following section:

1. **User Interface (Telegram Messaging):**It Represents the interface through which users interact with the system using Telegram messaging. Users can send commands to control devices and receive status updates and alerts.
2. **ESP32 Controller:**It Acts as the central hub of the system, responsible for handling user commands received via Telegram messaging. It connects to the internet via Wi-Fi and serves as a Telegram bot, processing incoming messages and executing corresponding actions. Additionally, it interfaces with the IR sensor (for visitor counting) and the LDR sensor (for intruder detection).
3. **Arduino Uno:** It Handles sensor data acquisition and environmental monitoring tasks. It interfaces with the DHT11 sensor for temperature and humidity measurement, as well as with the IR sensor (for visitor counting) and the LDR sensor (for intruder detection).
4. **IR Sensor (Visitor) and LDR Sensor (Intruder):** These sensors detect visitors and intruders, respectively. The IR sensor is used for visitor counting, while the LDR sensor detects changes in light levels to identify intruders.
5. **DHT11 Sensor (Temperature, Humidity):** Measures temperature and humidity levels in the environment.

This block diagram illustrates the overall architecture of the IoT-Based Bank Security System, showing the flow of information between the user interface, controllers, and various sensors.

3.3 CIRCUIT DIAGRAM

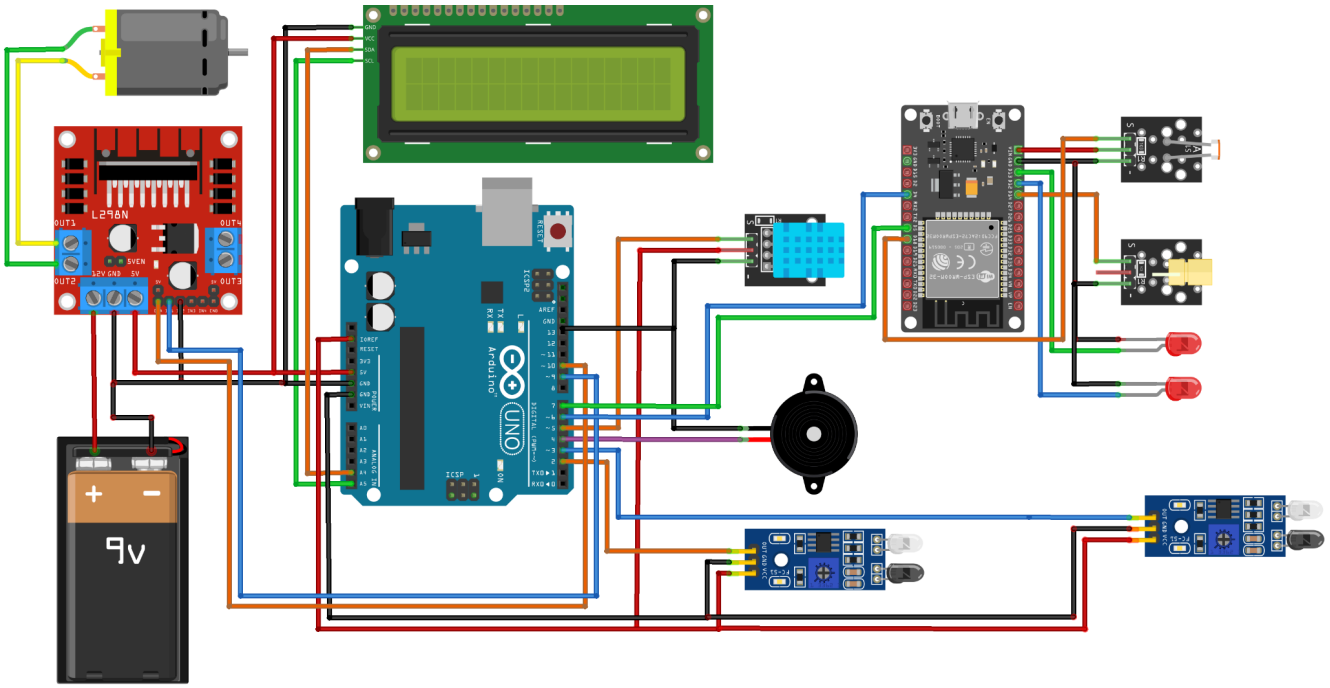


Fig. 3.3 : Circuit Diagram

3.4 HARDWARE COMPONENTS:

Hardware Components for the project are:

1. ESP32 Development Board
2. Arduino Uno
3. IR Sensors (2x)
4. LDR (Light Dependent Resistor)
5. DHT11 Temperature and Humidity Sensor
6. LEDS (for indication)
7. Buzzer
8. Laser Diode
9. DC Motor (for fan)
10. L298N Motor Driver Module

11. LCD Display with I2C Interface

12. Jumper Wires

13. Breadboard

14. Power Supply (5V DC)

15. Resistors (as required)

3.5 SOFTWARE :

Software essentials for the project are:

1. Arduino IDE (Integrated Development Environment) for programming Arduino Uno

2. ESP32 Development Environment (such as Arduino IDE or PlatformIO) for programming ESP32

3. Universal Telegram Bot Library for ESP32 (for Telegram integration)

4. ArduinoJson Library (for handling JSON data in Arduino)

5. LiquidCrystal_I2C Library (for interfacing with the LCD display)

6. DHT Sensor Library (for interfacing with the DHT11 sensor)

7. WiFi Library (for ESP32 Wi-Fi connectivity)

8. WiFiClientSecure Library (for secure Wi-Fi connections on ESP32)

9. Wire Library (for I2C communication with the LCD display)

10. LiquidCrystal_I2C Library (for interfacing with the LCD display via I2C)

11. LDR Library (if needed for additional functionality)

These components will enable you to build and program the IoT-based bank security system as described in your project requirements. Make sure to acquire all the necessary components and install the required software libraries before starting the project.

CHAPTER-4

HARDWARE DESCRIPTION

4.1 MICROCONTROLLERS

Microprocessors and microcontrollers are widely used in embedded systems products. Microcontroller is a programmable device. A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports and a timer embedded all on a single chip. The fixed amount of on-chip ROM, RAM and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.

4.2 ESP32

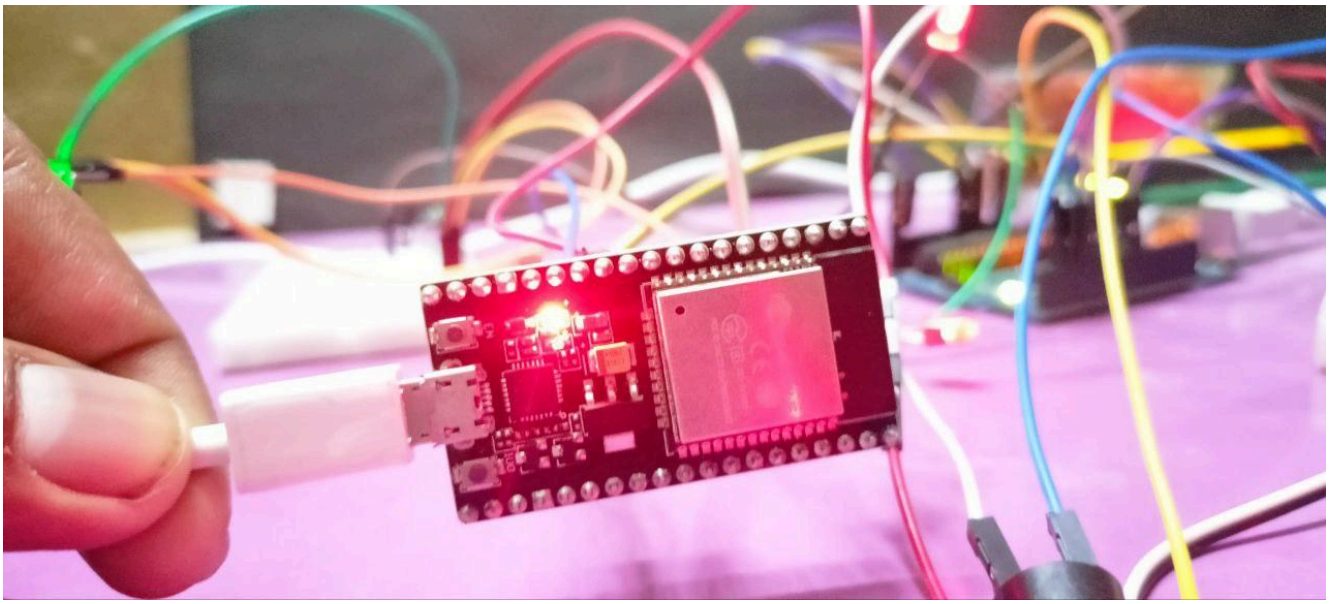


Fig 4.2.1: ESP32

The ESP32 is a series of low-cost and low-power System on a Chip (SoC) microcontrollers developed by Espressif that include Wi-Fi and Bluetooth wireless capabilities and dual-core processor.It is a tiny cheap module with a dual core 32-bit CPU and built in Wi-Fi and dual-mode Bluetooth with sufficient amount of 30 I/O pins for all basic electronics projects. It is a successor to the

ESP8266 microcontroller. All these features are very easy to use, since it can be programmed directly from the Arduino IDE.

Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

Pinout of ESP32:

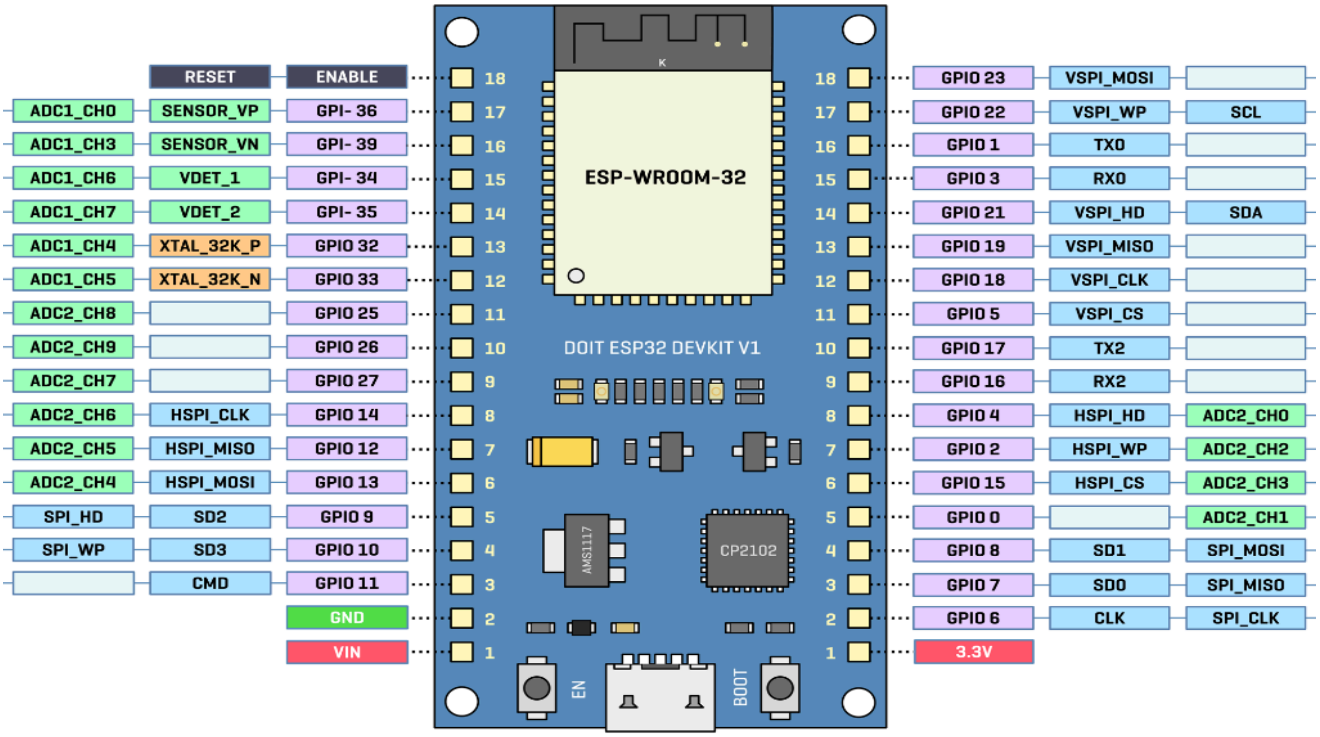


Fig 4.2.2:PIN DIAGRAM OF ESP32

Specifications of ESP32:

I made a list of some of the important specifications of ESP32 here. But for a complete set of specifications, I strongly suggest you refer to the datasheet.

- Single or Dual-Core 32-bit Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.

- 34 programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure boot and flash encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC

4.3 ARDUINO UNO

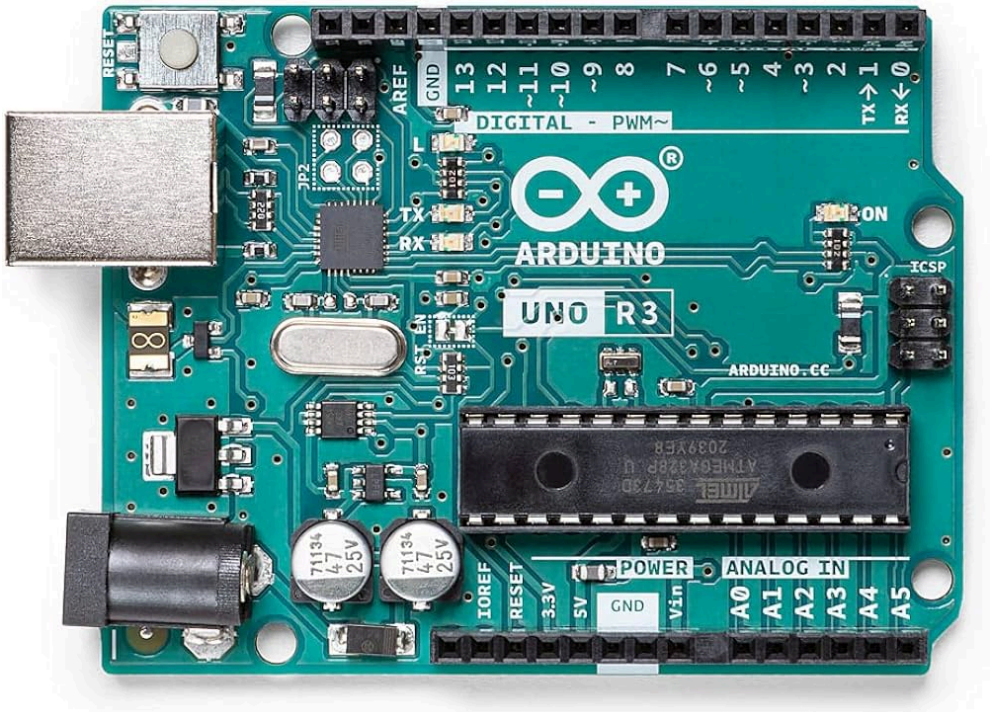


Fig 4.3.1 : Arduino UNO

The Arduino Uno is a popular microcontroller board with a specific pinout configuration. It has a total of 14 digital input/output pins, 6 analog input pins, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. The board is based on the ATmega328P microcontroller and has 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM. It can be

powered via USB connection or an external power supply. The Arduino Uno is widely used for prototyping and DIY projects due to its ease of use and versatility.

PINOUT OF Arduino UNO :

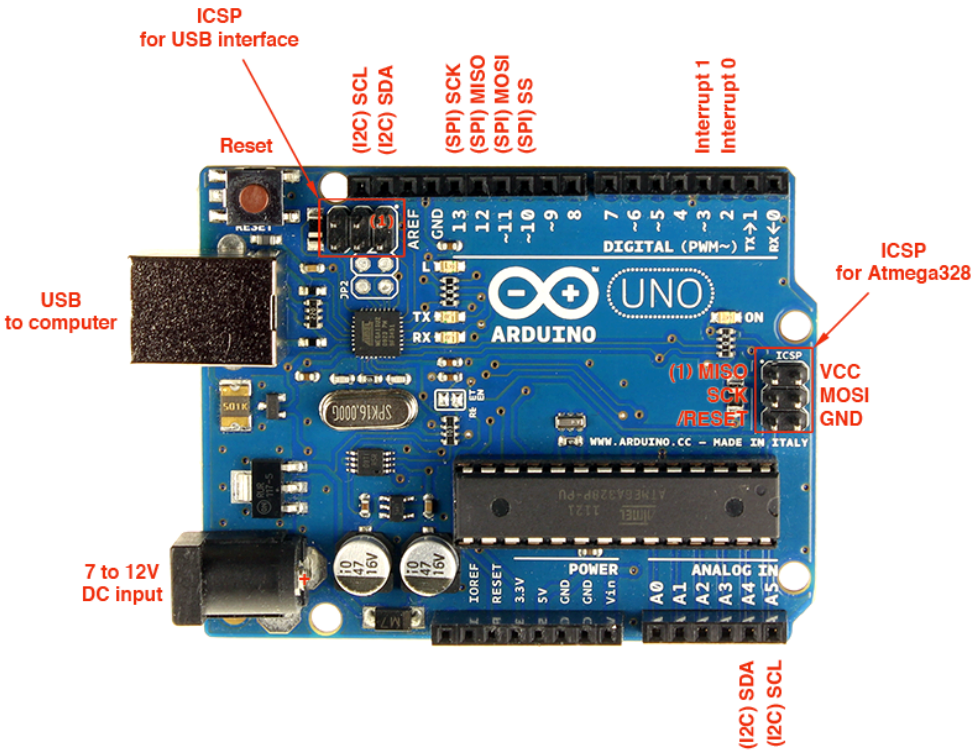


Fig.:4.3.2: Pin Diagram of Arduino UNO

1. Power Pins:

- VIN: This pin can be used to power the Arduino Uno using an external power source (7-12V).
- 5V: This pin provides a regulated 5V output for powering external components.
- 3.3V: This pin provides a regulated 3.3V output for powering low-power components.
- GND: These pins are the ground connections for the board.

2. Analog Pins:

- A0-A5: These pins can be used as analog inputs to read voltage levels from sensors or other analog devices.

3. Digital Pins:

- D0-D13: These pins can be used as digital inputs or outputs for interfacing with external components.

- PWM Pins (marked with "~"): Some of the digital pins also support Pulse Width Modulation (PWM) for controlling the intensity of output signals.

4. Communication Pins:

- RX/TX: These pins are used for serial communication with other devices.
- I2C Pins (A4/A5): These pins are used for I2C communication with other devices.

5. Reset Button: This button is used to reset the microcontroller.

6.Voltages:

- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V

Overall, the Arduino Uno is a versatile and easy-to-use microcontroller board that is suitable for a wide range of projects, from simple LED blinking to more complex robotics and IoT applications.

4.4 JUMPER WIRES

Generally, jumpers are tiny metal connectors used to close or open a circuit part. They have two or more connection points, which regulate an electrical circuit board.

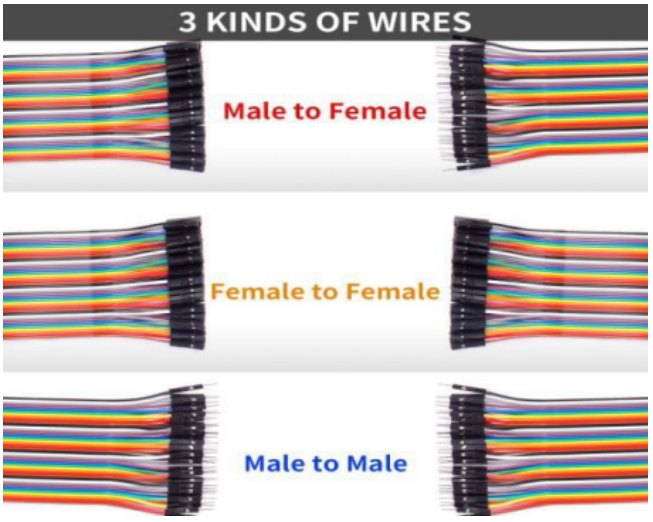


Fig 4.4: Jumper Wires

Their function is to conFig. the settings for computer peripherals, like the motherboard. Suppose your motherboard supported intrusion detection. A jumper can be set to enable or disable it. Jumper wires

are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering.

You can use jumper wires to modify a circuit or diagnose problems in a circuit. Further, they are best used to bypass a part of the circuit that does not contain a resistor and is suspected to be bad. This includes a stretch of wire or a switch. Suppose all the fuses are good and the component is not receiving power; find the circuit switch. Then, bypass the switch with the jumper wire.

How much current (I) and voltage (V) can jumper wires handle? The I and V rating will depend on the copper or aluminium content present in the wire. For an Arduino application is no more than 2A and 250V. We also recommend using solid-core wire, ideally 22 American Wire Gauge (AWG).

Types of jumper wires

Jumper Wires Come In Three Versions

- Male-to-male jumper
- Male-to-female jumper
- Female-to-female jumper

And two types of head shapes: square head and round head.

The difference between each is in the endpoint of the wire. Male ends have a pin protruding and can plug into things, while female ends do not but are also used for plugging. Moreover, a male connector is referred to as a plug and has a solid pin for centre conduction. Meanwhile, a female connector is referred to as a jack and has a centre conductor with a hole in it to accept the male pin. Male-to-male jumper wires are the most common and what you will likely use most often. For instance, when connecting two ports on a breadboard, a male-to-male wire is what you will need.

4.5 BREADBOARD

A breadboard is a fundamental tool used in electronics prototyping and experimentation. It provides a platform for building and testing electronic circuits without the need for soldering. Here's a brief overview of breadboards:

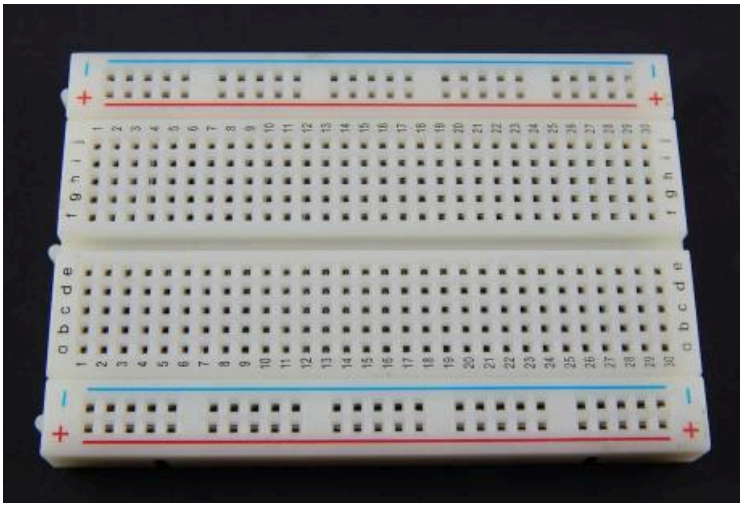


Fig 4.5 : Breadboard

Structure:

- Grid Pattern: Breadboards typically consist of a grid of holes arranged in rows and columns.
- Terminal Strips: The holes in the breadboard are connected internally in rows, called terminal strips, which run horizontally.
- Bus Strips: Adjacent terminal strips are usually connected by vertical strips, called bus strips, which run perpendicular to the terminal strips.
- Binding Posts: Some breadboards also feature binding posts for connecting external power sources.

Features:

- Reusable: Components can be inserted into the holes and removed without causing damage to the components or the breadboard.
- No Soldering: Breadboards eliminate the need for soldering, making circuit assembly and modification quick and easy.
- Versatility: Breadboards support a wide range of electronic components, including resistors, capacitors, integrated circuits (ICs), LEDs, and more.
- Prototyping: They are ideal for prototyping circuits before soldering them onto a PCB (printed circuit board).

- Modularity: Breadboards can be interconnected to form larger circuits or used individually for smaller projects.

Usage:

- Insert Components: Place electronic components, such as resistors, LEDs, and wires, into the holes on the breadboard.
- Connect Components: Use jumper wires to connect the terminals of components and create circuit connections. The jumper wires fit snugly into the holes on the breadboard.
- Test Circuits: Power the breadboard using a power source (e.g., batteries, USB power supply) and test the functionality of the circuit.
- Iterate and Modify: Breadboards allow for easy modification and iteration of circuit designs. Components can be added, removed, or rearranged as needed.

4.6 IR SENSOR

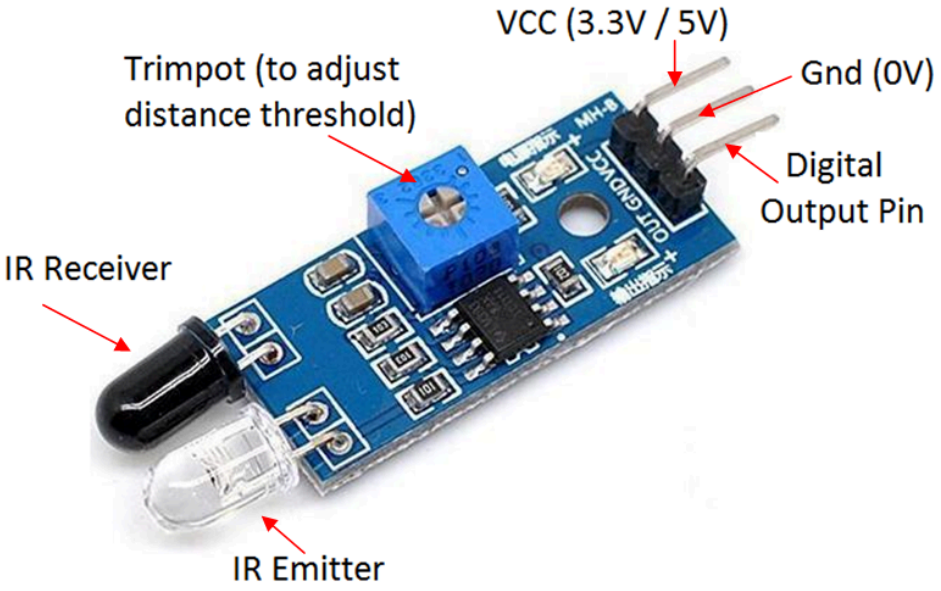


Fig 4.6: IR Sensor

An IR (Infrared) sensor is a device that detects infrared radiation in its surrounding environment. It operates based on the principle that all objects emit some level of infrared radiation, which is invisible to the human eye but can be detected by special sensors.

Pinout:

1. VCC: Connected to the positive supply voltage (typically 3.3V to 5V).
2. GND: Connected to the ground (0V) of the circuit.
3. OUT: Outputs a digital signal (typically HIGH or LOW) based on the detection of infrared radiation.
4. Enable/Control Pins: Some IR sensors may have additional pins for control purposes, such as enabling or adjusting sensitivity.

Specifications:

1. Operating Voltage: Typically operates within a voltage range of 3.3V to 5V.
2. Detection Range: The range within which the IR sensor can detect objects varies based on the specific sensor model and its design.
3. Sensitivity: Adjustable sensitivity allows for fine-tuning the detection range.
4. Response Time: Generally, IR sensors have fast response times to changes in infrared radiation levels.
5. Operating Temperature: Typically operates within a temperature range of -20°C to 70°C.
6. Dimensions: Physical dimensions may vary based on the specific sensor model and manufacturer.

Working Principle:

IR sensors work based on the principle of detecting infrared radiation emitted or reflected by objects in their vicinity. They typically consist of an IR transmitter and an IR receiver. The transmitter emits infrared light, which bounces off nearby objects and is then detected by the receiver.

When an object enters the detection range of the sensor, it reflects some of the emitted infrared light back to the receiver. The receiver detects this reflected light, and based on the intensity or presence of the received signal, it generates a corresponding digital output signal.

The output signal from the IR sensor can be used to trigger various actions in an electronic circuit, such as activating a relay, turning on a light, or sensing the presence of an object in proximity to the sensor. IR sensors find applications in various fields, including proximity sensing, object detection, motion detection, and obstacle avoidance in robotics.

4.7 LDR MODULE

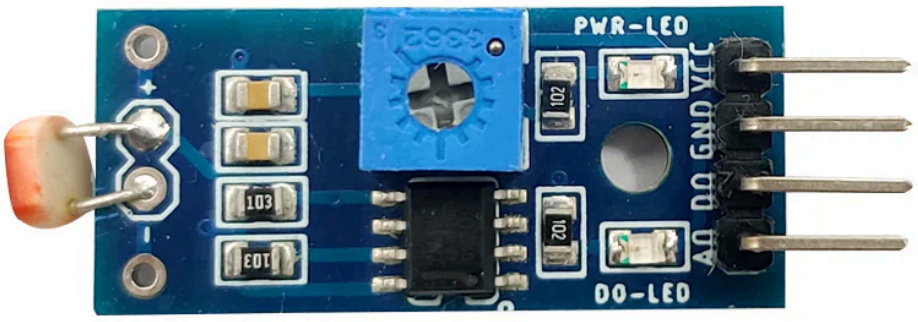


Fig 4.7 : LDR Module

A typical LDR (Light Dependent Resistor) module typically consists of an LDR sensor along with additional components for interfacing with external circuits. Here are the specifications and pinout details commonly found in such modules:

Pinout:

1. VCC: This pin is connected to the positive supply voltage (3.3V to 5V).
2. GND: This pin is connected to the ground (0V) of the circuit.
3. OUT: This pin outputs an analog voltage signal proportional to the light intensity detected by the LDR.
4. DO (Digital Output): Some modules may have a digital output pin that provides a digital signal (HIGH or LOW) based on a predefined light intensity threshold.

Specifications:

1. Operating Voltage: Typically operates within a voltage range of 3.3V to 5V.
2. Sensitivity: Sensitivity to light varies based on the specific LDR used in the module.
3. Resistance Range: Resistance varies from several kilohms in darkness to a few hundred ohms in bright light.
4. Response Time: Generally, LDRs have fast response times to changes in light intensity.
5. Operating Temperature: Typically operates within a temperature range of -20°C to 70°C.

6. Dimensions: The physical dimensions of the module may vary, but common sizes include 20mm x 20mm or similar.

4.8 : ACTIVE BUZZER

Fig. 4.8: Active Buzzer

An active buzzer is a type of electronic buzzer that generates sound when an electrical signal is applied to it. Unlike passive buzzers, which require an external oscillating signal to produce sound, active buzzers have an integrated oscillator circuit that generates the sound directly when powered.

Pinout:

- VCC: Connected to the positive supply voltage (typically 3.3V to 5V).
- GND: Connected to the ground (0V) of the circuit.
- Signal/Input Pin: Connected to a digital output pin of a microcontroller or other electronic device to control the buzzer.
- Specifications:

Operating Voltage: Typically operates within a voltage range of 3.3V to 5V.

Sound Frequency: The frequency of the sound produced by the buzzer depends on its internal oscillator circuit and design.

Sound Level: The loudness of the sound produced by the buzzer varies based on factors such as input voltage and design.

Dimensions: Physical dimensions may vary based on the specific buzzer model and manufacturer.

Working Principle:

Active buzzers contain an integrated oscillator circuit, which generates an oscillating electrical signal when power is applied. This oscillating signal drives a piezoelectric element inside the buzzer, causing it to vibrate and produce sound waves.

When a digital signal is applied to the input pin of the buzzer (typically a HIGH or LOW signal), it activates the internal oscillator circuit, causing the buzzer to produce sound at its predefined frequency. The sound continues as long as the input signal remains active.

Active buzzers are commonly used in electronic circuits to provide audible alerts, notifications, or feedback to users. They find applications in devices such as alarms, timers, countdowns, and warning systems. Their compact size, simplicity, and ease of use make them popular choices for adding sound output to various electronic projects.

4.9 : LASER DIODE:

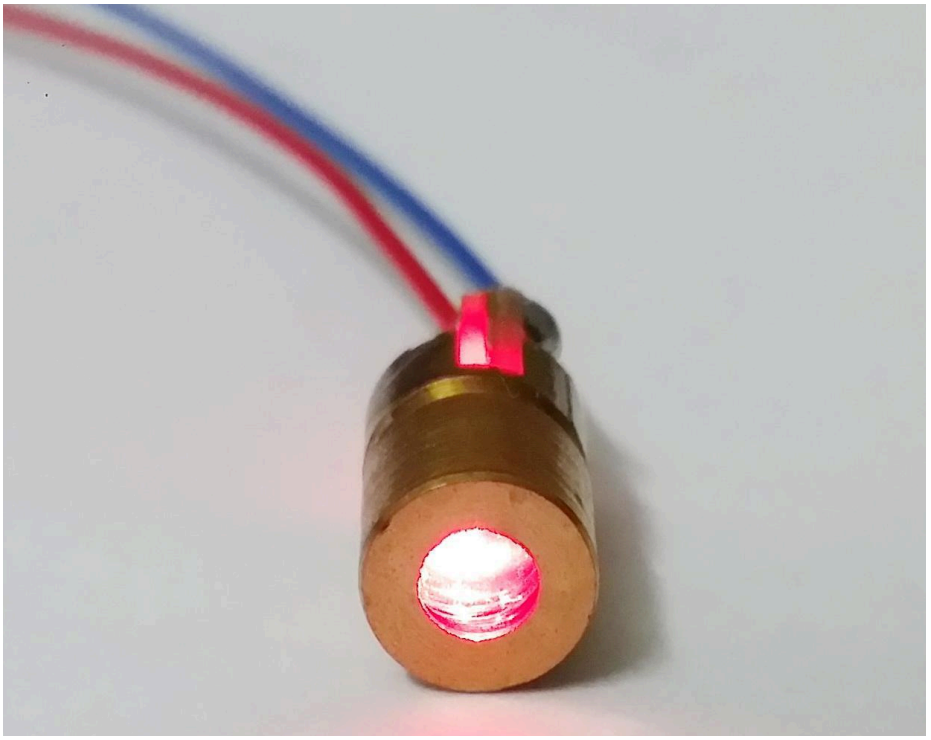


Fig 4.9 : Laser Diode

A laser diode is a semiconductor device that emits coherent light through stimulated emission of photons. It operates similarly to an LED but with a much narrower emission spectrum and the ability to produce a focused beam of light.

Pinout: Laser diodes typically have two pins:

1. Anode (A): Connected to the positive supply voltage (VCC).
2. Cathode (K): Connected to the ground (GND) of the circuit.

Specifications:

1. Wavelength: The wavelength of the laser light emitted, usually measured in nanometers (nm).
2. Operating Voltage: The voltage range required for the laser diode to operate effectively.
3. Operating Current: The current required for the laser diode to emit light at the specified wavelength.
4. Output Power: The optical power output of the laser diode, typically measured in milliwatts (mW).

Working Principle: When a forward-biased voltage is applied across the anode and cathode of the laser diode, it generates a population inversion within its semiconductor material. This leads to the emission of photons through stimulated emission, resulting in coherent laser light output.

4.10: LEDS (LIGHT-EMITTING DIODES):

Fig. 4.10 : LEDs

LEDs are semiconductor devices that emit light when current flows through them. They are widely used in various applications for indicator lights, displays, illumination, and more.

Pinout: LEDs typically have two pins:

1. Anode (A): Connected to the positive supply voltage (VCC) through a current-limiting resistor.
2. Cathode (K): Connected to the ground (GND) of the circuit.

Specifications:

1. Color: The color of light emitted by the LED, which depends on the semiconductor material used (e.g., red, green, blue).
2. Forward Voltage (Vf): The voltage drop across the LED when forward-biased.
3. Forward Current (If): The operating current required for the LED to emit light at its specified brightness.
4. Luminous Intensity: The brightness or intensity of the light emitted by the LED, measured in candelas (cd) or millicandelas (mcd).

Working Principle: When a forward voltage greater than the LED's forward voltage is applied across its anode and cathode, current flows through the semiconductor material, causing electron-hole recombination. This process releases energy in the form of photons, resulting in light emission.

4.11: L298N Motor Driver:

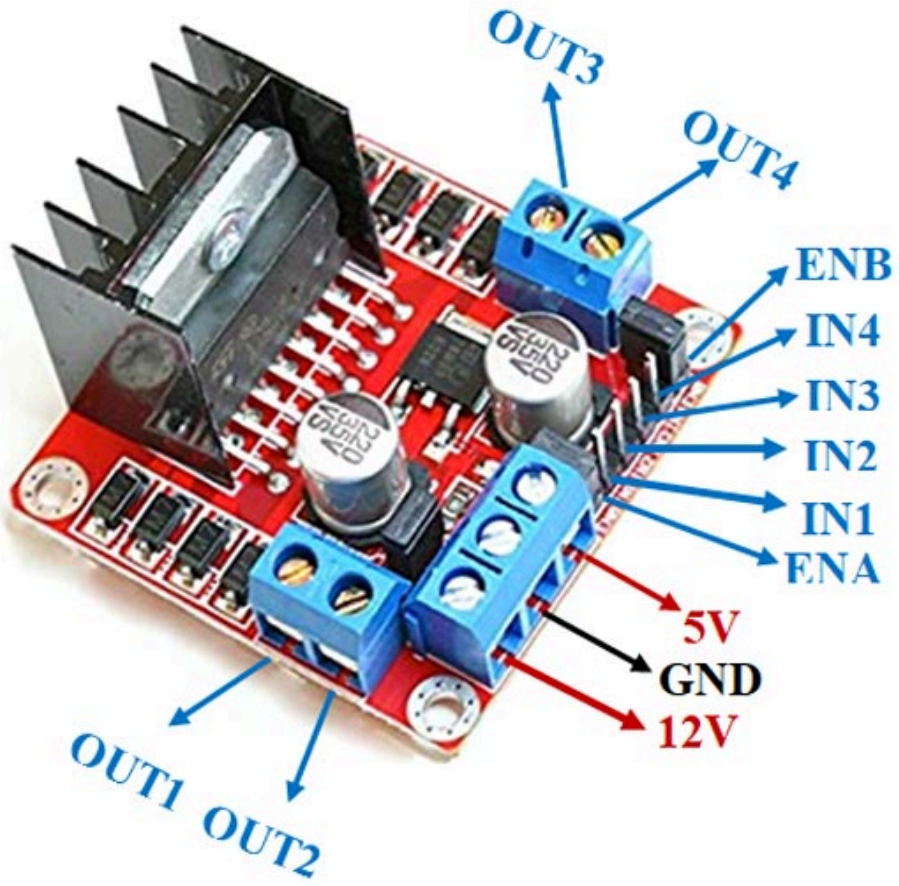


Fig. 4.11: Motor Driver L298N

The L298N is a dual H-bridge motor driver IC capable of driving two DC motors or one stepper motor. It provides bi-directional control for motors and is commonly used in robotics and motor control applications.

Pinout: The L298N typically has the following pins:

1. Input Pins (IN1, IN2, IN3, IN4): Connected to digital output pins of a microcontroller to control motor direction and speed.
2. Enable Pins (ENA, ENB): Connected to PWM-capable pins of a microcontroller to enable/disable motor outputs and control speed.
3. Motor Output Pins (OUT1, OUT2, OUT3, OUT4): Connected to the terminals of the DC motors.
4. Supply Voltage Pins (Vs): Connected to the motor power supply (typically 5V to 35V).

Specifications:

1. Maximum Continuous Output Current: The maximum current that the motor driver can deliver continuously to the motors.
2. Operating Voltage Range: The range of supply voltages over which the motor driver can operate effectively.
3. Maximum Logic Input Voltage: The maximum voltage level accepted at the input pins.

Working Principle: The L298N uses H-bridge configurations to control the direction and speed of DC motors. By controlling the input signals at the IN1, IN2, IN3, and IN4 pins, along with the PWM signals at the ENA and ENB pins, the motor driver can drive the motors forward, backward, or stop them.

4.12 : DHT11 TEMPERATURE AND HUMIDITY SENSOR:

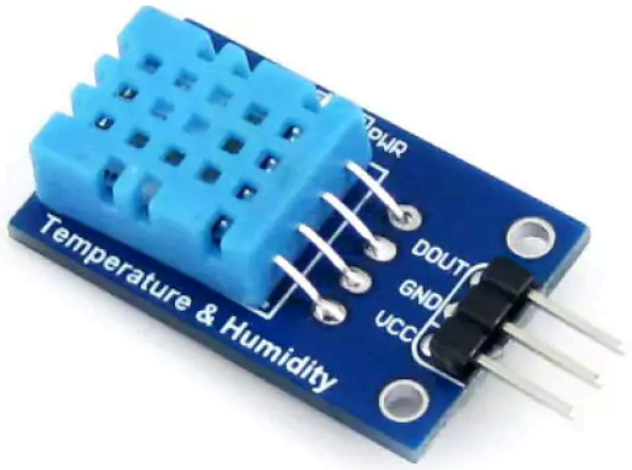


Fig. 4.12 : DHT11 Sensor

The DHT11 is a low-cost digital sensor that measures temperature and relative humidity. It provides reliable data for environmental monitoring applications.

Pinout: The DHT11 sensor typically has three pins:

1. VCC: Connected to the positive supply voltage (typically 3.3V to 5V).
2. Data Out: Connected to a digital input/output pin of the microcontroller for data transmission.

3. GND: Connected to the ground (0V) of the circuit.

Specifications:

1. Temperature Range: The range of temperatures that the sensor can measure accurately.
2. Humidity Range: The range of relative humidity values that the sensor can measure accurately.
3. Accuracy: The precision of temperature and humidity measurements provided by the sensor.

Working Principle: The DHT11 sensor utilizes a capacitive humidity sensor and a thermistor to measure humidity and temperature, respectively. It converts the analog sensor readings into digital signals, which are then transmitted to the microcontroller via a single-wire serial interface.

4.13: LCD (Liquid Crystal Display) with I2C Backpack:

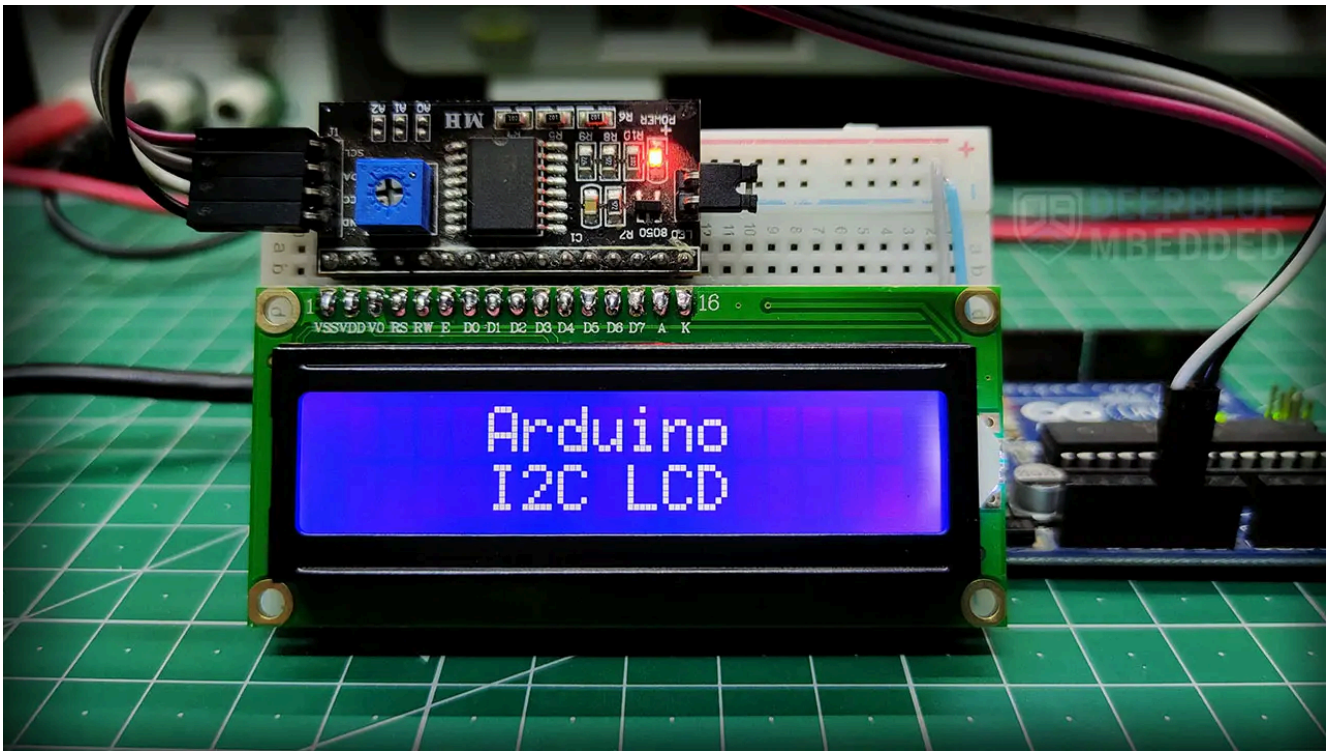


Fig 4.13 : LCD with I2C

An LCD with an I2C backpack is a combination of a liquid crystal display and an I2C communication module. It simplifies the interfacing of the LCD with microcontrollers by reducing the number of required digital pins and providing a convenient serial interface.

Pinout: The LCD with I2C backpack typically has four pins:

1. VCC: Connected to the positive supply voltage (typically 5V).
2. GND: Connected to the ground (0V) of the circuit.
3. SDA (Serial Data): Connected to the I2C data line of the microcontroller.
4. SCL (Serial Clock): Connected to the I2C clock line of the microcontroller.

Specifications:

1. Display Size: The dimensions of the LCD screen in terms of rows and columns.
2. Communication Protocol: The communication protocol used for data transmission between the LCD and the microcontroller (typically I2C).
3. Backlight: The presence of an LED backlight for improved visibility in low-light conditions.

Working Principle: The LCD with I2C backpack receives serial data from the microcontroller via the I2C communication protocol. It decodes the incoming data and drives the liquid crystal pixels on the display to generate text, numbers, or custom graphics. The backlight, if present, can be controlled separately to adjust the display's brightness.

4.13 : POWER ADAPTER(5V)

Power Adapter is a source of input supply to the project. It is an essential thing in any project. If we provide you all your project requirements, how can you miss the power adapter? Here in this category,

Fig.:4.13:-Power Adapter

we stock power Adapter/ DC power supply with differential power output ratings ranging from 5V to 12V dc @ 1A to 5A. You can also find the replacement cable for your charger or Adapter.

4.14: BATTERIES

Fig 4.14: Batteries

Lithium-ion is the most popular rechargeable battery chemistry used today. Lithium-ion batteries power the devices we use every day, like our mobile phones and electric vehicles. Lithium-ion batteries consist of single or multiple lithium-ion cells, along with a protective circuit board.

The anode and cathode store the lithium. The electrolyte carries positively charged lithium ions from the anode to the cathode and vice versa through the separator. The movement of the lithium ions creates free electrons in the anode which creates a charge at the positive current collector

Li-ion batteries are able to be recharged hundreds of times and are more stable. They tend to have a higher energy density, voltage capacity and lower self-discharge rate than other rechargeable batteries. This makes for better power efficiency as a single cell has longer charge retention than other battery types.

4.15:Type B USB CABLE

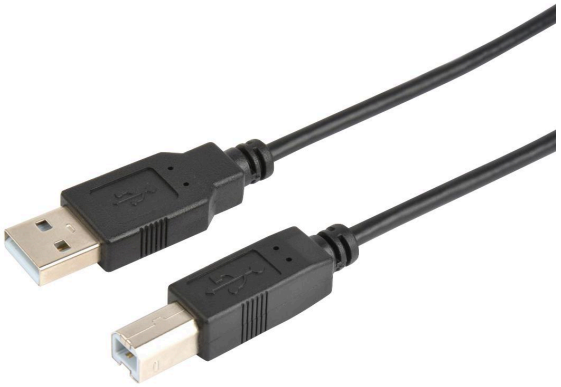


Fig 4.15: Type B USB CABLE

A Type B USB cable, also known as a USB Type B connector, is one of the standard connectors used in USB (Universal Serial Bus) technology. It is primarily used to connect peripheral devices such as printers, scanners, and external hard drives to computers and other host devices.

- The Type B USB connector features a rectangular shape with a square-like plug at one end.
- It typically has a trapezoidal shape with slightly beveled corners to ensure proper orientation during insertion.
- The connector has a wider body compared to Type A USB connectors, allowing it to accommodate more pins.

4.16:MICRO USB CABLE

Fig:4.16: Micro USB cable

A micro USB is a miniaturized version of the Universal Serial Bus interface developed for connecting compact and mobile devices, such as Smart phones,Mp3 players,Global programming system devices, Printers and digital cameras.

Micro USB connectors exist in four connector types: micro USB type A, micro USB type B, micro USB 2.0, and micro USB 3.0 Micro USB 3.0 is much like micro USB 2.0 but with an additional pin group on the side for twice the wires, enabling USB 3.0's greater speed.

As a standard USB, the micro versions are plug-and-play and hot -swappable.

CHAPTER-5

SOFTWARE COMPONENTS

The project aims to create an IoT-based bank security system that allows remote control and monitoring of various parameters such as lights, fans, lasers, and visitor count. It incorporates ESP32 and Arduino Uno microcontrollers to handle different functionalities and communicate with Telegram for remote control.

ESP32 Code:

The ESP32 code establishes a connection to the Wi-Fi network and Telegram bot using the provided credentials. It listens for incoming messages from Telegram users and interprets commands to control lights, fans, and lasers. Additionally, it detects intruders using an LDR sensor and sends warning messages via Telegram.

Arduino Uno Code:

The Arduino Uno code initializes components such as IR sensors and LDR sensors for visitor counting and intrusion detection. It reads sensor data and updates the LCD display with visitor count information. Additionally, it integrates a DHT11 sensor to measure temperature and humidity, displaying the readings on the LCD.

5.1 ARDUINO IDE :

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino boards are microcontroller-based kits that can be programmed to sense and control a variety of inputs and outputs, such as lights, motors, and sensors. They can be used to create a wide range of DIY electronic projects, from simple circuits to complex home automation systems.

Fig. 5.1: Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software application that you can use to write, upload, and debug code for Arduino boards. It is available for Windows, Mac, and Linux operating systems, and can be downloaded for free from the Arduino website.

The Arduino IDE is a simple, easy-to-use program that allows you to write code in the Arduino programming language and upload it to your Arduino board. It includes a text editor for writing code, a code compiler for converting the code into a form that the board can understand, and a serial monitor for displaying the output from the board.

Using the Arduino IDE, you can create a variety of projects with your Arduino board, from simple circuits to complex home automation systems. The IDE includes a library of pre-written code snippets and example projects to help you get started, as well as a community forum where you can ask for help and share your projects with others.

Overall, the Arduino IDE is a powerful and easy-to-use tool that makes it easy to create projects with Arduino boards.

5.1.1 HOW TO DOWNLOAD ARDUINO IDE?

To download the Arduino Integrated Development Environment (IDE), follow these steps:

1. Go to the Arduino website (https://www.arduino.cc/).
2. Click on the "Software" tab in the top menu.
3. Scroll down to the "Integrated Development Environment" section and click on the "Download the Arduino IDE" button.
4. Select the appropriate version for your operating system (Windows, Mac, or Linux).

5. Once the download is complete, open the installer file and follow the on-screen instructions to install the Arduino IDE on your computer.

Note that the Arduino IDE is a free, open-source software application, and is available for download from the Arduino website. It is not available for download from the app store on your phone or tablet.

To use the Arduino IDE, you will need to install it on a computer and use that computer to write, upload, and debug code for your Arduino board.

As you download and open the IDE software, it will appear like an image below:

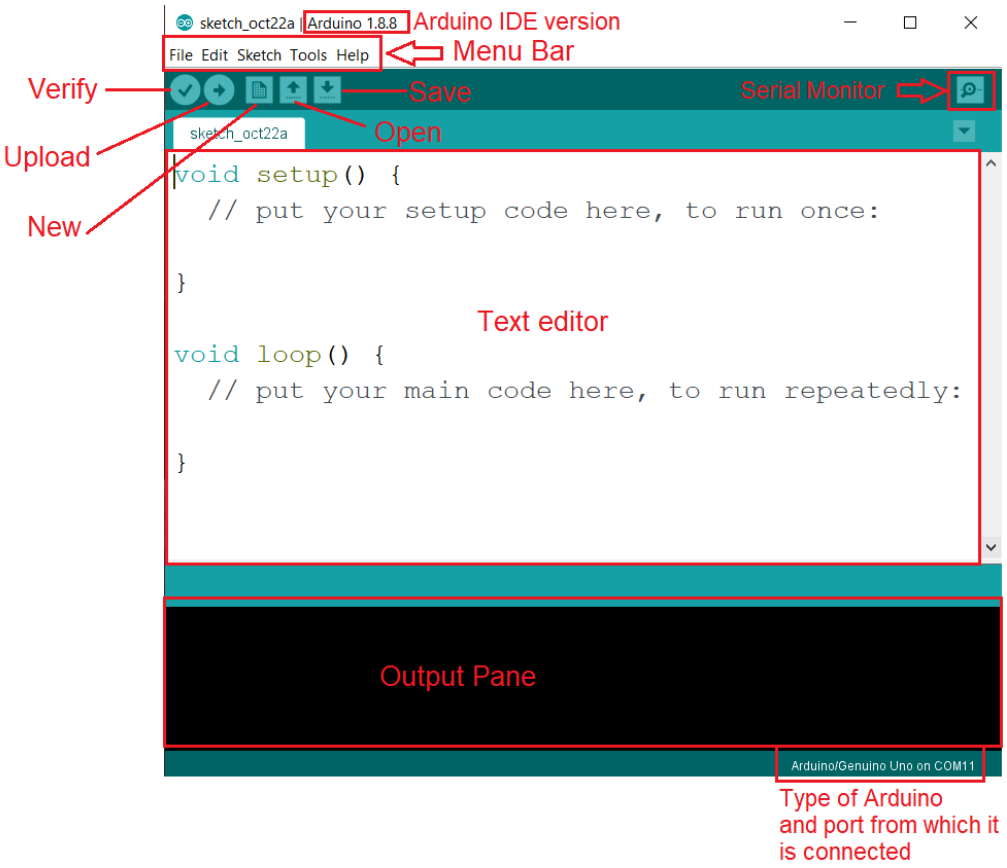


Fig. 5.1.1:Downloading Arduino IDE

Arduino IDE permits the languages C and C++ the use of special hints to set up code. The Arduino IDE uses a software program software library called Wiring from the Wiring task, which offers many, not unusual, enter and output techniques. a massive Arduino C/C++ cool animated film embodies abilities that might be compiled and related with a utility stub vital () into an executable cyclic government software:

• Setup (): a feature that runs as fast as on the start of software and which can initialize settings.

5.2 INSTALLING ESP32 BOARD SUPPORT

For programming ESP32 boards with Arduino IDE the first step is to add ESP32 board support on Arduino IDE. For this follow the steps below:

1. Before going to Arduino IDE make sure you have internet access in your computer because further steps will need internet access for installing esp32 board files on Arduino IDE.

2. Open Arduino IDE, go to “File” in the menu bar and open “Preferences”.

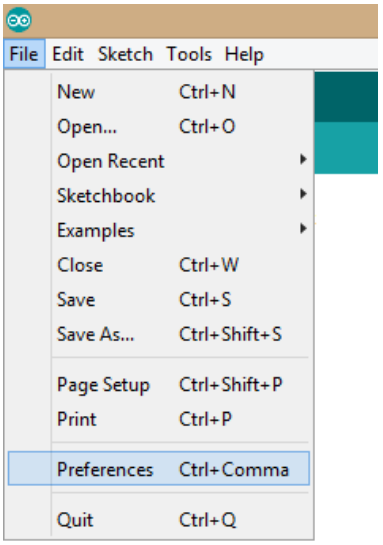


Fig. 5.2.1 : Preferences

3. As “Preferences” dialog box opens, copy the URL in “Additional Board Manager URLs” box highlighted in image below.After this select “OK”.URL-https://dl.espressif.com/dl/package_esp32_index.json

Fig. 5.2.2 : Additional Board Manager

4. Now go to “Tools>Board>Board Manager”.

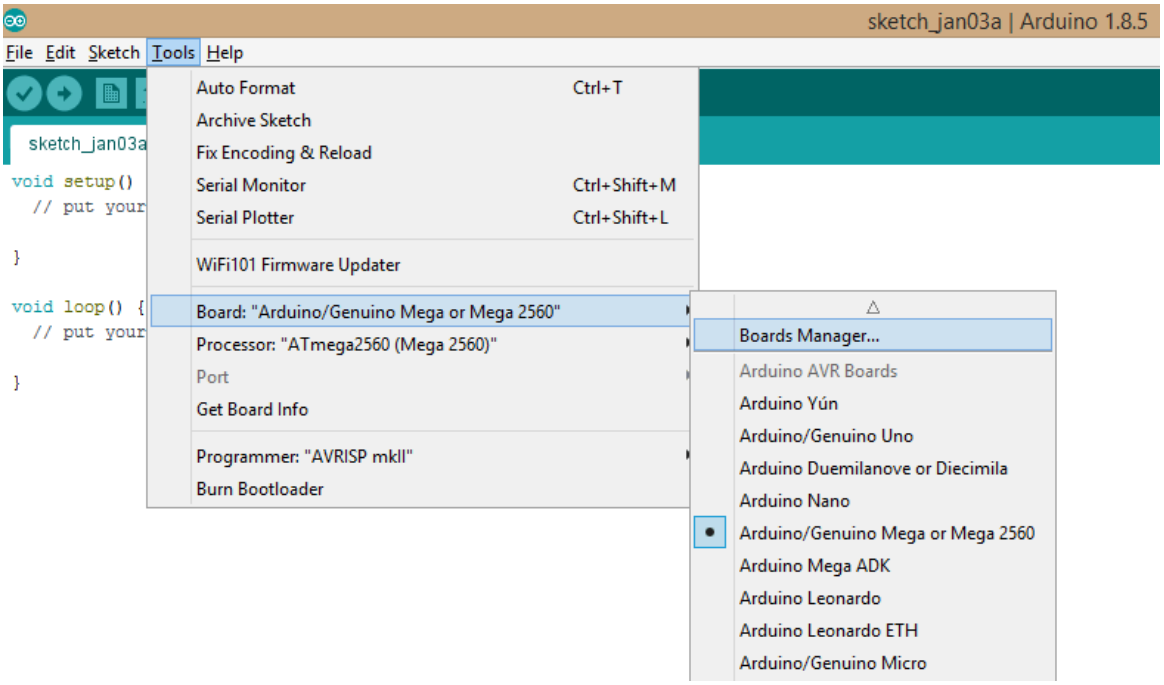


Fig. 5.2.3 : Additional Board Manager

5. As soon as you open Board Manager you can search for “esp32” in the search box.When we can see the esp32 package by Espressif Systems in Board Manager select it and then select “Install”.

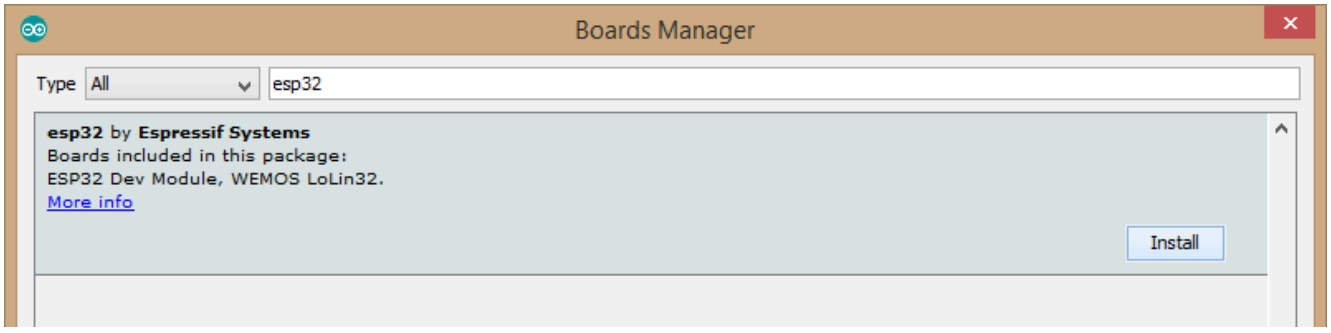


Fig. 5.2.4 : ESP32 Board Manager

6. Once installation starts, wait for a while till the installation process gets completed. As the process is completed you can see “INSTALLED” written beside the esp32 board name.

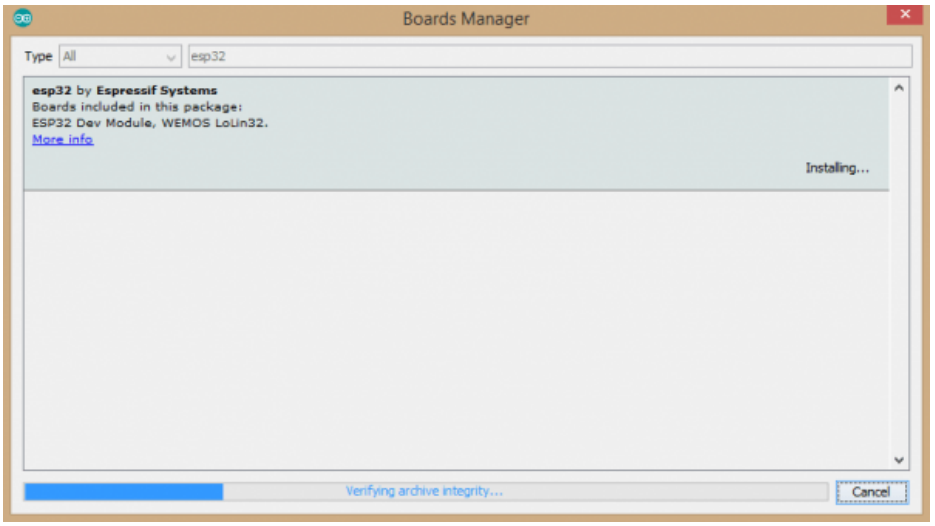


Fig. 5.2.5 : ESP32 Board Manager

7. Now close the Board Manager and goto “Tools>Board” and scroll down, there you can see a complete category of different esp32 boards under the name “ESP32 Arduino” written in grey colored fonts. Select “ESP32 Dev Module” if you are using a standard ESP32 board made by Espressif Systems or select any other depending on which board you are using.

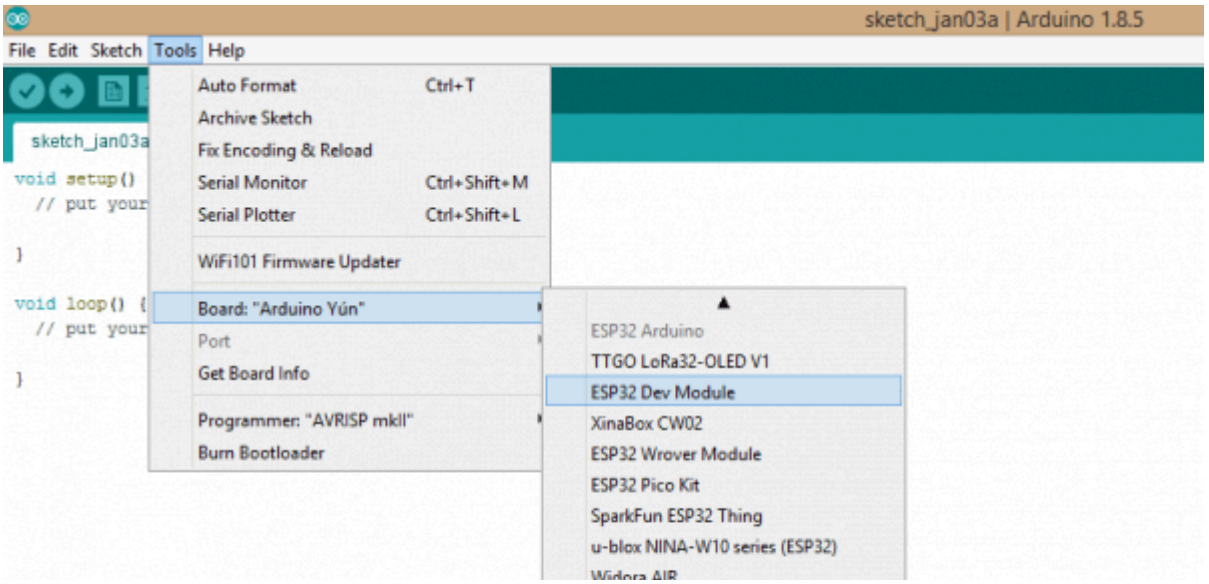


Fig. 5.2.6 : ESP32 Board Manager

8. After selecting the esp32 board you can see the board name at the bottom right corner of your Arduino screen changes to the board name selected by you. Beside board name you can see

certain other parameters,these are mainly parameters related to the code upload process on esp32. You can change them from “Tools” in the menu bar. For now let them remain as it is.

5.3 CREATE A SKETCH AND SAVE IT

To create a new sketch and save it in the Arduino IDE, follow these steps:

1. Open the Arduino IDE.
2. Click on the "File" menu at the top of the window and select "New" to create a new sketch.
3. A new editor window will appear where you can enter your code.
4. Write or copy the code you want to use for your project in the new editor window.
5. Click on the "File" menu again and select "Save As".
6. Choose the location where you want to save the sketch and type a name for the sketch in the "File name" field. Remember that the Arduino sketches have the extension .ino.
7. Click on the "Save" button to save the sketch to your computer.

You can also save your sketch with a shortcut key by pressing ctrl+S (Windows) or cmd+S (Mac)

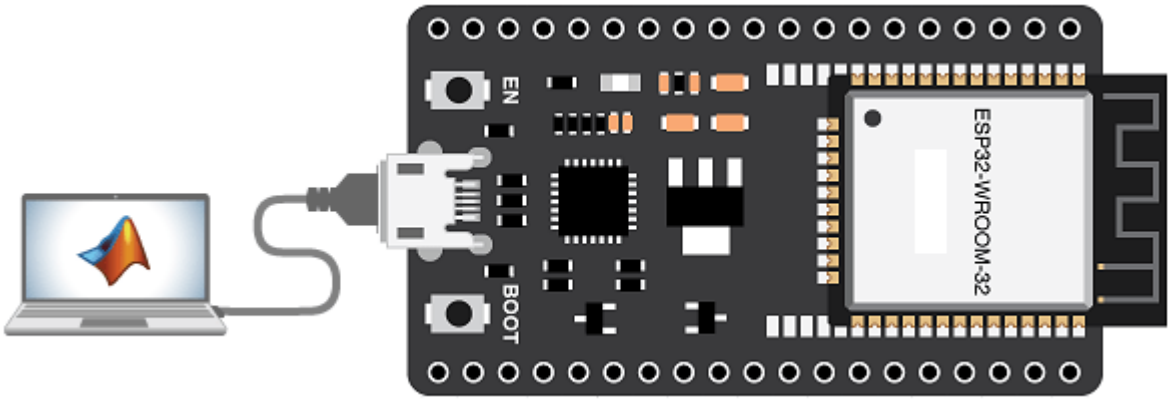
It's a good practice to save your sketch regularly while you're working on it to avoid losing any unsaved changes. Once you've saved your sketch, you can open it again later by clicking on the "File" menu and selecting "Open". Then, navigate to the location where you saved the sketch and select it.

Additionally, the Arduino IDE allows you to have multiple sketches open at the same time, you can switch between sketches using the tabs on the top of the code editor window.

It's also recommended to give meaningful names to your sketches, as it will make it easier to find and identify them later.

5.4 UPLOAD CODE IN ESP32

To upload a sketch to an ESP32 board using the Arduino IDE, you will need to first set up the ESP32 board for use with the Arduino IDE. Here's a general overview of the process:



1. Go to Tools > Board: "Board Name" > Boards Manager and search for ESP32. Select the version that you want to install.
2. Next, you will need to connect your ESP32 board to your computer using a USB cable. Once connected, you should see the ESP32 appear in the Arduino IDE under the "Port" menu.
3. Before uploading the sketch, you need to select the correct board from the Arduino IDE. Go to Tools > Board > "ESP32 Dev Module"
4. Now you're ready to upload the sketch to the ESP32. Click on the "Upload" button in the Arduino IDE, or use the shortcut key ctrl+U (Windows) or cmd+U (Mac) to upload the sketch to the board.
5. The Arduino IDE will compile the sketch and then upload it to the ESP32 board. You will see the status of the upload in the Arduino IDE's console window, and also the board's on-board LED will blink during the upload.
6. Once the upload is complete, the sketch will start running on the ESP32 board. You can check the board's output by opening the Serial Monitor on the Arduino IDE, or using the Serial communication on the ESP32 directly.

Note that if you encounter any issues uploading the sketch, ensure that you have selected the correct board and port in the Arduino IDE, and that the ESP32 board is properly connected to your computer. Also, double check that your sketch is properly written and there are no syntax errors in the code.

5.5 INTRODUCING TELEGRAM APP & CREATING BOT

Telegram Messenger is a cloud-based instant messaging and voice over IP service. You can easily install it in your smartphone (Android and iPhone) or computer (PC, Mac and Linux). It's free and without any ads. Telegram allows you to create bots that you can interact with.

"Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to Telegram Bot API". The Arduino UNO will interact with the Telegram bot to receive and handle the messages, and send responses. In this tutorial, you'll learn how to use Telegram to send messages to your bot to request a new photo taken with the Arduino UNO.

You can receive the photo wherever you are (you just need Telegram and access to the internet).

We will create a Telegram bot to interact with the Arduino UNO to request a new photo. We can request a new photo using your Telegram account from anywhere. We just need to have access to the internet on your smartphone.

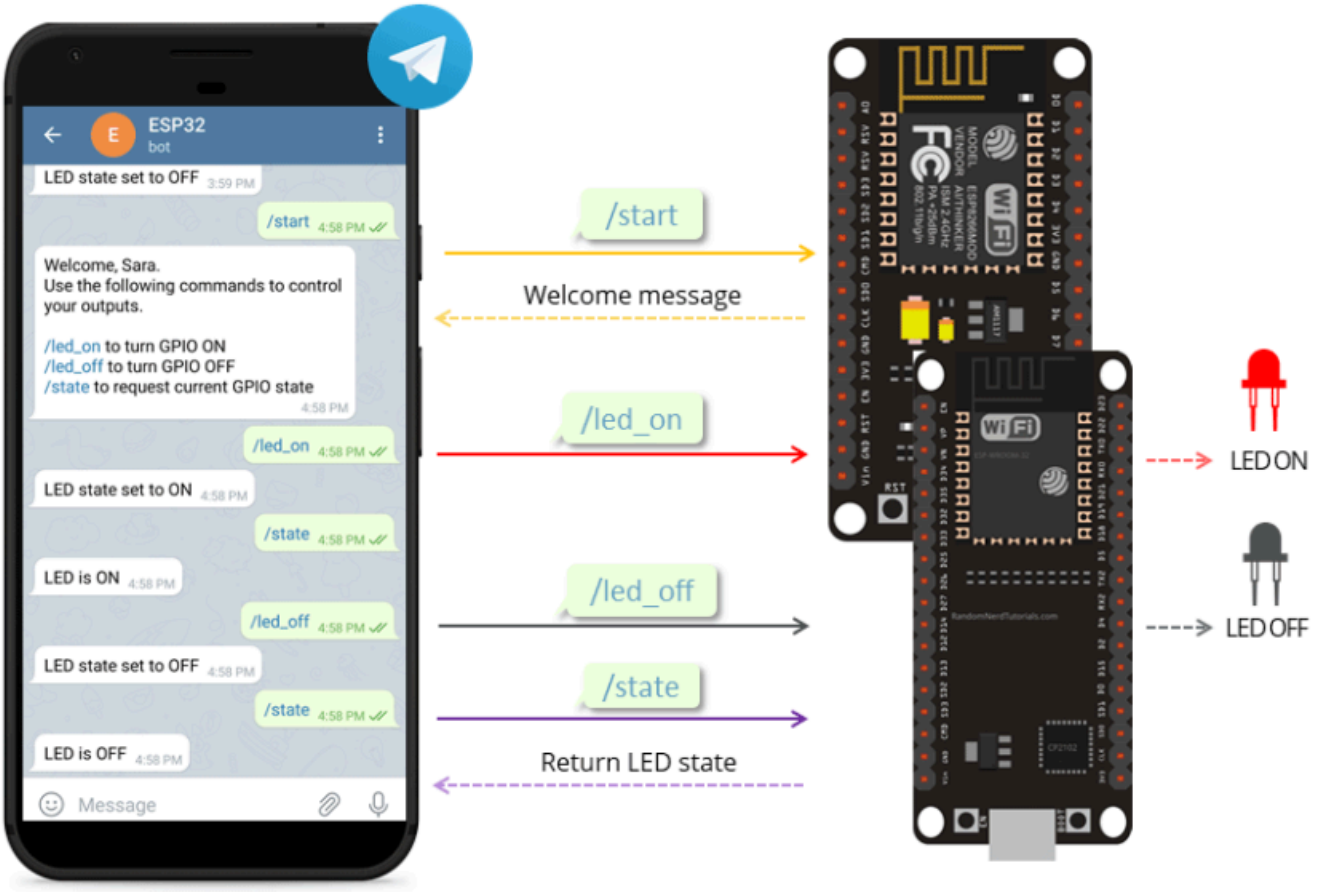


Fig.:5.5: INTERFACING THE TELEGRAM APPLICATION WITH ESP32

5.6 CREATING A TELEGRAM BOT

1. Go to Google Play or App Store, download and install Telegram.

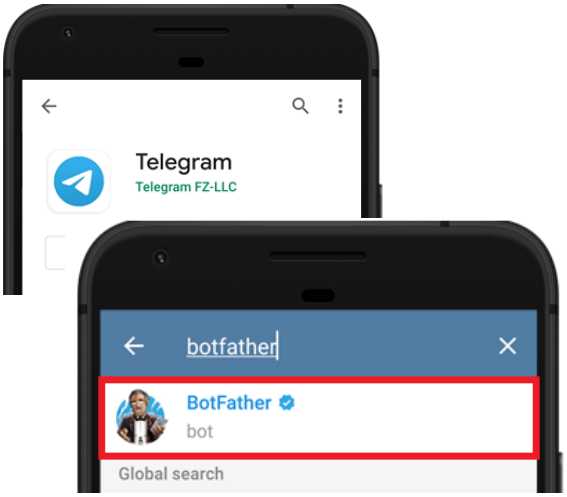


Fig.:-5.6.:INSTALLING TELEGRAM AND OPENING BOTFATHER

2. Open Telegram and follow the next steps to create a Telegram Bot. First, search for "**botfather**" and click the BotFather as shown below. Or open this link t.me/botfather in your smartphone.
3. The following window should open and you'll be prompted to click the **start** button.

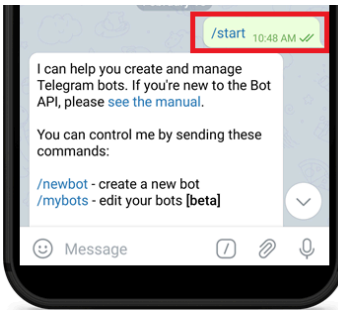


Fig.:5.6.1:-CLICK START

4. Type **/newbot** and follow the instructions to create your bot. Give it a name and username.
5. If your bot is successfully created, you'll receive a message with a link to access the bot and the **bot token**. Save the bot token because you'll need it so that the ESP32 can interact with the bot.

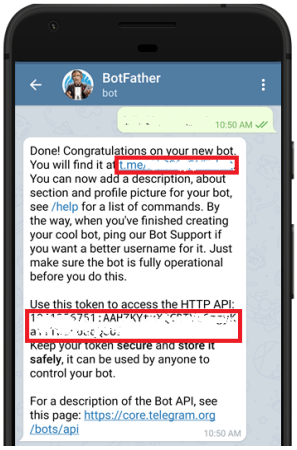


Fig.:5.6.2: CREATION OF NEW BOT

5.6 UNIVERSAL TELEGRAM BOT LIBRARY

To interact with the Telegram bot, we'll use the Universal Telegram Bot Library created by Brian Lough that provides an easy interface for the Telegram Bot API.

Download link for Universal Telegram Bot Library:

https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot

Follow the next steps to install the latest release of the library.

1. Click here to download the Universal Arduino Telegram Bot library.
2. Go to Sketch > Include Library > Add.ZIP Library...
3. Add the library you've just downloaded.

Important: don't install the library through the Arduino Library Manager because it might install a deprecated version.

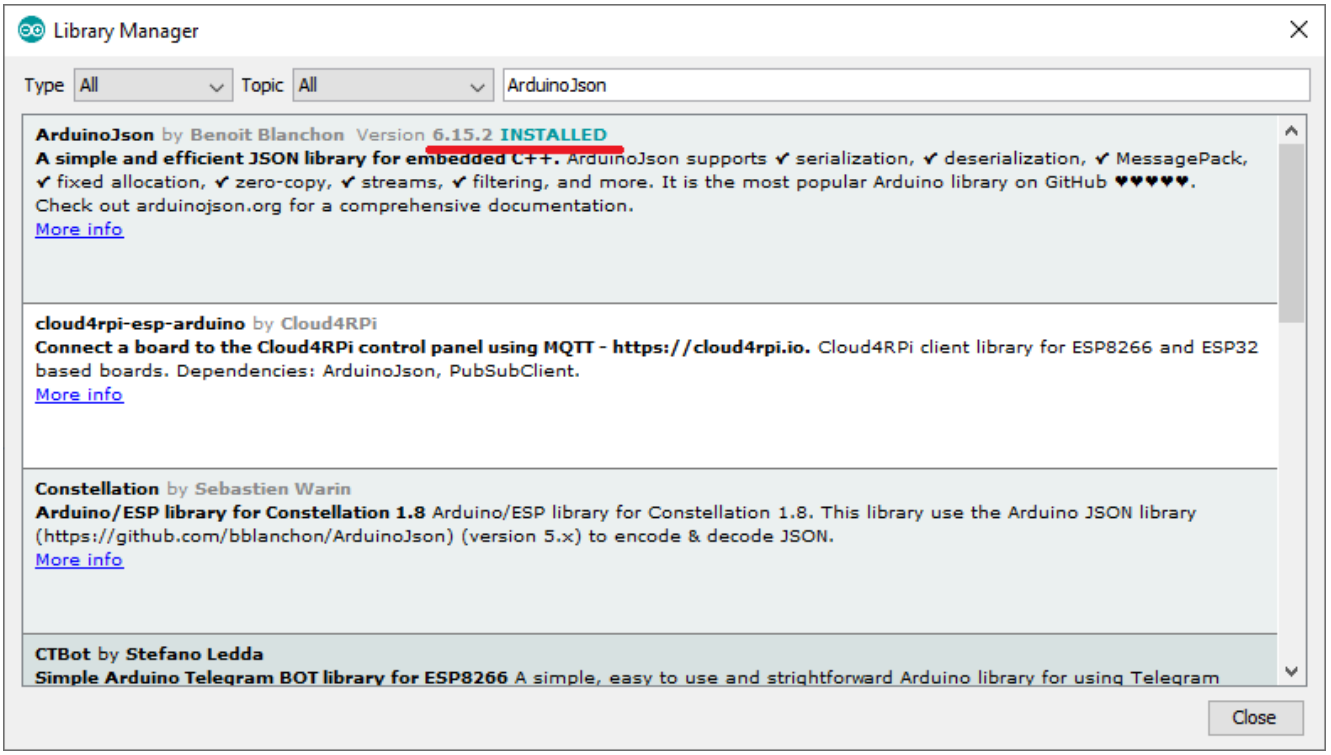
5.7 ARDUINOJSON LIBRARY

You also have to install the ArduinoJson library. Follow the next steps to install the library.

1. Go to Sketch > Include Library > Manage Libraries.

2. Search for “ArduinoJson”.

3. Install the library.



We’re using the ArduinoJson library version 6.15.2.

Fig.:5.10:-ArduinoJson Library

5.8 SOURCE CODE

The source code allows you to control your ESP32 and Arduino UNO GPIOs by sending messages to a Telegram Bot.

5.9 SET THE CODE

To make this sketch work for you, you need to

1.insert your network credentials (SSID and password)

2.insert the Telegram Bot token and your Telegram group ID.x`

5.9.1 NETWORK CREDENTIALS

Insert your network credentials in the following variables.

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";

const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

5.9.2 TELEGRAM CHAT ID

Insert your chat ID. The one you've got from the IDBot.

```
String CHAT_ID = "XXXXXXXXXX";
```

5.9.3 TELEGRAM BOT TOKEN

Insert your Telegram Bot token you've got from Botfather on the BOTtoken variable.

```
String BOTtoken = "XXXXXX:XXXXXXXXXXXXXXXXXXXXXXX";
```

5.10 UPLOADING THE CODE IN ARDUINO UNO

To upload code to an Arduino Uno, follow these steps:

1. Connect Your Arduino Uno: Use a USB cable to connect your Arduino Uno to your computer. Ensure that the USB cable is securely connected to both the Arduino Uno and your computer's USB port.
2. Open Arduino IDE: Launch the Arduino IDE software on your computer. If you haven't already installed the Arduino IDE, download and install it from the official Arduino website.
3. Select Board and Port: In the Arduino IDE, go to the "Tools" menu. Under the "Board" submenu, select "Arduino Uno" as the target board. Then, under the "Port" submenu, select the port to which your Arduino Uno is connected.
4. Verify Your Code: Click on the checkmark icon (✓) in the toolbar to verify your code. This will compile the code and check for any errors. If there are errors, they will be displayed in the console at the bottom of the Arduino IDE window.
5. Upload Your Code: Once your code has been successfully verified without any errors, click on the arrow icon (→) next to the checkmark icon. This will upload your code to the Arduino Uno.

6. Monitor Upload Progress: During the upload process, you'll see a progress bar at the bottom of the Arduino IDE window. Wait for the upload to complete. Once the upload is finished, you should see a message in the Arduino IDE indicating "Done uploading."
7. Verify Upload: After the upload is complete, disconnect the Arduino Uno from your computer (if needed) and power it using an external power source, such as a battery or a USB power adapter. Verify that your code is running correctly on the Arduino Uno by observing its behavior or connecting any required peripherals.

That's it! You've successfully uploaded your code to the Arduino Uno, and it should now be executing the instructions you've written. If you encounter any issues during the upload process, double-check your connections, ensure that the correct board and port are selected in the Arduino IDE, and review any error messages displayed in the console.

5.11: SYSTEM DESIGN AND INTERACTIONS VIA UML DIAGRAMS

5.11.1 Class Diagram of the system :

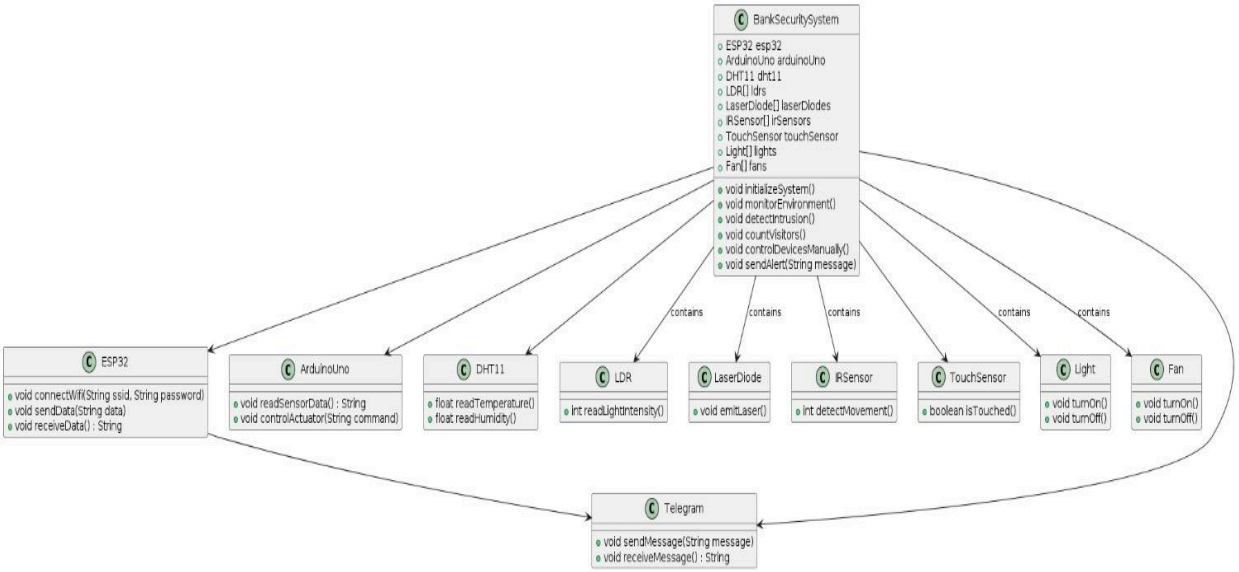


Fig: 5.11.1 Class Diagram

5.11.2 Use Case of the System

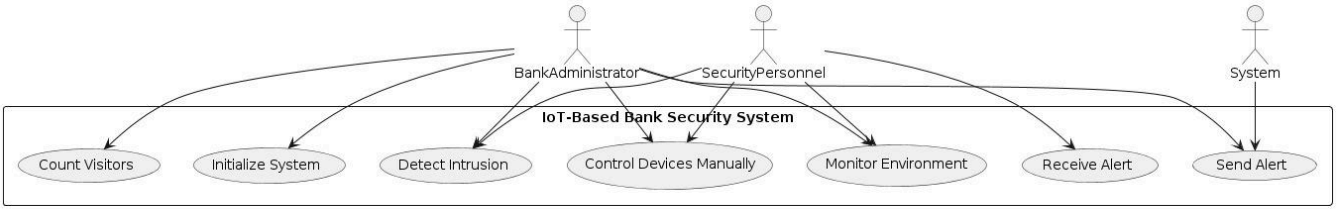


Fig 5.11.2 : Use Case Diagram of the system

5.11.3 Sequence of Operations

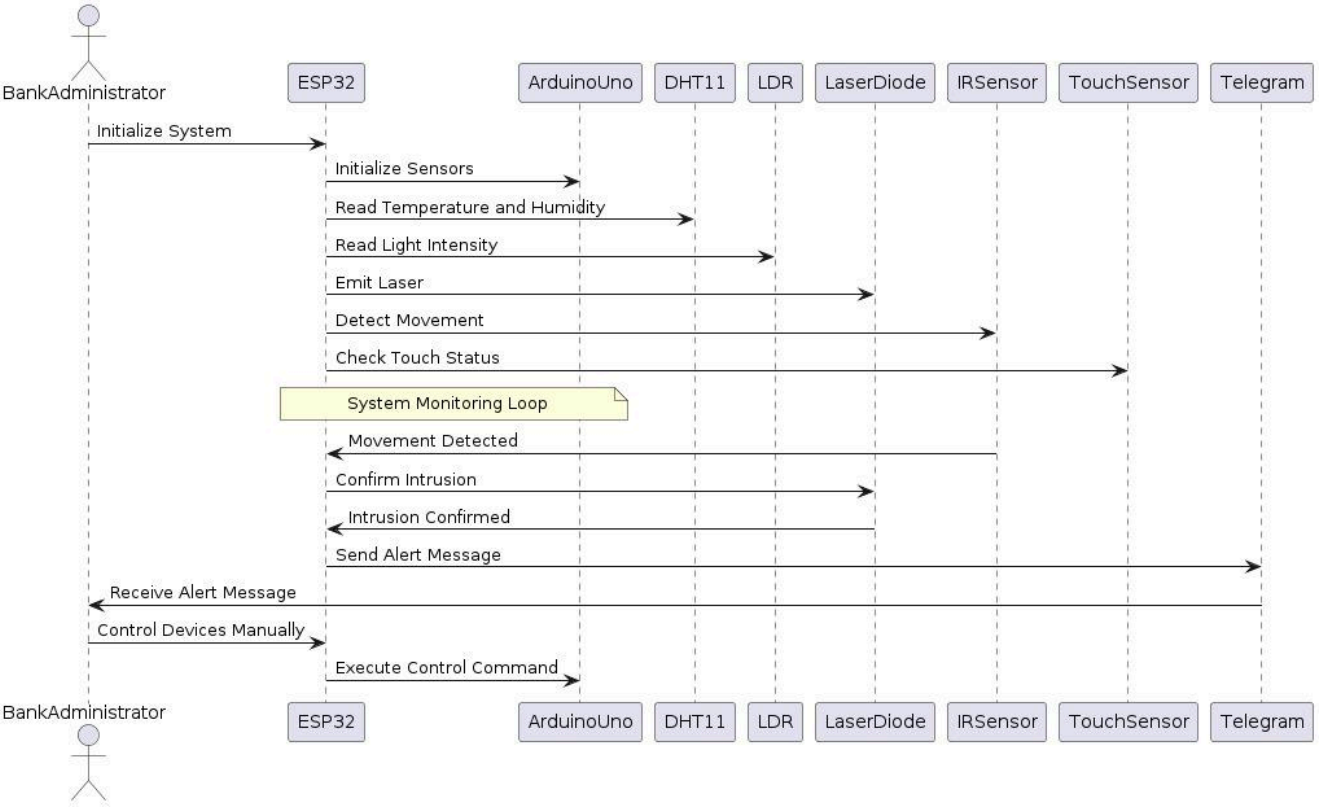


Fig 5.11.3 : Sequence of Operations

CHAPTER-6

RESULTS

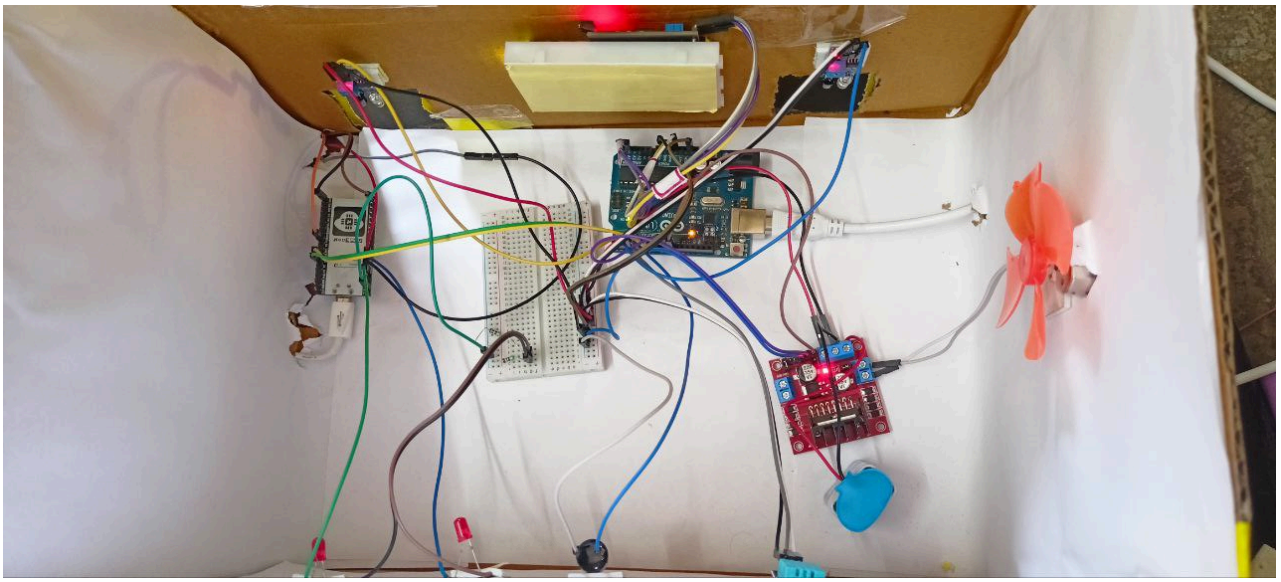


Fig.6.1: Wiring and model of the project

6.1 SUMMARY:

The successful implementation of the project, based on the provided codes, demonstrates a functional IoT-based bank security system with Telegram integration. The system incorporates various hardware components and utilizes ESP32 and Arduino Uno microcontrollers for processing and control.

6.2 KEY FEATURES:

Here's a detailed explanation of the results and working mechanism:

1. Telegram Integration:

- The system is integrated with Telegram, allowing users to control and monitor devices remotely via chat commands.
- Users can interact with the system by sending specific commands through Telegram, such as turning on/off lights, fans, lasers, and changing warning modes.

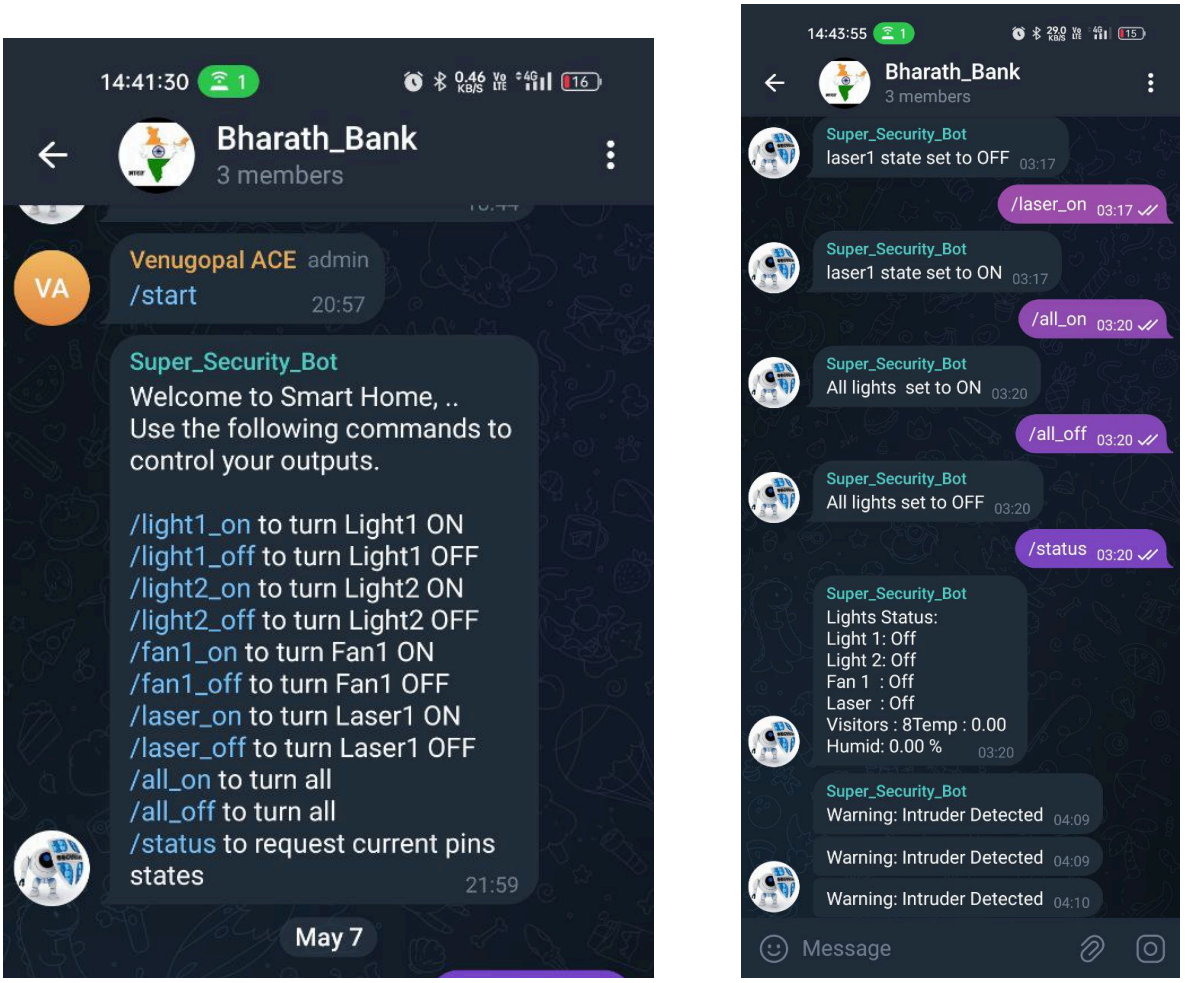


Fig. 6.2.1,6.2.2: Telegram Integration, Device Control with warning mode

2. Device Control:

The system enables control of multiple devices, including lights (Light 1 and Light 2), a fan (Fan 1), and a laser (Laser 1), through Telegram commands.

- Each device can be individually controlled, providing flexibility and customization options for users.

3. Warning Mode:

Users can switch between different warning modes using Telegram commands, altering the system's behavior in response to security threats or intruder detection.

4. Real-time Status Updates:

Users can request real-time status updates of all connected devices through Telegram.

- The system responds with the current state of each device, providing users with immediate feedback on device status.

5. Intruder Detection:

The system includes functionality for detecting intruders using an LDR sensor.

- When an intruder is detected, the system sends a warning message to Telegram, alerting users to potential security breaches.

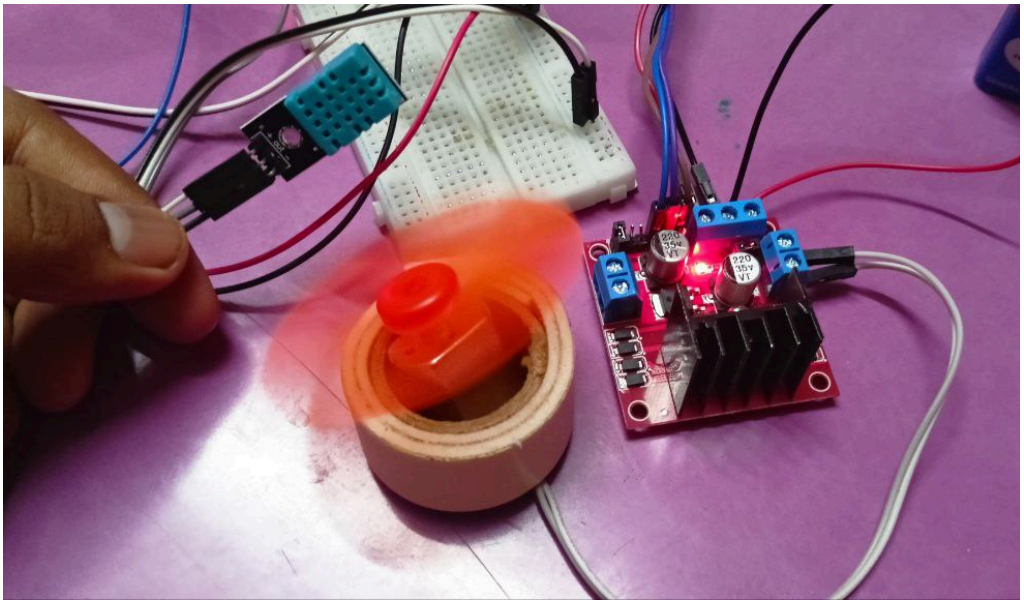


Fig.6.2.6 :Temperature based Automatic Fan Speed Control

6. Temperature and Humidity Monitoring:

The project includes provisions for temperature and humidity monitoring using a DHT11 sensor.

- However, the code snippet for this feature appears incomplete or missing, and further integration and testing may be required.

7. LCD Display:

An LCD display connected to the Arduino Uno provides real-time updates on the number of visitors detected by IR sensors.

- While the code snippet for LCD functionality is provided, detailed functionality and integration with the rest of the system are not explicitly mentioned.

8. Integration with ESP32:

The system utilizes an ESP32 microcontroller for Wi-Fi connectivity and Telegram communication, enhancing the system's remote control and monitoring capabilities.

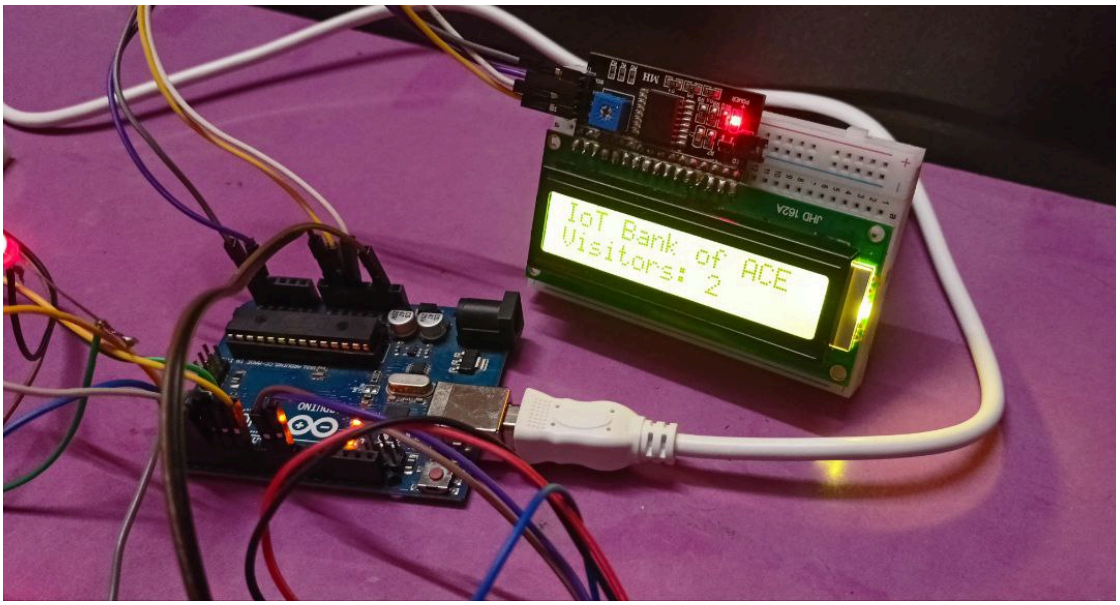


Fig.6.2.9 : Visitor count displaying on LCD

9. Visitor Counting:

The system incorporates two IR sensors for visitor counting, one for detecting incoming visitors (IR_IN) and the other for outgoing visitors (IR_OUT).

- Each time a visitor passes through the sensors, the respective count is incremented or decremented accordingly.
- The visitor count is displayed in real-time on an LCD connected to the Arduino Uno, providing visibility into the number of people present in the monitored area.

10. Automatic Fan Speed Control Based on Temperature:

The project includes provisions for automatic fan speed control based on temperature readings from a DHT11 sensor.

- The fan speed adjusts dynamically based on the ambient temperature to maintain a comfortable environment.
- Thresholds for temperature ranges (e.g., below 30°C, between 30°C and 45°C, above 45°C) determine the fan speed settings, ensuring energy efficiency and optimal comfort levels.

11.LASER SECURITY:

A laser diode and an LDR sensor form a perimeter security system, detecting intrusions by monitoring interruptions in the laser beam.

6.3 INTEGRATION:

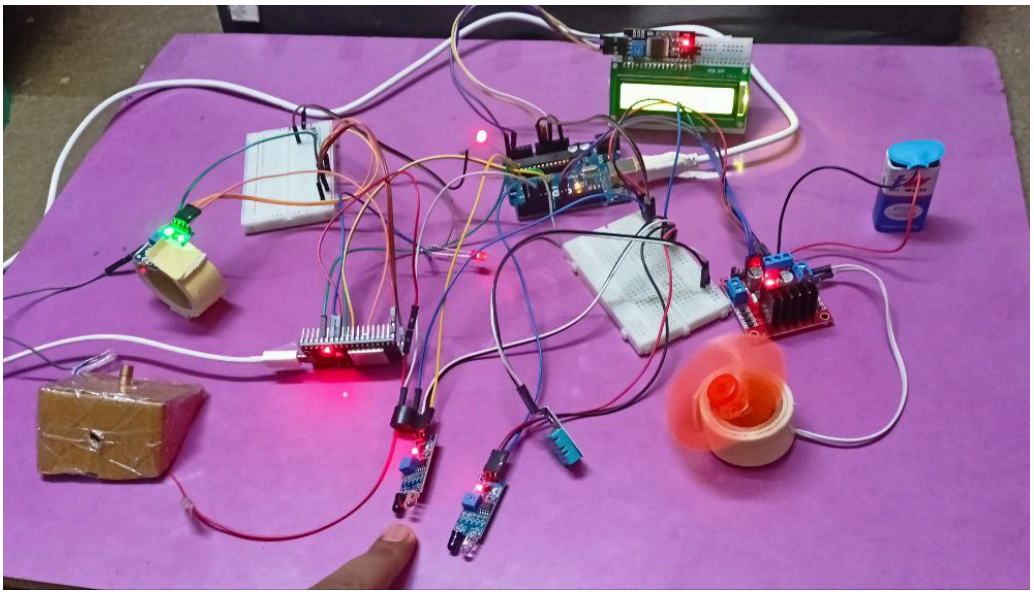


Fig. 6.3: Integration of Components and working

The hardware components are interconnected on a breadboard, facilitating easy prototyping and testing. The software components include Arduino and ESP32 code for controlling sensors and actuators, as well as a Telegram bot for communication and remote control.

6.4 CONCLUSION:

The IoT-Based Bank Security system offers a comprehensive solution for safeguarding bank premises, combining visitor counting, environmental monitoring, and intrusion detection functionalities. By leveraging IoT technology and integrating multiple sensors and actuators, the system provides enhanced security, automation, and user interaction capabilities, contributing to a safer and more efficient banking environment.

CHAPTER-7

ADVANTAGES, DISADVANTAGES, APPLICATIONS

7.1 ADVANTAGES:

1. Enhanced Security: The project provides a robust security system by incorporating features such as visitor counting, laser security, and remote monitoring via Telegram, helping to prevent unauthorized access and intrusions.
2. Automation: Automatic fan speed control based on temperature ensures optimal environmental conditions within the bank premises, enhancing comfort and energy efficiency.
3. Real-time Monitoring: Users can receive instant alerts and notifications about security events via Telegram, enabling timely responses and interventions.
4. User Interaction: The Telegram interface allows users to control lights, fans, and other components remotely, offering convenience and flexibility in managing the security system.
5. Scalability: The modular design of the system allows for easy expansion and integration of additional sensors or functionalities as per specific requirements.

7.2 DISADVANTAGES:

6. Dependency on Internet Connectivity: The system relies on an internet connection for Telegram communication, which could be a limitation in areas with poor network coverage or during network outages.
7. Complexity: Integrating multiple hardware components and writing code for various functionalities may require technical expertise, posing a challenge for beginners or non-technical users.
8. Power Consumption: Continuous operation of sensors, actuators, and communication modules may lead to increased power consumption, requiring consideration of power management strategies.

7.3 APPLICATIONS:

1. Bank Security: The project is specifically designed for enhancing security in bank premises, but it can also be adapted for use in other commercial or residential environments requiring surveillance and access control.
2. Smart Buildings: The system's automation and remote monitoring capabilities make it suitable for deployment in smart buildings, where it can manage environmental parameters and security systems intelligently.
3. Retail Stores: Retail establishments can utilize the project for monitoring customer traffic, controlling lighting and HVAC systems, and implementing security measures to prevent theft or vandalism.
4. Office Spaces: Offices can benefit from the project's security features, including visitor counting and intrusion detection, as well as its ability to optimize workspace conditions for employee comfort and productivity.
5. Residential Security: Homeowners can deploy the system to enhance the security of their residences, monitor surroundings, and receive alerts about potential security breaches or environmental anomalies.

CHAPTER-8

8.1 FUTURE SCOPE:

1. **Enhanced Security Features:** Future iterations of the project could incorporate advanced security features such as facial recognition, biometric authentication, or advanced intrusion detection algorithms to further enhance security measures.
2. **Integration with Smart Systems:** The project can be integrated with existing smart building management systems to provide a comprehensive solution for managing security, environmental control, and energy efficiency.
3. **Mobile Application Development:** Developing a dedicated mobile application alongside the Telegram interface could offer additional flexibility and functionality for users to monitor and control the security system from their smartphones.
4. **Machine Learning Algorithms:** Implementing machine learning algorithms for data analysis could enable the system to learn and adapt to changing patterns, improving its ability to detect anomalies and security threats.
5. **Scalability and Modularity:** Designing the system with scalability and modularity in mind would allow for easier expansion and integration of additional sensors, actuators, or functionalities as per evolving requirements.
6. **Cloud Integration:** Integrating the system with cloud platforms for data storage, processing, and analytics could enable advanced features such as historical data analysis, predictive maintenance, and remote software updates.

Overall, the IoT-based bank security system presents a solid foundation for future enhancements and innovations, catering to the evolving needs of security, automation, and remote management in various domains.

8.2 REFERENCES:

Here are a few references that are helpful for information on project :

8.2.1 Research Papers:

- Li, X., & Yan, J. (2018). Internet of Things (IoT) Security: Current Status, Challenges and

Prospective Measures. In 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 778-782). IEEE. DOI: 10.1109/IMCEC.2018.00174

8.2.2 Books:

- Mazidi, M. A., & Naimi, S. (2015). The 8051 Microcontroller and Embedded Systems Using Assembly and C (2nd ed.). Prentice Hall. ISBN-13: 978-0135080443
- Patel, H. B., Shah, S., & Patel, B. (2018). Internet of Things (IoT): Applications, Challenges and Future Scope. In 2018 3rd International Conference on Communication and Electronics Systems (ICCES) (pp. 382-386). IEEE. DOI: 10.1109/CESYS.2018.8610595

8.2.3 Blogs:

- PyroElectro. (n.d.). Using LDRs to Detect Light. Retrieved from http://www.pyroelectro.com/tutorials/using_ldrs/
- Last Minute Engineers. (n.d.). L298N Dual H-Bridge Motor Controller. Retrieved from https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

8.2.4 Websites:

- Arduino. (n.d.). Arduino - Home. Retrieved from https://www.arduino.cc/
- Digi-Key Electronics. (n.d.). The Basics of Using an L298N Dual H-Bridge Motor Controller. Retrieved from https://www.digikey.com/en/maker/blogs/2020/the-basics-of-using-an-l298n-dual-h-bridge-motor-controller
- Adafruit Industries. (n.d.). DHT11 Temperature & Humidity Sensor Tutorial. Retrieved from https://learn.adafruit.com/dht