Hainan Normal University
Undergraduate thesis

Thesis Title:

Intelligent home system design and Implementation

NAME: KHADKA PRATIK (符号)

Student ID: 202009B089

Major: Automation

Grade: 2020

College of Physics and Electronic Engineering

Tutor: Wang YingPin（王映品）

Completion: April 2024

Undergraduate thesis originality statement

I declare that the graduation thesis submitted by me is my research work and achievements under the guidance of my tutor. Except for the places specially marked and acknowledged in the article, there are no behaviors such as copying other's research achievements and falsifying data in this thesis. The comrades who work with me have made a clear explanation and expressed their gratitude for any contribution to this research.

Author's signature：____________________ Date：

Undergraduate thesis Authorization statement

Hainan Normal University has the right to retain and send copies and disks of graduation thesis to relevant departments or institutions of the state and allow graduation thesis to be consulted and borrowed. I authorize Hainan Normal University to incorporate all or part of the contents of this graduation thesis into the relevant database for retrieval, and to save and compile the graduation thesis by photocopying, shrinking or other means of copying. The use authorization statement of Undergraduate thesis

Author's signature：____________________ Date：

Tutor's signature：____________________ Date：

Intelligent home system design and Implementation

Author: Khadka Pratik Tutor: Wang YingPin

(Department of Automation, School of Physics and Electronic Engineering,

Hainan Normal University, Haikou 571158)

Abstract: The integration of smart devices and automation systems has revolutionized residential environments, offering unprecedented levels of convenience, comfort, security, and safety. This thesis presents a comprehensive study on the design and implementation of a home automation system aimed at enhancing various aspects within the residential setting. The project is divided into four main sections: Weather analysis and control, Security systems, Hazard detection, and control design. For this project, Arduino Mega 2560 MCU is used as a core controller.

In the weather analysis and control section, we employ photo diodes, DHT11 sensors, and rain sensors to collect environmental data, including ambient light levels, temperature, humidity, and precipitation. Intelligent algorithms automate lighting, fan operation, and window closure to enhance energy efficiency and occupant comfort based on changing weather conditions.

The security systems section focuses on enhancing home security through the implementation of door locking systems, outdoor surveillance, and hazard detection. Utilizing the AS608 Optical Fingerprint Sensor, the door locking system ensures secure access control with advanced authentication mechanisms. Outdoor surveillance is facilitated by the ESP32-CAM module, integrating surveillance cameras with motion detection algorithms to monitor exterior spaces of the residence. Homeowners receive real-time alerts and notifications in case of suspicious activity. Moreover, hazard detection systems, employing the MQ135 Gas Detection Sensor detects potential threats such as smoke, fire, or gas leaks, prioritizing the safety of occupants.

In the control design section, we focus on designing and implementing control mechanisms to seamlessly integrate and operate various home automation features. We develop user-friendly interfaces, including remote and voice control, to provide intuitive control and monitoring capabilities to homeowners. This includes the utilization of IR for remote control functionality, enabling users to conveniently manage home automation tasks from a distance. Additionally, we incorporate the DF Robot Gravity: offline language learning voice recognition sensor for voice commands, allowing users to interact with the system using natural language, enhancing accessibility and usability.

Keywords: Home Automation; Weather Analysis; Door Locking System; Outdoor Surveillance; Hazard Detection Voice Control; Remote Control; Home Safety.

Table of Contents

1. Introduction

In recent years, the integration of smart devices and automation systems has fundamentally transformed residential environments, offering unparalleled levels of convenience, comfort, security, and safety. This paradigm shift has not only revolutionized the way we interact with our living spaces but has also redefined the very notion of what it means to feel at home. Embracing this transformative trend, this thesis embarks on a comprehensive exploration of the design and implementation of a sophisticated home automation system aimed at elevating various aspects within residential settings.

The reason behind this project is simple: We believe that technology can make our lives better. Nowadays, we're all about making things more efficient, sustainable, and personalized. That's why we're diving into home automation. We want to use automation to make our homes smarter, more intuitive, and safer than ever. It's all about creating living spaces that work seamlessly with our lifestyles and keep us safe and secure.

1.1 Background and motivation for the project:

As technology continues to evolve, there is a growing emphasis on creating smarter and more efficient solutions for various aspects of life, including residential environments. This trend has led to an increasing interest in home automation systems, which offer the promise of enhanced convenience, comfort, security, and safety for homeowners.

Furthermore, with the rising concerns about energy conservation, environmental sustainability, and personalization, there is a clear need to harness the power of automation to address these challenges within residential settings. By leveraging innovative technologies and intelligent algorithms, home automation systems have the potential to optimize energy usage, adapt to changing environmental conditions, and provide personalized experiences tailored to individual preferences.

Considering these factors, the motivation behind this project is to explore the possibilities offered by home automation and develop a comprehensive solution that integrates cutting-edge technologies to enhance various aspects of residential living. By doing so, we aim to create a more comfortable, efficient, and secure environment for homeowners, ultimately improving their overall quality of life.

1.2 Objectives of the Thesis:

The primary objective of this thesis is to conceptualize, develop, and evaluate a robust home automation system that seamlessly integrates cutting-edge technologies to enhance the quality of life for residents. With this overarching goal in mind, specific objectives include:

1. Conducting a thorough analysis of weather patterns and environmental conditions to inform intelligent automation strategies.
2. Designing and implementing advanced security systems to safeguard residential properties against potential threats.
3. Developing hazard detection mechanisms to preemptively identify and mitigate risks such as smoke, fire, and gas leaks.
4. Creating intuitive control interfaces, including remote and voice control, to empower homeowners with effortless management of automation features.

1.3 Overview of the Four Sections:

The thesis is organized into four distinct sections, each focusing on a key aspect of the home automation system:

1. **Weather Analysis and Control**: This section delves into the utilization of environmental sensors and intelligent algorithms to optimize lighting, ventilation, and window management based on real-time weather data.
2. **Security Systems**: Here, the focus is on enhancing home security through the integration of sophisticated door locking systems and outdoor surveillance.
3. **Hazard Detection:** This section explores the implementation of hazard detection systems, leveraging sensors to detect and mitigate potential risks to occupants' safety.
4. **Control Design:** Finally, the thesis concludes with a discussion on the design and implementation of user-friendly control interfaces, including remote and voice control to facilitate seamless interaction with the home automation system.

By systematically addressing these areas, the thesis aims to contribute to the advancement of home automation technology while simultaneously enriching the lives of homeowners with enhanced comfort, security, and peace of mind.

1.4 Arduino Mega 2560 as MCU:

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 chip. It offers an extensive range of features and capabilities, making it suitable for a wide variety of projects, particularly those requiring a large number of I/O pins, memory, and processing power. Here's a brief overview of the specifications and features of the Arduino Mega 2560:

- **Microcontroller:** Powered by the ATmega2560 operating at 16 MHz, with 256 KB flash memory, 8 KB SRAM, and 4 KB EEPROM.
- **I/O Pins:** 54 digital I/O pins (15 PWM) and 16 analog inputs for connecting sensors, actuators, and peripherals.
- **Compatibility:** Fully compatible with Arduino IDE and ecosystem, supporting various libraries and shields for easy integration.
- **USB Interface:** Features USB interface for serial communication with computers and USB host functionality for connecting peripherals.
- **Power Supply:** Can be powered via USB or external 7-12V supply with a voltage regulator for stable operation.
- **Expansion Options:** Supports additional shields and modules for added functionality, with SPI, I2C, and UART communication protocols.

Overall, the Arduino Mega 2560 is versatile, with extensive I/O capabilities and compatibility, making it suitable for various projects.

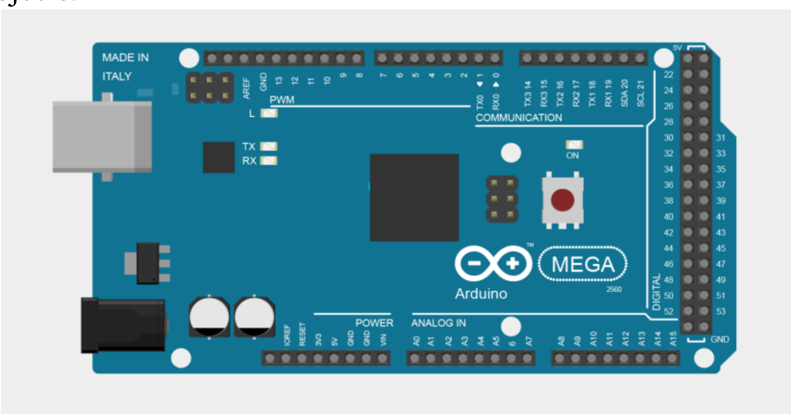


Fig 1.1: Arduino Mega 2560 R3 board

1.4.1 Selection of Mega 2560 as MCU for Intelligent Home Design and Implementation:

Using the Arduino Mega 2560 as our microcontroller unit (MCU) for our intelligent home project offers several advantages due to its large number of digital and analog pins, increased memory, and enhanced processing power compared to other Arduino boards.

Pin Availability: The Mega 2560 has a generous number of digital and analog pins, allowing us to connect multiple sensors, actuators, and other peripherals without running out of pins. This is essential for a complex project like home automation, where we may have numerous components to control and monitor.

Memory Capacity: The Mega 2560 features 256KB of flash memory and 8KB of SRAM, providing ample space for storing our program code and handling data processing tasks. This is especially beneficial when implementing sophisticated algorithms or managing large datasets, such as storing historical sensor data or maintaining user authentication records.

Processing Power: With its 16MHz clock speed and 8-bit AVR microcontroller architecture, the Mega 2560 can handle multiple tasks simultaneously, making it suitable for multitasking applications like home automation. We can use interrupts, timers, and hardware serial ports to efficiently manage various tasks and ensure smooth operation of our system.

Compatibility: The Mega 2560 is fully compatible with the Arduino ecosystem, including the Arduino IDE, libraries, and shields. This simplifies development and prototyping, as we can leverage existing code and hardware components to expedite the implementation of our intelligent home project.

Expansion Options: If we require additional functionalities or connectivity options beyond what the Mega 2560 offers natively, we can easily expand its capabilities using compatible shields or modules. The Mega 2560 comes equipped with multiple UART (Universal Asynchronous Receiver-Transmitter) ports, allowing for simultaneous serial communication with multiple devices. With these expansion options, we can enhance the connectivity and versatility of our intelligent home project to meet our specific requirements.

Reliability and Stability: The Mega 2560 is a robust and reliable MCU that is widely used in various projects and applications. Its proven track record for stability and performance makes it a dependable choice for critical systems like home automation, where reliability is paramount.

Overall, the Arduino Mega 2560 provides a versatile and powerful platform for implementing our intelligent home project, offering ample resources and flexibility to accommodate our specific requirements and scale as needed. With careful design and programming, we can leverage the capabilities of the Mega 2560 to create a sophisticated and reliable home automation system that enhances convenience, efficiency, and safety in our living environment.

2. Weather Analysis and Control

Weather analysis and control refer to the process of collecting, analyzing, and utilizing weather data to make informed decisions and adjustments within a residential environment. This involves integrating various sensors and technologies to monitor environmental conditions such as temperature, humidity, sunlight levels, and precipitation.

2.1 Introduction to Weather Analysis:

Weather analysis plays a crucial role in the field of home automation, serving as the foundation for making intelligent decisions and adjustments within residential environments. This section provides an overview of the significance of weather analysis and its integration into home automation systems.

In a weather analysis and control system, sensors like photo resistor, temperature, and humidity sensor (DHT11), and rain sensor are strategically deployed to gather real-time data on weather conditions both inside and outside the home. These sensors continuously monitor environmental variables and provide valuable insights into the current weather patterns.

Intelligent algorithms are then employed to analyze the collected data and determine the appropriate actions to optimize energy efficiency, enhance occupant comfort, and ensure safety and security. For example, based on weather forecasts and sensor readings, the system can automatically adjust heating and cooling systems, regulate lighting levels, and control window operations to maintain optimal indoor conditions.

2.1.1 Importance of Weather Analysis in Home Automation:

Weather conditions have a direct impact on various aspects of our daily lives, including indoor comfort, energy consumption, and safety. By analyzing weather patterns and environmental data, home automation systems can adapt and respond dynamically to optimize energy efficiency, enhance occupant comfort, and mitigate potential risks.

- **Energy Efficiency:** Weather analysis allows home automation systems to anticipate changes in temperature, humidity, and sunlight levels, enabling them to adjust heating, cooling, and lighting systems accordingly. By optimizing energy usage based on weather forecasts, homeowners can reduce their utility bills and minimize their environmental footprint.
- **Occupant Comfort:** By monitoring weather conditions, home automation systems can create comfortable indoor environments tailored to occupants' preferences. For example, during hot weather, the system can automatically activate fans or air conditioning to maintain a comfortable temperature indoors. Similarly, on cold days, the system can close windows to prevent heat loss.
- **Safety and Security:** Weather analysis also plays a role in enhancing home safety and security. For instance, the system can detect inclement weather conditions such as storms or heavy rainfall and take proactive measures to protect the home, such as closing windows and activating security features.

Overall, weather analysis serves as a valuable tool in home automation, enabling systems to adapt and respond intelligently to changing environmental conditions. By integrating weather data into automation processes, homeowners can enjoy greater comfort, energy savings, and peace of mind in their living spaces.

2.2 System Design:

The Weather Analysis and Control system functions by continuously gathering data from a network of sensors strategically positioned both indoors and outdoors. These sensors are designed to measure key environmental parameters such as temperature, humidity, light intensity, and precipitation. Once the data is collected, it is processed by the system's control logic. Based on the analyzed data, the system triggers corresponding actions through its output components.

Description of the components used in this section:

Table 2.1: Lists of components used for Weather analysis and Control System

INPUT	OUTPUT
LIGHT INTENSITY	
• LDR MODULE	• LED (WS2812B RGB)
HUMIDITY AND TEMPERATURE	
• DHT11	• FAN with L9110 motor driver
PRECIPATION	
• YL-83 Rain Sensor with control board	• Servo SG90 (attached to window) • 1602 LCD I2C

2.2.1 LDR MODULE:

Light dependent Resistor (LDR), also known as a photoresistor or photocell, is a type of resistor whose resistance changes in response to the intensity of light incident upon it. When light interacts with this material, it absorbs the radiation, causing electrons to transition from the valence band to the conduction band within the semiconductor.

Most photo resistive light sensors utilize **cadmium sulfide (CdS)** due to its spectral response resembling that of the human eye and its mouldability with a handheld torch. CdS has a peak sensitivity between 560-600nm, within the visible spectrum. The ORP12 CdS photoconductive cell is widely used, with a spectral response around 610 nm.

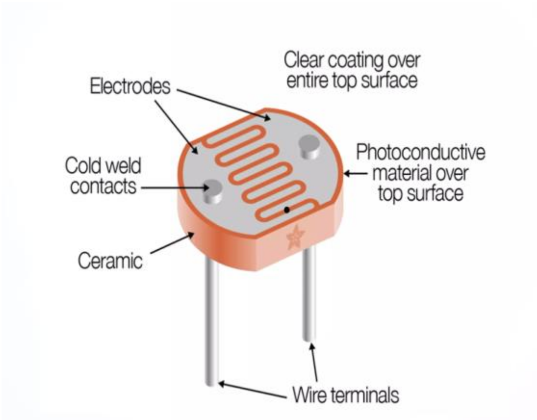


Fig 2.1: LDR Sensor

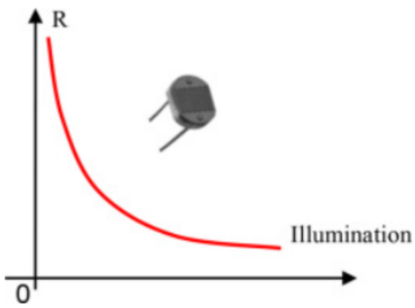


Fig 2.2: Graph showing the relation between Illumination and resistance.

In darkness, its resistance is high at around 10MΩ, dropping to about 100Ω when illuminated.

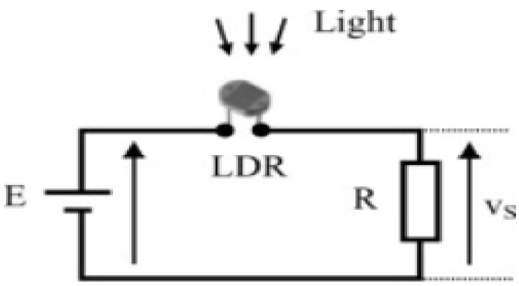


Fig 2.3: Basic Circuit with LDR and Resistor R

As LDR and R are connected in series; Using Voltage Divider rule we have;

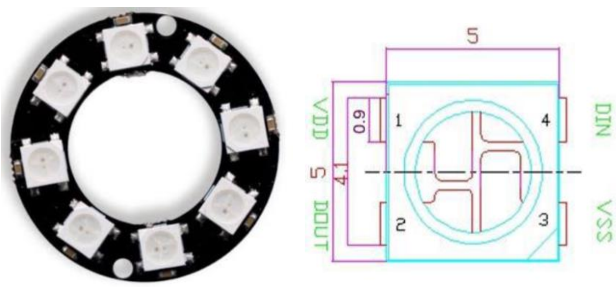
$$Vs = \frac{R}{R + R_{LDR}} E \tag{2.1}$$

Case 1: When $R << R_{LDR}; V_s \cong 0$

Case 2: When $R >> R_{LDR}; V_s \cong E$

2.2.2 LED (WS2812B RGB):

The WS2812B is an intelligent LED light source where both the control circuit and RGB chip are integrated into a compact 5050 component package. It includes an intelligent digital port for data latch and a circuit for reshaping and amplifying signals. Additionally, it features a precise internal oscillator and programmable constant current control to ensure consistent color brightness across pixels.



Symbol	Function
VDD	Power Supply
DOUT	Control data signal output
VSS	Ground
DIN	Control data signal input

Fig 2.4: WS2812B*8LEDs with its pins on each LED.

RGB IC characteristic parameter:

Table 2.2: RBC IC characteristics parameter

Emitting Color	Wavelength(nm)	Luminous intensity(mcd)	Voltage(V)
RED	620-625	390-420	2.0-2.2
GREEN	522-525	660-720	3.0-3.4
BLUE	465-467	180-200	3.0-3.4

This LED utilizes a single NZR communication mode for data transfer. Upon power-on reset, the first pixel collects initial 24-bit data from the controller and passes it to the internal data latch. Subsequent data, reshaped by the internal circuit, is transmitted to the next cascade pixel through the DO port. Each pixel's signal reduces by 24 bits during transmission. The pixel employs automatic reshaping transmit technology, allowing for cascading without signal transmission limitations, dependent solely on the speed of signal transmission.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

The data of D1 is send by MCU, and D2, D3, D4 through pixel internal reshaping amplification to transmit. We need to follow this order of GRB to send data and the high bits are sent at first.

2.2.3 DHT11 Humidity and Temperature Sensor:

The DHT11 (Digital Humidity Temperature) sensor is a popular and inexpensive digital temperature and humidity sensor. The DHT11 sensor integrates a sensing element composed of a thermistor for temperature measurement and a humidity-sensitive capacitor. These elements produce analog signals, which are converted into digital format by an onboard analog-to-digital converter (ADC). A microcontroller or dedicated integrated circuit (IC) manages signal processing, data conversion, and communication with external devices. Using a simple digital communication protocol, the sensor transmits temperature and humidity data via a single data line. Encapsulated within a protective casing, the sensor is shielded from environmental factors, ensuring reliable operation. Additional components such as resistors, capacitors, and voltage regulators stabilize output signals and compensate for operating variations, ensuring accurate and stable measurements.

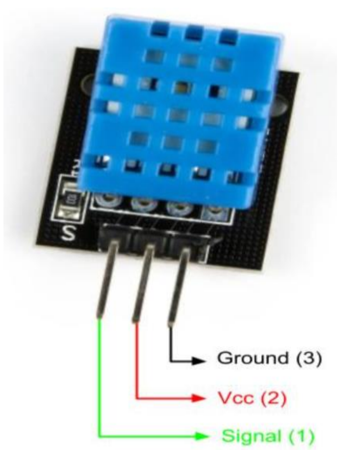


Fig 2.5: DHT11 Sensor with its pinouts.

Here are some key features and details about the DHT11 sensor:

- **Functionality**: The DHT11 sensor is capable of measuring both temperature and humidity. It provides digital output signals corresponding to the temperature and humidity readings.
- **Accuracy**: While the DHT11 sensor is affordable and widely available, it typically offers moderate accuracy compared to more expensive sensors. The temperature accuracy is usually within ±2°C, and the humidity accuracy is within ±5%
- **Operating Range**: The operating temperature range of the DHT11 sensor is usually between 0°C to 50°C (32°F to 122°F), and the humidity range is between 20% to 90% relative humidity.
- **Communication Protocol**: The DHT11 sensor communicates over a single-wire digital interface, making it relatively simple to integrate into microcontroller-based projects. It uses a proprietary communication protocol to transmit data to the **microcontroller.**
- **Power Requirements**: The DHT11 sensor typically operates at 3.3V or 5V DC power supply. It consumes very low power during operation, making it suitable for battery-powered applications.

2.2.4 FAN with L9110 motor driver:

For the colling system prototype a DC fan with L9110 driver module is used.

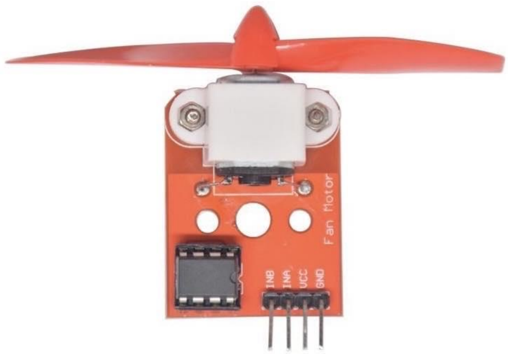


Fig 2.6: FAN with L9110 motor driver

L9110 IC consists of 4 pins: INA, INB, OUTA, OUTB, VCC and GND. VCC and GND are for the power supply; OUTA and OUTB are connected to motor; INA and INB are the input data pins which function is described below:

Table 2.3: Inputs configuration of L9110 and corresponding outputs to dc motor

INA	INB	OUTPUT	MOTOR
LOW	LOW	LOW	STOP
LOW	HIGH	HIGH	ANTI-CLOCKWISE
HIGH	LOW	HIGH	CLOCKWISE
HIGH	HIGH	LOW	STOP

2.2.5 YL-83 Rain Sensor with control board:

The YL-83 rain sensor is a simple sensor module used to detect rain or moisture. It typically consists of a circuit board with a moisture-sensing probe or plate. When rainwater or moisture comes in contact with the probe, it triggers a signal change, indicating the presence of water.

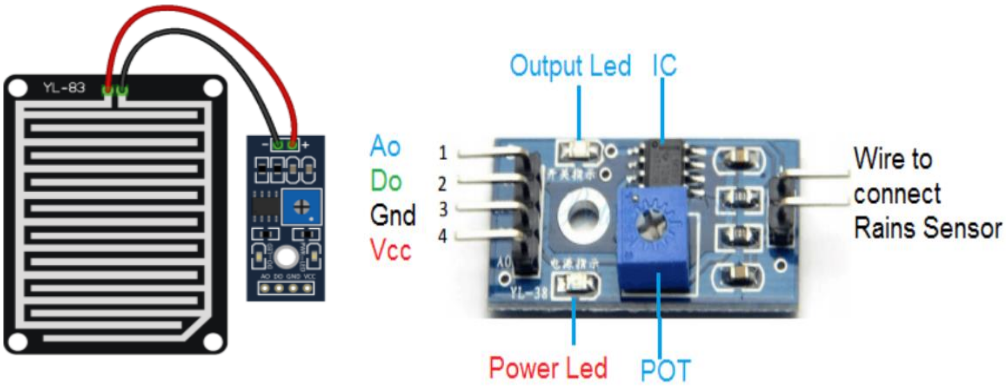


Fig 2.7: YL-83 Rain Sensor with control board configuration.

A0 represents analog input pin and **D0** represents digital. The analog output of the sensor detects changes in rainfall levels. When connected to a 5V power supply, the LED illuminates when the sensor board detects no raindrops, and the D0 output is high. When there's a slight presence of water droplets, the D0 output becomes low, triggering the indicator switch to turn on. Removing the water droplets restores the sensor to its initial state, causing the output to return to a high level.

Hence, When the sensor board is:

Table 2.4: Working principle of YL-83 Rain Sensor.

Dry	• The resistance is less, with higher output voltage.
Wet	• The resistance is more , with lower output voltage.

2.2.6 SG90 Servomotor:

Unlike the common dc motor which are generally used for continuous rotations, servo motors are used for the precise angular movement of the shaft.

Fig 2.8: SG90 Servo motor

The servo motor working principle is divided into two major sections: Gear section and electronic section.

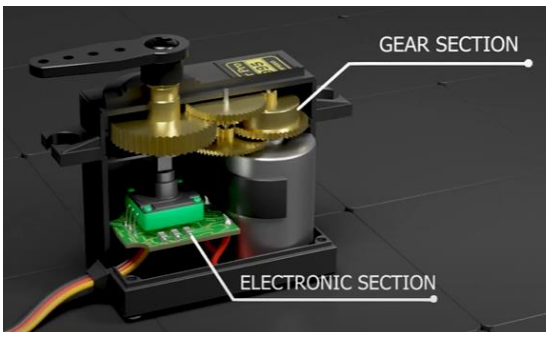


Fig 2.9: Interior of Servo motor

I. **Gear section:** This section contains the combined gear system which is intended to produce low speed high torque output. The combined system works on the gear ratio mechanism. It uses the compound gear technique. It has 260 gear ratio, which refers that the output gear rotates 1 time while the input/primary gear rotates 260 times with 1/260th times of torque than output gear.

II. **Electronic section:** Servo motor use potentiometer as a feedback sensor, potentiometer sends the signal representing the current position of the shaft. The rotation of the servo motor is controlled by the PWM signals. When the signal is suppressed via signal wire, the dc motor inside it moves and altogether with compound gear rotates the shaft to the precise angle which is then feedback to the microcontroller unit with the help of potentiometer as a feedback sensor. If the shaft reaches the expected angle, the potentiometer surpasses the signal to MCU to stop the dc motor.

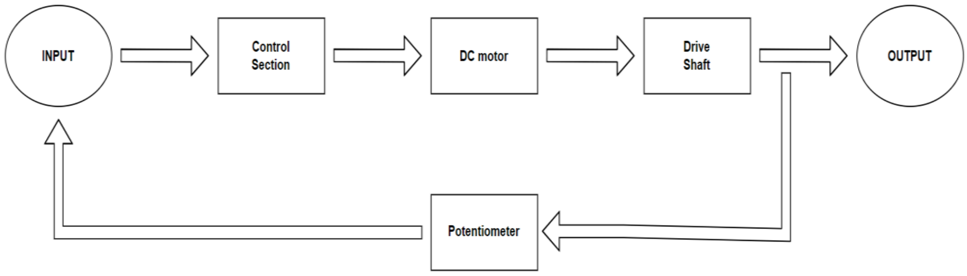


Fig 2.10: Working of Servo motor.

2.2.7 1602 LCD with I2C module:

The 1602 LCD, also known as the 16x2 LCD, is a widely used alphanumeric liquid crystal display module with 16 characters per line (2 lines). It's a popular choice for displaying text and basic graphics in various electronic projects due to its simplicity, ease of use, and low cost.

Fig 2.11: 1602 LCD connected with I2C module

Here are some key features of the 1602 LCD:

- **Display Size:** The 1602 LCD module has a display area of 16 characters per line (2 lines) , hence the name "16x2". Each character is typically a 5x8 pixel matrix.
- **Communication Interface:** Most 1602 LCD modules come with a parallel interface, which requires a significant number of I/O pins for interfacing with microcontrollers. However, there are also versions of the 1602 LCD with an **I2C interface (I2C backpack/module)** that simplifies the wiring and requires only two pins for communication.
- **Backlight:** Many 1602 LCD modules feature a built-in LED backlight, which enhances visibility in low-light conditions. The backlight can usually be controlled separately from the display itself.

- **Controller:** The 1602 LCD is typically driven by the Hitachi HD44780 or compatible controller chip. This controller simplifies the process of interfacing with the display and provides features such as clear display, cursor positioning, and scrolling.
- **Character Set:** The 1602 LCD supports a standard character set, including alphanumeric characters, symbols, and some special characters. Custom characters can also be defined and displayed on the LCD.
- **Power Supply:** The operating voltage of the 1602 LCD module is usually around 5 volts, making it compatible with most microcontrollers and development boards.

While using the 1602 LCD with an I2C module/ backpack, several key points should be considered. The I2C module, acting as an interface or backpack, simplifies communication by converting the LCD's parallel interface to an I2C interface, significantly reducing the required connections of 8 pin(GND, VDD,VO,RS,E,D0,D1,D2,D3,D4,D5,D6,D7,BLA,BLK); to just two wires (SDA and SCL). These modules typically come with a fixed I2C address, such as 0x27 or 0x3F, though adjustments can sometimes be made to accommodate multiple LCDs on the same I2C bus. Wiring is simplified, requiring only connections for SDA, SCL, VCC, and GND. Programming the 1602 LCD with I2C interface necessitates the use of compatible libraries like "LiquidCrystal_I2C" for Arduino, streamlining initialization and usage. The advantages of using an I2C module include reduced wiring complexity, conservation of GPIO pins, and simplified interfacing with other I2C-compatible devices on the same bus.

2.3 Control Logic and Automation:

In this sub section, the logic and algorithms governing the control of various home automation elements, including lighting, fan, and window. Automation strategies based on sensor readings will be explored to create an intelligent and responsive environment within the home.

The control of home automation elements relies on intelligent algorithms that process sensor readings and determine appropriate actions. For instance, the lighting system can be switched on/off based on ambient light intensity readings from Light-Dependent Resistor(LDR) sensor. Similarly, the fan operation can be controlled based on temperature and humidity data obtained from DHT11. Window operations may be governed by rain sensors, which detect precipitation and trigger automatic closing to prevent water ingress.

2.3.1 Lighting System:

As LDR (Light Dependent Resistor) sensor and LED are used as components of a lighting automation system. Here's how they work within the design:

- **LDR Sensor (Light Dependent Resistor)**:
  The LDR sensor is connected to pin **A0** of the **Arduino Mega (const int LDR = A0;).**
  It measures the intensity of light falling on it. When exposed to light, the resistance of the LDR decreases, and when in darkness, its resistance increases.
  In the **void loop()** function, the light variable is assigned the value of the analog reading from the LDR sensor:
  The value of light is then used to determine whether to turn on or off the LED based on a predefined threshold:

```
light > 500 ? lighton() : lightoff();
```

  If the light intensity surpasses the threshold value of 500, the lighton() function is called, which turns on the LED. Otherwise, the lightoff() function is called to turn off the LED.

- **WS2812B LED:**
  The LED is connected to pin **5** of the **Arduino Mega** (const int led = 5;).
  It serves as an indicator or a light source that can be controlled based on the input from the LDR sensor.
  Two functions**, lighton()** and **lightoff()**, are defined to control the LED:

```
void lighton()
{
  LED.fill(LED.Color(255,255,255));
  LED.show();
}
void lightoff()
{
  LED.fill(LED.Color(0,0,0));
  LED.show();
}
```

The lighton() function fills the LED strip with white color (**RGB values of 255, 255, 255**) to turn it on, while the lightoff() function turns off the LED by filling it with no color (**RGB values of 0, 0, 0**). The **LED.show()** function is called after setting the LED color to actually display the changes made.

Overall, the LDR sensor detects ambient light levels, and based on its readings, the LED is either turned on or off to provide illumination or indicate certain conditions within the home automation system.

The overall operation is provided by the given flowchart :

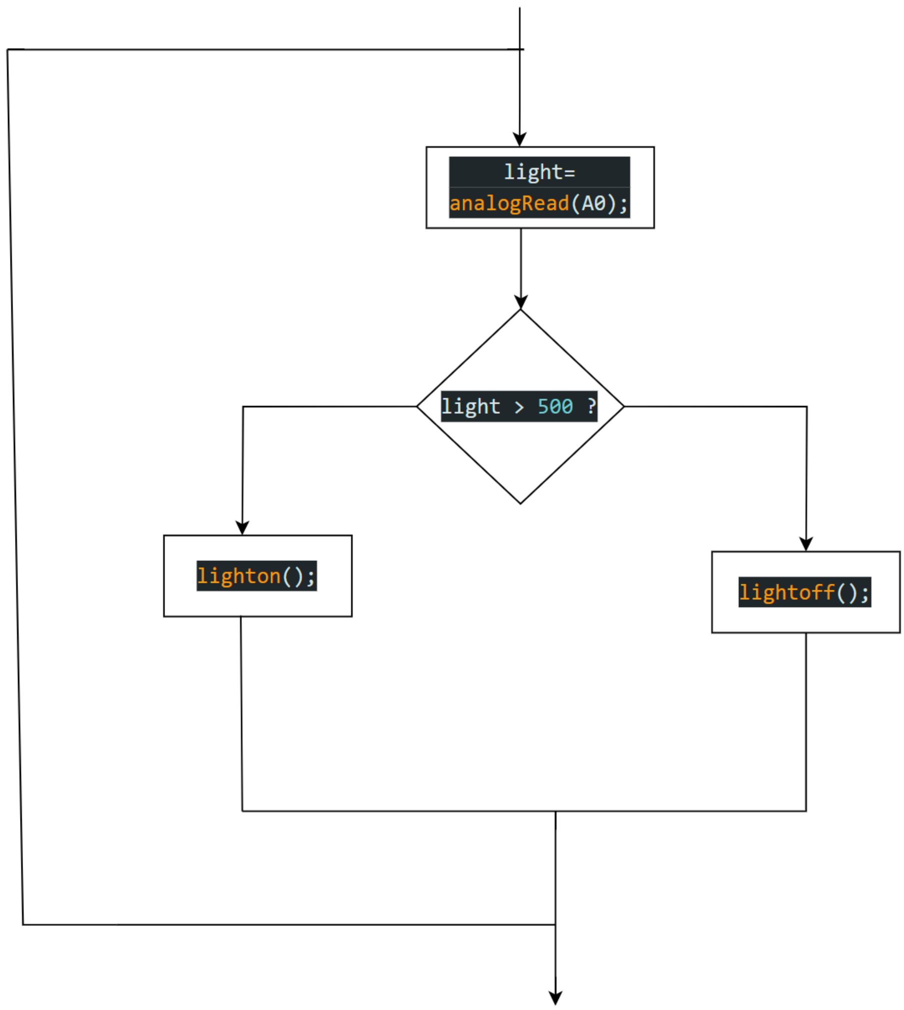


Fig 2.12: Operation of lighting system.

Fig 2.13: Function of WS2812B LED in response to light intensity from LDR.

2.3.2 Window Operation System:

As YL-83 Rain Sensor rain sensor and Servo motor are used as components of automatic window operation system. Here's how they work within the design:

- **YL-83 Rain Sensor:**

  The rain sensor is connected to pin **A1** of the Arduino Mega (**const int rain = A1;**). It measures the presence or absence of precipitation (rain). In the **void loop()** function, the precipitation` variable is assigned the value of the analog reading from the rain sensor: **precipitation = analogRead(rain);**.
  The value of precipitation is then used to determine whether to open or close the window based on a predefined threshold:

```
precipation > 600 ? windowopen() : windowclose();
```

  If the precipitation level surpasses the threshold value of 600, the **windowopen()** function is called to open the window. Otherwise, the **windowclose()** function is called to close the window.

- **Servo Motor for Window:**

  A Servo motor is used to control the movement of the window. The Servo motor is connected to the `window` pin of the Arduino Mega (**const int window = 7;**). Two angles are defined to control the window's open and close positions: **windowopenangle= 175;** and **windowcloseangle= 10.** Two functions, **windowopen()** and **windowclose()**, are defined to control the Servo motor and hence the window's movement:

```
void windowopen()
{
  Window.write(windowopenangle); // Rotate servo to open angle
}
void windowclose()
{

  Window.write(windowcloseangle); // Rotate servo to close angle
}
```

  The windowopen() function rotates the Servo motor to the windowopenangle, which corresponds to opening the window. Similarly, the windowclose() function rotates the Servo motor to the windowcloseangle to close the window.

Overall, the rain sensor detects the presence of precipitation, and based on its readings, the Servo motor controlling the window is either rotated to open or close the window, providing an automated system.

The operation of window with respect to precipitation is provided with the flowchart:

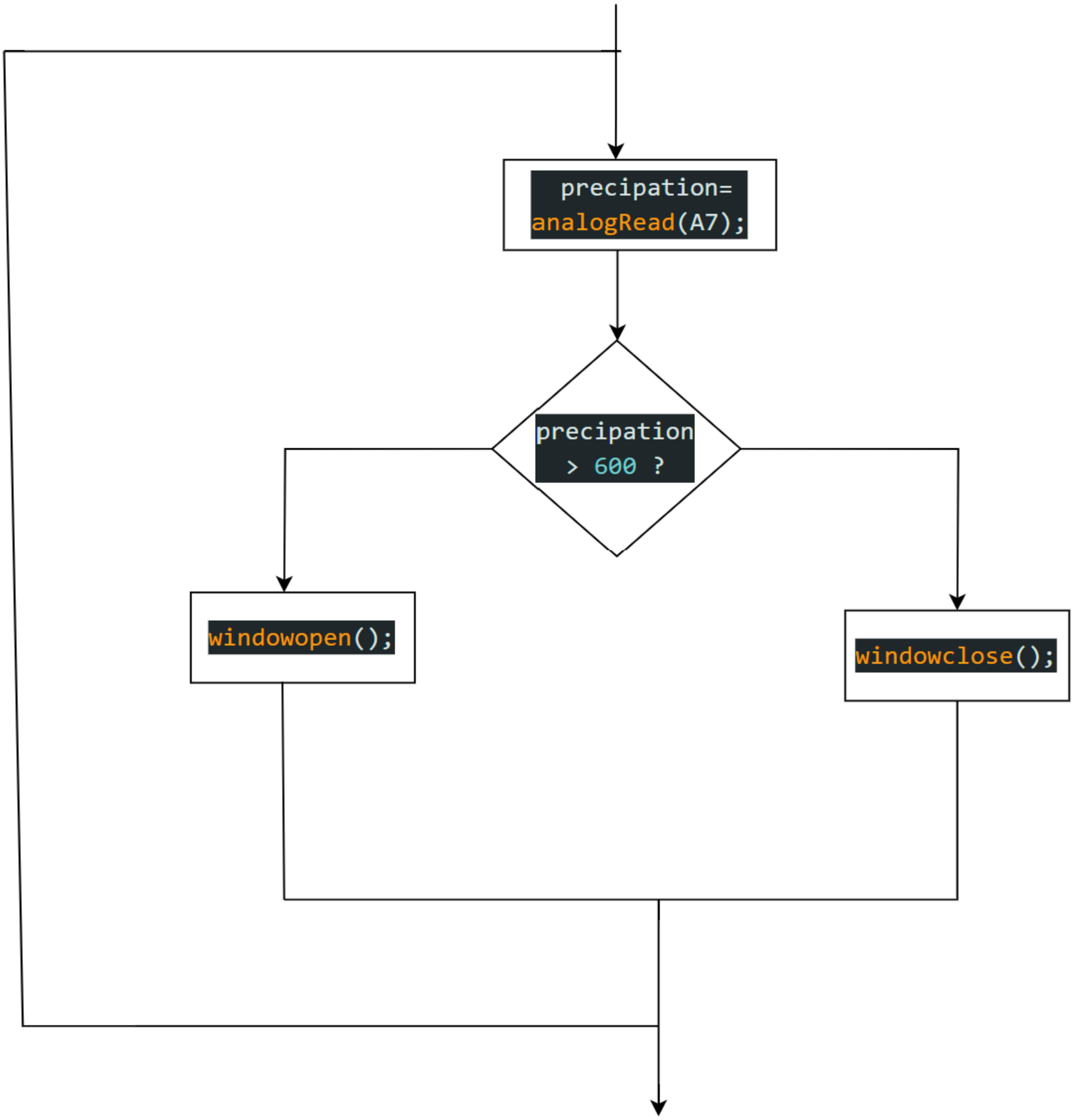


Fig 2.14: Operation of automatic window opening/closing system.

Fig 2.15: Function of Servo for window in response to precipitation from YL-83 sensor

2.3.3 FAN with L9110 motor driver as Cooling system:

As DHT11 humidity and temperature sensor and Fan with L9110 motor driver is used as components of automatic cooling system. Here's how they work within the design:

- **DHT11 Sensor:**

  The DHT11 sensor is connected to pin dht=8. The readTemperature() and readHumidity() functions are used to retrieve the temperature and humidity readings respectively. These readings are then stored in variables tempC and humid.

```
tempC > 28 ? fanon() : fanoff();
```

  If the temperature (in degree centigrade) reading of DHT11 is more than 28 than the fanon() function is called to environment otherwise fanoff() function to keep the fan turned on.

- **FAN with L9110 motor driver:**

  The fan is typically a DC motor, and the L9110 motor driver is a dual H-bridge motor driver designed to control the direction and speed of DC motors. Two Inputs pins of this driver; IN1 and IN2 are connected to the digital pins Digital pins D9 and D10 respectively. Here's how the functions work:
  ***fanon()** function: This function is responsible for turning the fan on. It achieves this by setting the appropriate pins connected to the motor driver. In this case, it sets IN1 to HIGH and IN2 to LOW. This configuration is commonly used to rotate the motor in one direction, either clockwise or counterclockwise, depending on the specific wiring of the motor and motor driver. In the context of the L9110 motor driver, setting IN1 high and IN2 low usually corresponds to rotating the motor in one direction.
  ***fanoff()** function: This function is responsible for turning the fan off. It achieves this by setting both IN1 and IN2 pins to LOW. This stops the motor from spinning by effectively cutting off power to both sides of the motor. It's a standard way to stop the motor when it's no longer needed.

Overall, if the temperature exceeds 28 degrees Celsius, the fan is turned on to help cool the environment. Otherwise, if the temperature is 28 degrees Celsius or lower, the fan is turned off to conserve energy. This mechanism helps regulate the temperature within the desired range by automatically controlling the fan based on environmental conditions.

The operation of FAN with respect to Temperature from DHT11 sensor is provided with the flowchart:

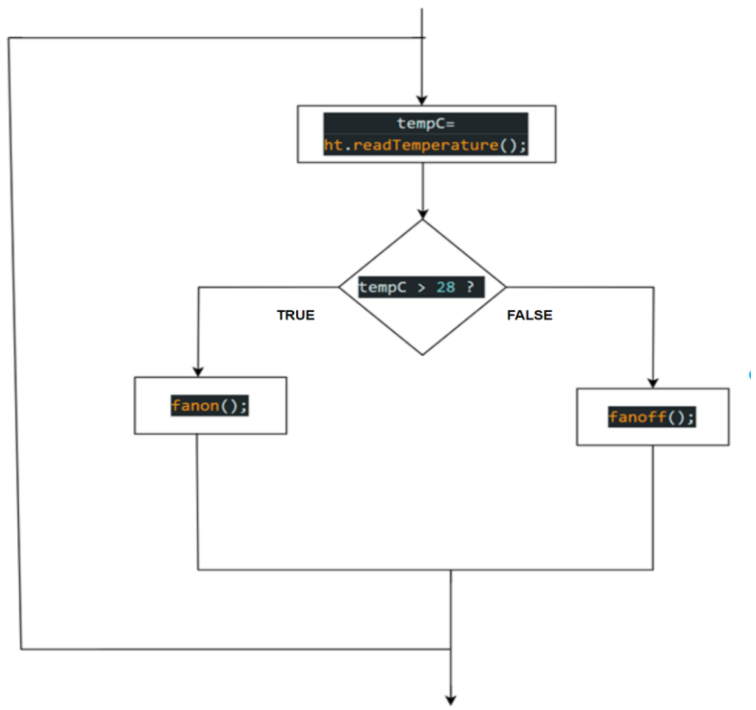


Fig 2.16: Operation of automatic cooling system with respect to temperature.

Fig 2.17: Function of Fan for colling in response to temperature reading from DHT11 sensor.

2.4 Implementation:

The implemented intelligent home system incorporates a lighting control system that adjusts indoor lighting based on ambient light levels detected by the LDR, ensuring optimal illumination while conserving energy. Precipitation detection is facilitated by a rain sensor, triggering the automatic opening and closing of windows to provide ventilation during rainfall and prevent water ingress. Furthermore, a fan control system regulates the operation of a fan in response to temperature readings from the DHT11 sensor, activating the fan to provide cooling when temperatures exceed a predefined threshold and turning it off to conserve energy when temperatures are lower. Together, these systems contribute to creating a comfortable and efficient indoor environment by dynamically adjusting lighting, ventilation, and cooling according to environmental conditions such as light intensity, precipitation, and temperature, enhancing overall convenience and energy efficiency.

2.4.1 Weather analysis and control overall functionality and designed circuit.

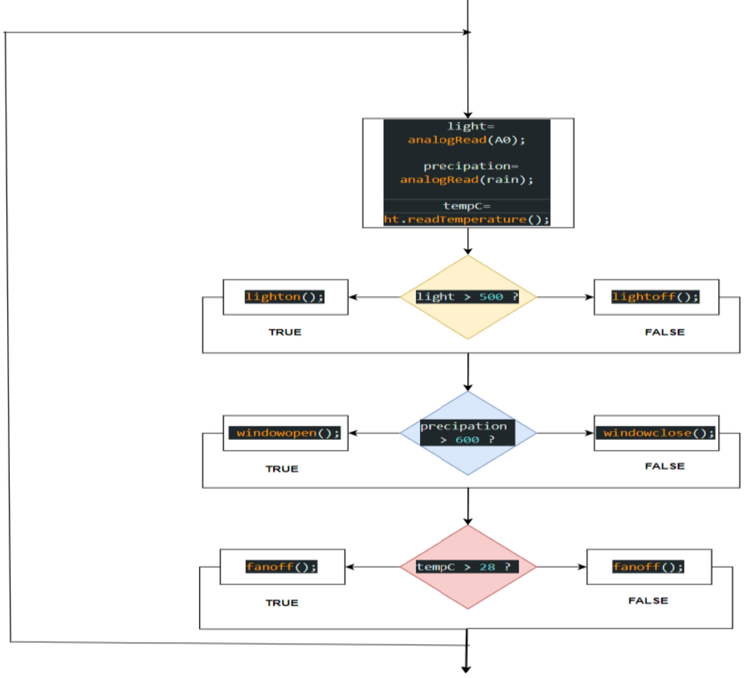


Fig 2.18: Overall working of Weather Analysis and Control.

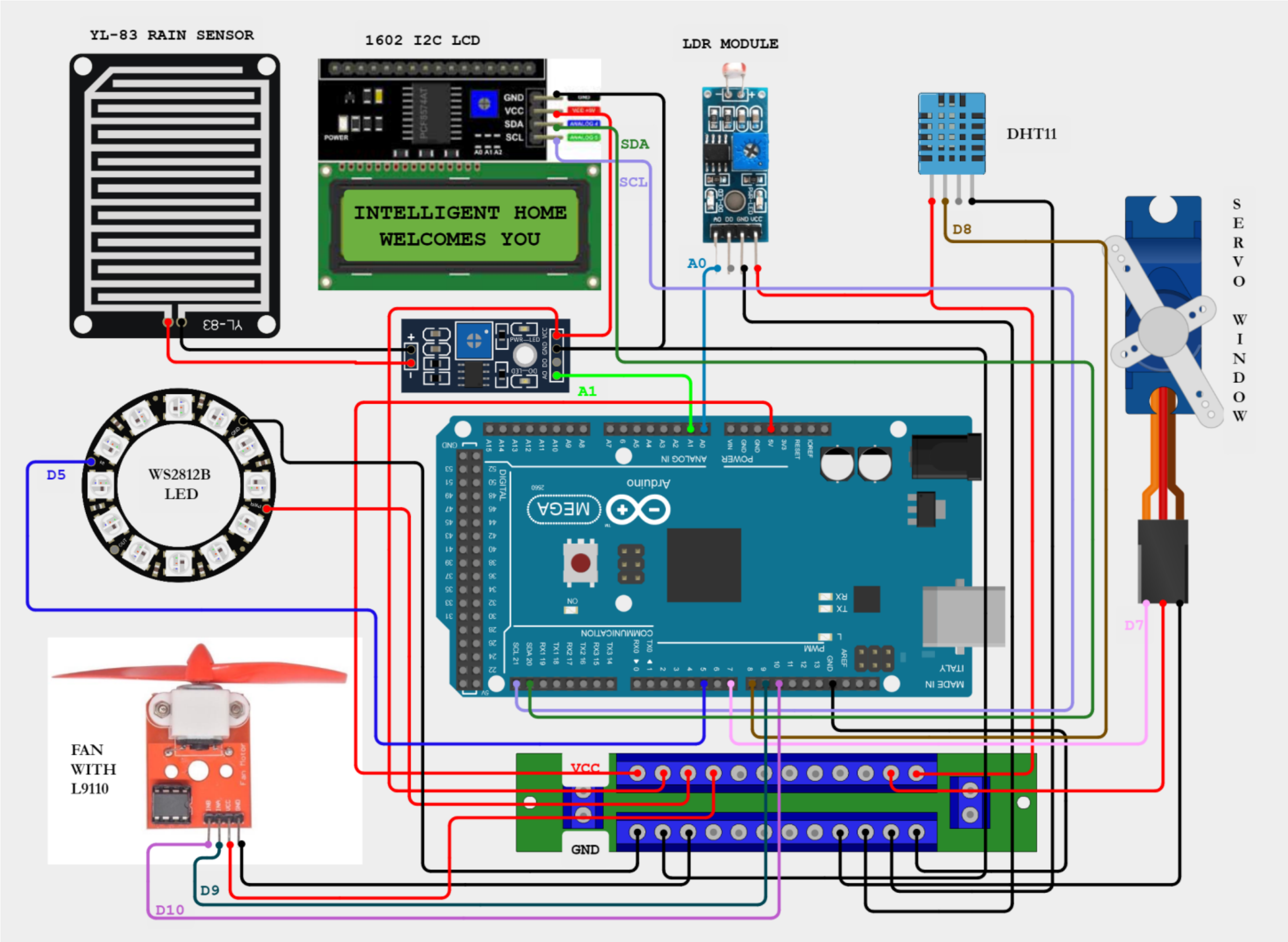


Fig 2.19: Circuit Diagram for Weather Analysis and Control.

2.5 Result and Discussion:

2.5.1 Evaluation of weather analysis and control system:

The Weather Analysis and Control demonstrates a effective level of automation and control, enabling seamless management of various household functions in response to changing environmental conditions. This system integrates sensors such as the Light Dependent Resistor (LDR) for light sensing, the rain sensor for detecting precipitation, and the DHT11 sensor for monitoring temperature and humidity levels. By harnessing data from these sensors, the system can dynamically adjust its operations to suit prevailing circumstances, enhancing convenience and comfort for occupants while promoting energy efficiency and cost-effectiveness.

The adaptability of the system is evident in its ability to regulate lighting levels based on ambient light conditions, facilitate natural ventilation by opening windows during rain, and activate the fan to maintain a comfortable indoor temperature when it exceeds a certain threshold. This intelligent management not only improves the living environment but also contributes to sustainability by curbing unnecessary energy consumption. By leveraging sensor data, the system optimizes resource usage, reducing both environmental impact and utility costs.

2.5.2 Impact on Energy Efficiency and comfort:

The integration of intelligent control mechanisms within the home system significantly impacts both energy efficiency and comfort levels. By dynamically adjusting lighting levels, natural ventilation, and temperature regulation based on environmental conditions sensed by the LDR, rain sensor, and DHT11 sensor, respectively, the system optimizes energy usage. This proactive approach ensures that lights are only utilized when necessary, windows are opened to leverage natural cooling during rain, and the fan is activated to maintain a comfortable indoor temperature. As a result, unnecessary energy consumption is minimized, leading to reduced utility bills and a lower carbon footprint.

Furthermore, the system enhances comfort levels within the home by providing automated responses to changing environmental conditions. For instance, during hot weather, the fan is activated to circulate air and maintain a pleasant indoor temperature, while the opening and closing of windows facilitate natural airflow. This not only enhances thermal comfort but also improves indoor air quality. Additionally, the system's ability to adapt to user preferences through various control methods ensures that occupants can effortlessly customize their environment to suit their comfort preferences. Overall, the integration of intelligent control mechanisms enhances energy efficiency while simultaneously optimizing comfort levels within the home.

2.5.3 Challenges and limitations:

a. **Complex Integration:** Integrating multiple sensors, actuators, and control mechanisms into a cohesive system poses a significant challenge due to the technical complexity involved. Ensuring seamless communication and compatibility among various components requires expertise in electronics and programming.
b. **Reliability and Accuracy:** The reliability and accuracy of sensor data can be affected by environmental factors such as interference, calibration issues, and sensor degradation over time. These factors may lead to potential inaccuracies in the system's decision-making process.
c. **Privacy and Security Concerns:** Intelligent home systems raise concerns regarding privacy and security. Unauthorized access, hacking, and interception of data are potential risks that need to be addressed through robust encryption protocols and authentication mechanisms.
d. **Cost of Ownership:** The initial investment required for purchasing sensors, actuators, and control devices, coupled with installation and ongoing maintenance costs, can be prohibitive for some homeowners. The complexity of the system may necessitate professional installation and technical support, further adding to the overall cost of ownership.
e. **Interoperability Issues:** Incompatibility between different smart home devices and platforms can hinder seamless integration and interoperability. Variations in protocols, standards, and communication interfaces may limit the functionality of smart home devices, requiring homeowners to select products from a single manufacturer or ecosystem to ensure compatibility.

3. Security Systems:

Ensuring home security is paramount in today's environment, where threats of burglary and unauthorized access are prevalent. The door locking system and outdoor surveillance are integral components of a comprehensive security setup aimed at safeguarding the premises and its occupants. For Door locking system **RFIDcard reader** and **AS608 fingerprint sensor** has been equipped and for outdoor surveillance **ESP 32 cam** can been installed at the exterior of residentials.

3.1 Introduction to Security Systems:

In the design and implementation of an intelligent home, integrating robust security systems is paramount for safeguarding both property and occupants. Here's how incorporating specific components enhances the security aspect:

a) Protection of Property and Assets:
Utilizing advanced surveillance systems like the ESP32 cam for outdoor surveillance ensures comprehensive monitoring of the property perimeter, deterring potential intruders and protecting valuable assets from theft or damage. Additionally, implementing secure door lock/unlocking mechanisms using RFID and fingerprint scanners, such as the RFID card reader and AS608 fingerprint sensor, adds an extra layer of security. These systems allow only authorized individuals to enter the premises, thereby safeguarding property and assets effectively.

b) Safety of Occupants:
Implementing secure door lock/unlocking mechanisms using RFID and fingerprint scanners ensures the safety of occupants by providing reliable access control. These systems offer an extra layer of security, allowing only authorized individuals to enter the premises.

c) Peace of Mind:
The integration of state-of-the-art security systems, including RFID and fingerprint scanners, provides homeowners with peace of mind. Knowing that their home is equipped with advanced technology to control access effectively brings a sense of security and reassurance. Additionally, the presence of surveillance systems like the ESP32 cam for outdoor monitoring further enhances this peace of mind. Homeowners can remotely view live camera feeds, ensuring constant vigilance over their property and allowing them to respond promptly to any security concerns. This comprehensive surveillance capability contributes significantly to the overall sense of security and tranquility in an intelligent home environment.

d) Remote Monitoring and Control:
With systems like the ESP32 cam for outdoor surveillance and RFID/fingerprint scanners for door access, homeowners can remotely monitor and control security aspects of their home. They can access live camera feeds and manage door access permissions from anywhere, enhancing security even when away from home.

e) Avoidance of Criminal Activity:
Visible security measures such as outdoor cameras and secure door lock mechanisms act as strong deterrents to potential intruders. The presence of these systems reduces the likelihood of criminal activity and break-ins, making the home a less attractive target.

Overall, integrating components such as the ESP32 cam for outdoor surveillance and RFID/fingerprint scanners for door access control enhances the overall security of an intelligent home. These systems provide comprehensive protection for both property and occupants, offering peace of mind, remote monitoring and control capabilities, deterrence against criminal activity, seamless integration with home automation, and adaptability to evolving security threats.

3.2 System Design:

The security system incorporates smart locks for door locking, equipped with advanced features like biometric authentication, keyless entry, and remote access control. These locks are complemented by sensors and authentication mechanisms to regulate access effectively. Integration with home automation platforms enables seamless management and monitoring of security devices. Additionally, the system integrates cutting-edge components such as the ESP32 cam for outdoor surveillance, RFID card reader, and fingerprint scanner. These components enhance the overall security by providing multiple layers of authentication and surveillance capabilities. The ESP32 cam ensures comprehensive monitoring of outdoor areas, while the RFID card reader and fingerprint scanner offer secure and convenient access control for residents. Together, these components create a robust security infrastructure that combines advanced technology with user-friendly features for optimal protection of the intelligent home.

Description of the components used in this section:

Table 3.1: Lists of components used for Security system.

DOOR LOCKING SYSTEM:	
• **Radio Frequency Identification (RFID) reader**	• **Servo(Door open/close)**
• **AS608 Optical Finger Sensor**	
OUTDOOR SURVEILLANCE:	
• **ESP 32 CAMERA MODULE**	• **SERVO*2 (PAN/TILT movement)**

3.2.1 Radio Frequency Identification (RFID) reader:

Radio Frequency Identification (RFID) is a technology that allows for the wireless transmission of data using radiofrequency waves. It involves two main components: RFID tags and RFID readers.

- **RFID Readers:** Also known as interrogators, RFID readers are devices that send out radiofrequency signals to communicate with RFID tags. When an RFID tag comes within the range of an RFID reader, it receives the radiofrequency signal and responds by transmitting its stored data. RFID readers can be fixed or mobile, with fixed readers typically used in stationary applications such as inventory tracking and access control systems, while mobile readers are used for applications requiring mobility, such as asset tracking in warehouses or logistics operations.

Fig 3.1: RFID reader

- **RFID Tags and Cards:** These are small electronic devices that consist of a microchip and an antenna. The microchip stores data, which can be unique identification numbers or other information, and the antenna enables communication with RFID readers. RFID tags come in various forms, including passive, active, and semi-passive. Passive tags do not have their own power source and rely on the energy from the RFID reader to transmit data. Active tags have their own power source, usually a battery, allowing them to transmit data independently. Semi-passive tags also have their own power source but rely on the RFID reader for communication.

Fig 3.2: RFID Tag and Card

RFID technology is widely used across various industries for applications such as inventory management, supply chain logistics, asset tracking, access control, and contactless payment systems. Its ability to provide fast, accurate, and automated data capture makes it valuable for improving efficiency, enhancing security, and streamlining processes in diverse environments. Additionally, RFID offers advantages such as scalability, durability, and versatility, making it suitable for a wide range of applications and environments.

3.2.2 AS608 Optical Finger Sensor:

The AS608 Optical Fingerprint Sensor is a biometric authentication device known for its high accuracy and robust security features. Utilizing optical sensing technology, it captures high-resolution fingerprint images and stores corresponding templates securely onboard. With fast response times, it enables quick and reliable authentication processes, making it suitable for various applications such as door locks, safes, and time attendance systems. Its versatility, compact size, and compatibility with different interfaces make it an ideal choice for applications requiring secure access control.

Fig 3.3: AS608 Optical Fingerprint Sensor

The AS608 Optical Fingerprint Sensor operates based on the principle of optical sensing. When a user places their finger on the sensor, the device emits a series of light pulses onto the surface of the finger. These light pulses illuminate the ridges and valleys of the fingerprint, creating a high-resolution image of the unique fingerprint pattern.

The sensor then captures this fingerprint image using an optical sensor array, which consists of numerous light-sensitive elements. Each element detects the intensity of light reflected from the fingerprint surface at a specific point. By analyzing the variation in light intensity across the sensor array, the device constructs a digital representation of the fingerprint.

Once the fingerprint image is captured, the sensor processes it using sophisticated algorithms to extract key features and characteristics. These features are then converted into a template, which is a mathematical representation of the fingerprint's unique pattern. This template is securely stored within the sensor's onboard memory.

During authentication, when a user attempts to access a system or device, they place their finger on the sensor again. The sensor captures a new fingerprint image and compares it with the stored template using pattern recognition algorithms. If the captured fingerprint matches the stored template within a predefined threshold, the authentication is successful, and access is granted.

Overall, the working principle of the AS608 Optical Fingerprint Sensor involves capturing fingerprint images using optical sensing technology, extracting key features, and comparing them with stored templates to authenticate users accurately and securely.

3.2.3 ESP-32 Camera Module:

The ESP32-CAM module is a versatile and compact development board based on the ESP32 microcontroller chip. It is specifically designed for applications requiring camera functionalities in addition to WiFi and Bluetooth connectivity. Here's a detailed description of the ESP32-CAM module:

- **ESP32 Microcontroller:** The heart of the ESP32-CAM module is the ESP32 microcontroller chip, which is a powerful yet low-cost system-on-chip (SoC) solution. It features a dual-core Xtensa 32-bit LX6 microprocessor, which can operate at frequencies up to 240 MHz. The ESP32 microcontroller also integrates WiFi and Bluetooth connectivity capabilities, making it ideal for IoT applications.

- **OV2640 Camera:** The ESP32-CAM module is equipped with an OV2640 camera module, which features a 2-megapixel CMOS image sensor. This camera can capture high-resolution still images as well as video footage. It supports various image formats, including JPEG, BMP, and YUV, and offers configurable parameters such as resolution, frame rate, and compression ratio.

- **GPIO Pins:** The ESP32-CAM module includes several GPIO (General Purpose Input/Output) pins that allow it to interface with external components and peripherals. These pins can be used for tasks such as digital input/output, analog input, PWM (Pulse Width Modulation) output, and more. The ESP32-S chip boasts a total of 32 GPIO pins; however, due to internal allocation for the camera and PSRAM, the ESP32-CAM module only offers 10 GPIO pins for external use. These pins are flexible and can be configured for various peripheral functions such as UART, SPI, ADC, and Touch by configuring the relevant registers through programming. Additionally, some GPIO pins are dedicated for camera control and communication.

- **SD Card Slot:** The ESP32-CAM module features an onboard microSD card slot, which allows users to expand the storage capacity for storing captured images and videos. This is particularly useful for applications that require local storage or data logging capabilities.

- **UART Interface:** The ESP32-CAM module includes UART (Universal Asynchronous Receiver-Transmitter) interfaces, which facilitate serial communication with external devices. This allows for easy integration with other microcontrollers, sensors, displays, and communication modules.

- **Programming and Development:** The ESP32-CAM module can be programmed using the Arduino IDE, which provides a familiar and easy-to-use development environment for writing, compiling, and uploading code. Additionally, the module supports OTA (Over-The-Air) firmware updates, allowing for remote updating of software without physical access to the device.

Overall, the ESP32-CAM module offers a cost-effective and feature-rich solution for projects requiring camera capabilities, WiFi connectivity, and Bluetooth communication. Its compact size, versatility, and ease of use make it suitable for a wide range of IoT applications, including home automation, surveillance systems, smart devices, and more.

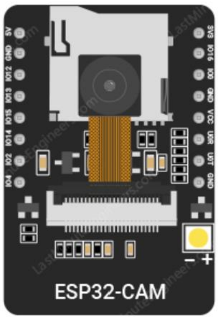


Fig 3.4: ESP 32 CAM

3.3 Implementation:

The implementation of the security system involves integrating various components to create a robust and comprehensive security infrastructure for the intelligent home. Here's how each component is implemented:

3.3.1 RFID Card Reader as Door locking System:

The RFID card reader is installed near the entrance door(servo) to authenticate users using RFID cards. When a user presents their RFID card to the reader, it detects the card's unique identifier and grants access if authorized by moving the servo assigned for door to door opening angle. For our project we have used both TFID card and tag with the corresponding hex values:

```
byte knownID1[] = {0xF3, 0x1D, 0x68, 0xA6}; // TFID CARD 1
byte knownID2[] = {0x2A, 0xE3, 0x0F, 0x0B}; // TFID TAG  2
```

Hence, if the TFID reader recognizes one of these two known IDs then the **dooropen();** function is called and after 2 seconds **doorclose()** function is called.

```
void OpenDoor()
 {
  Door.write(dooropenangle);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DOOR OPEN!");
  delay(1000);
  lcd.clear();
 }
 void CloseDoor()
 {
  Door.write(doorcloseangle);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DOOR CLOSED!");
  delay(1000);
  lcd.clear();
 }
```

```
    if (checkID1(mfrc522.uid.uidByte, mfrc522.uid.size) ||
checkID2(mfrc522.uid.uidByte, mfrc522.uid.size))

  {
    OpenDoor();
    delay(2000);// open door for 2 seconds and then close
    CloseDoor();
```

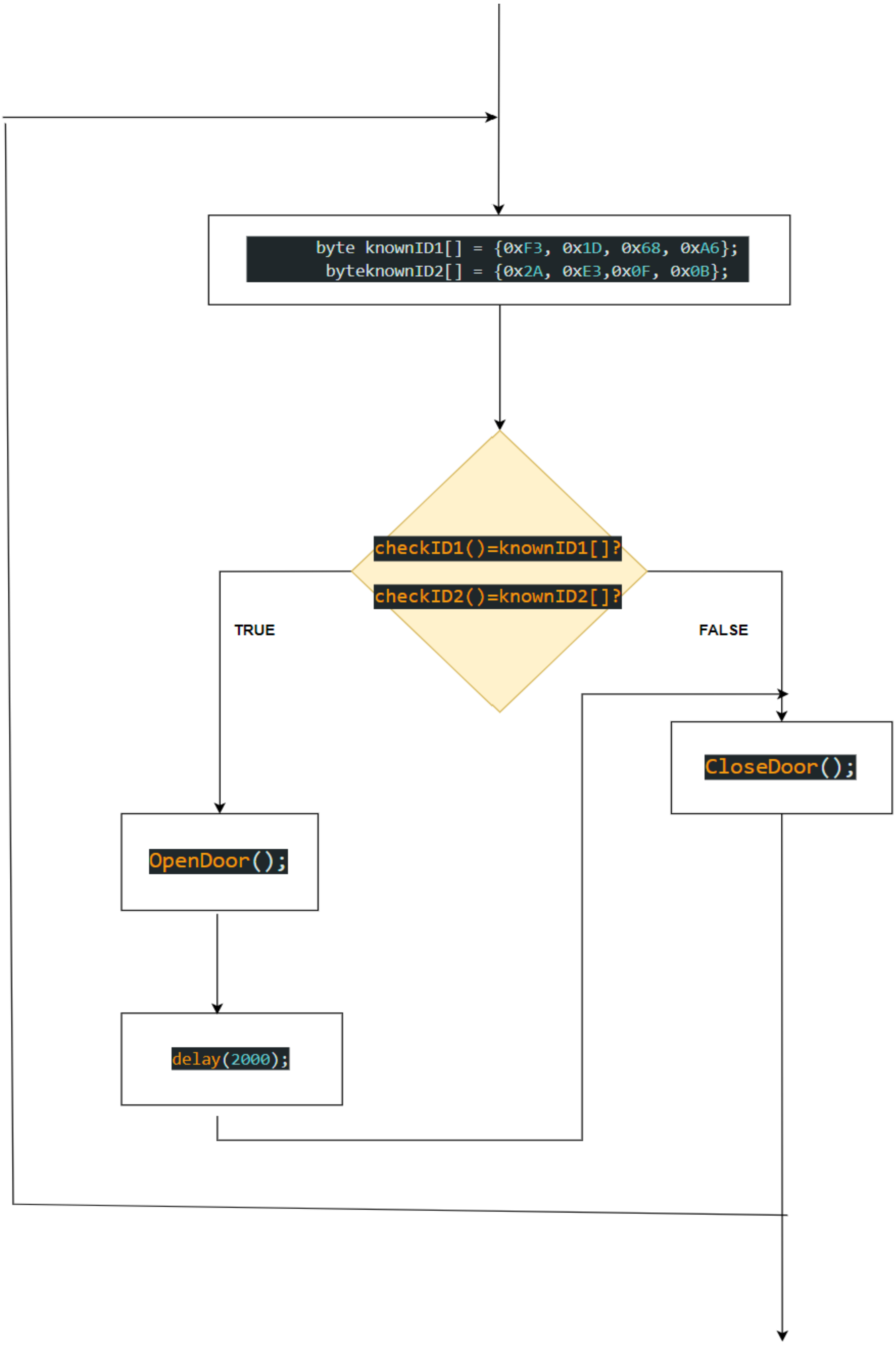


Fig 3.5: Operation of RFID for Door Locking System.

Fig 3.6: Function of Door Servo with response to TFID card/tag detected.

3.3.2 AS608 Optical Finger Sensor for Door Locking System:

The AS608 fingerprint sensor is integrated into the door lock mechanism to provide biometric authentication. Users can register their fingerprints, and the sensor compares the scanned fingerprint with stored templates to authenticate and unlock the door. The program for this sensor works as follow:

```
//FINGERPRINT SENSOR FOR DOOR
  if (finger.getImage() == FINGERPRINT_OK) {
    if (finger.image2Tz() == FINGERPRINT_OK) {
      if (finger.fingerSearch() == FINGERPRINT_OK) {
        OpenDoor();
        delay(2000);
        CloseDoor();
      }
    }}
```

- It first checks if the fingerprint sensor successfully captures an image of the fingerprint **(finger.getImage()).**
- If the image is captured successfully, it converts the image into a template **(finger.image2Tz()).**
- Then, it searches for a matching fingerprint template stored in the sensor's memory **(finger.fingerSearch()).**
- If a match is found, it proceeds to open the door **(OpenDoor()),** delays for 2 seconds, and then closes the door **(CloseDoor()).**

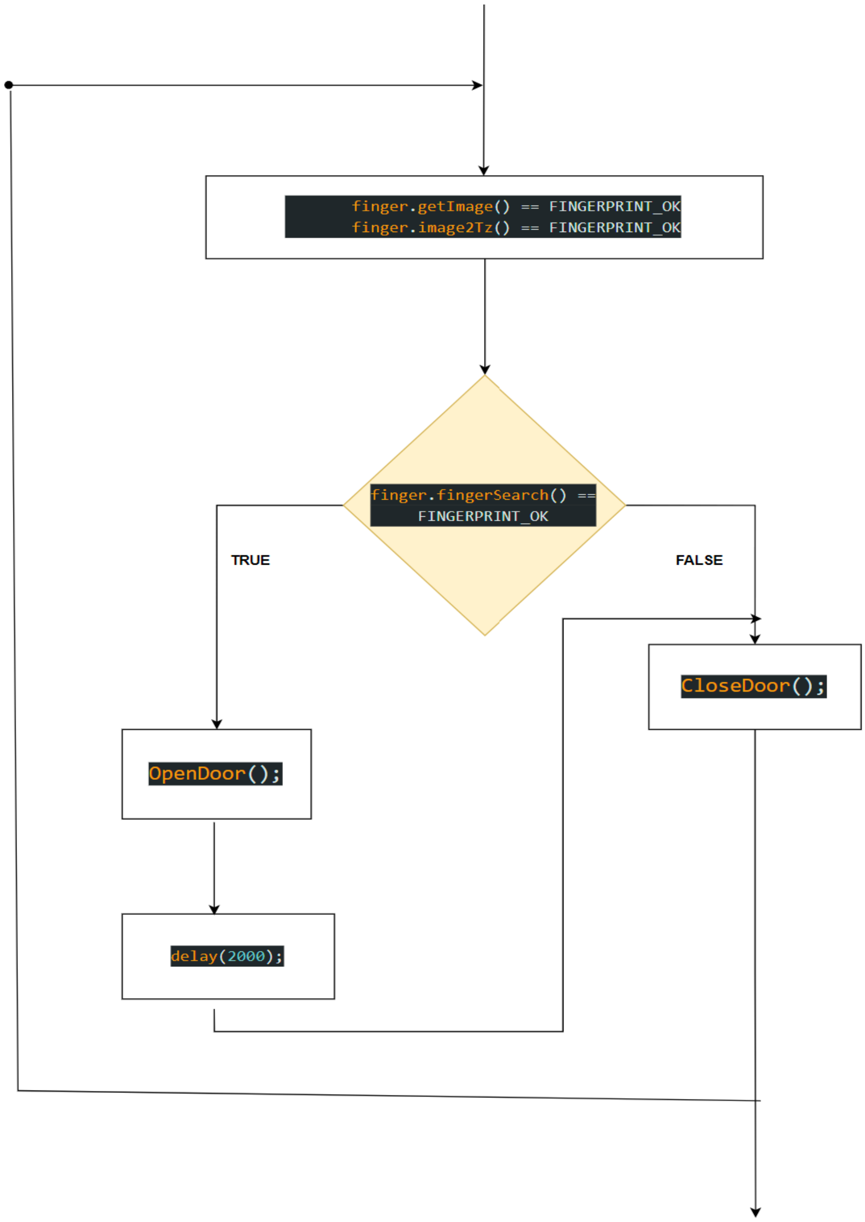


Fig 3.7: Operation of AS608 Optical Finger Sensor for Door Locking System.

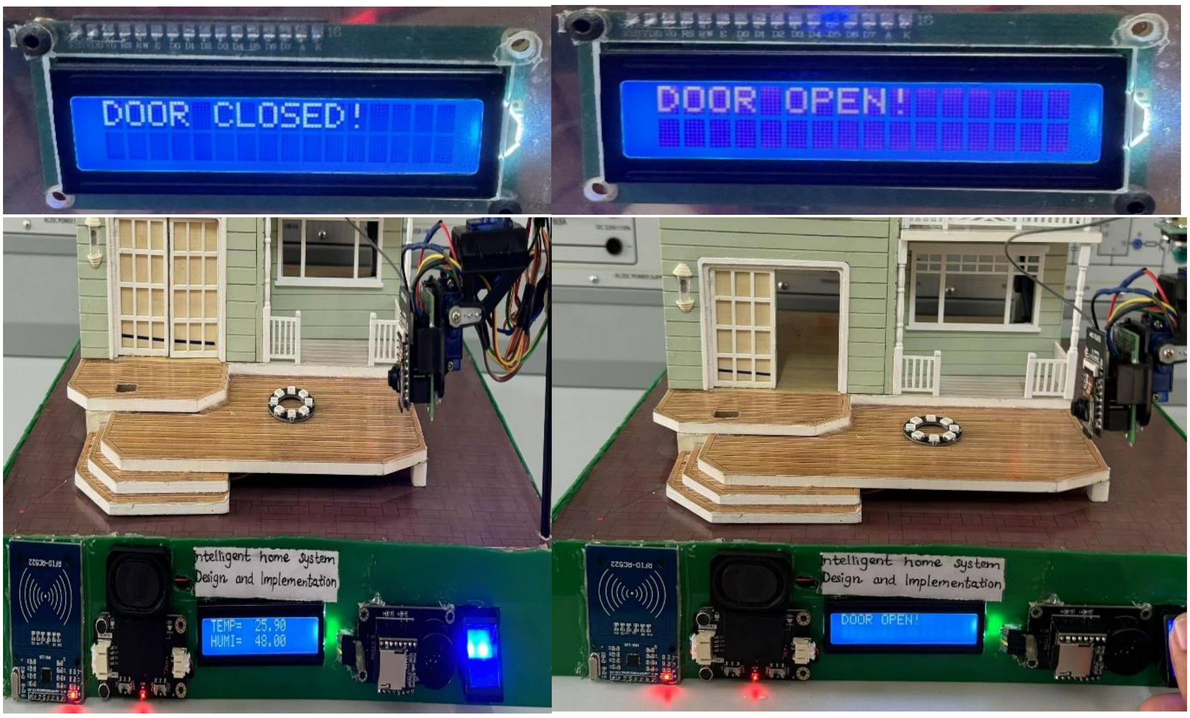


Fig 3.8: Function of Door Servo with response to stored finger matches.

3.3.3 Door locking system overall functionality and circuit diagram:

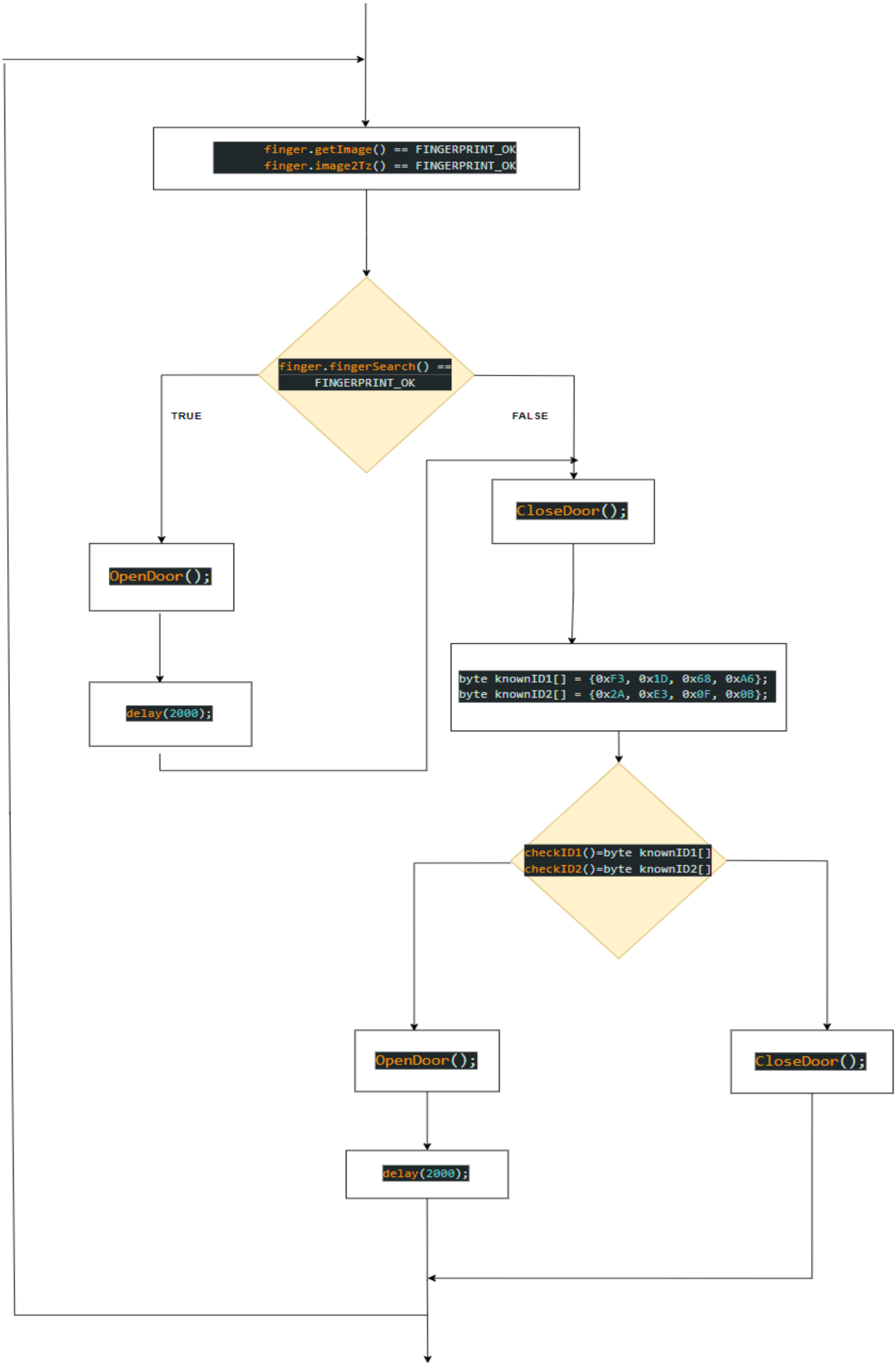


Fig 3.9: Operation of overall Door Locking System

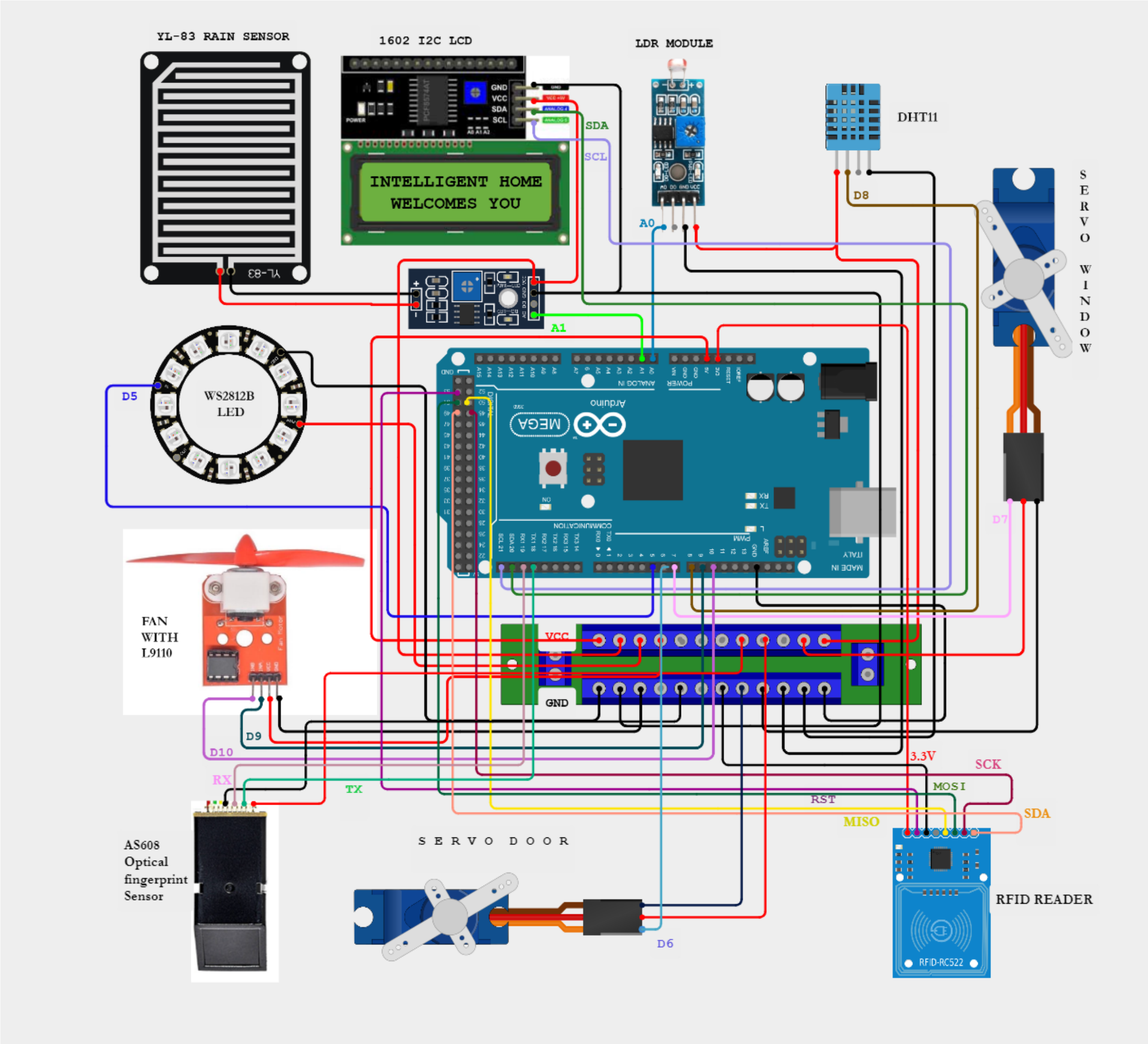


Fig 3.10: Circuit Diagram with Security system .

3.3.4 Outdoor Surveillance with ESP 32 camera module:

For outdoor surveillance, we'll set up the ESP32-CAM on a pan-and-tilt stand featuring two SG90 servo motors. This setup enables us to adjust the camera's position both vertically (up and down) and horizontally (left and right), providing comprehensive coverage for surveillance needs.

The ESP32-CAM functions as a web server, offering live video streaming and servo motor control capabilities. Through a user-friendly web interface, individuals can access the video feed and manipulate buttons to adjust the camera's orientation as required. This setup offers flexibility and convenience for outdoor surveillance tasks. Servo motors are pivotal in this setup, as they allow precise control over the camera's movements. By adjusting the shaft's position between 0 and 180 degrees, we can direct the camera's view effectively.

Controlling servo motors typically involves pulse width modulation (PWM) signals, where the duration of the signal pulse determines the shaft's position. To simplify this process, we can utilize the ESP32Servo library, which abstracts the PWM signal management, making it easier to control the servo motors directly from the ESP32-CAM. This approach streamlines the implementation and management of the surveillance system. For the better range and connection, we can use external antenna.

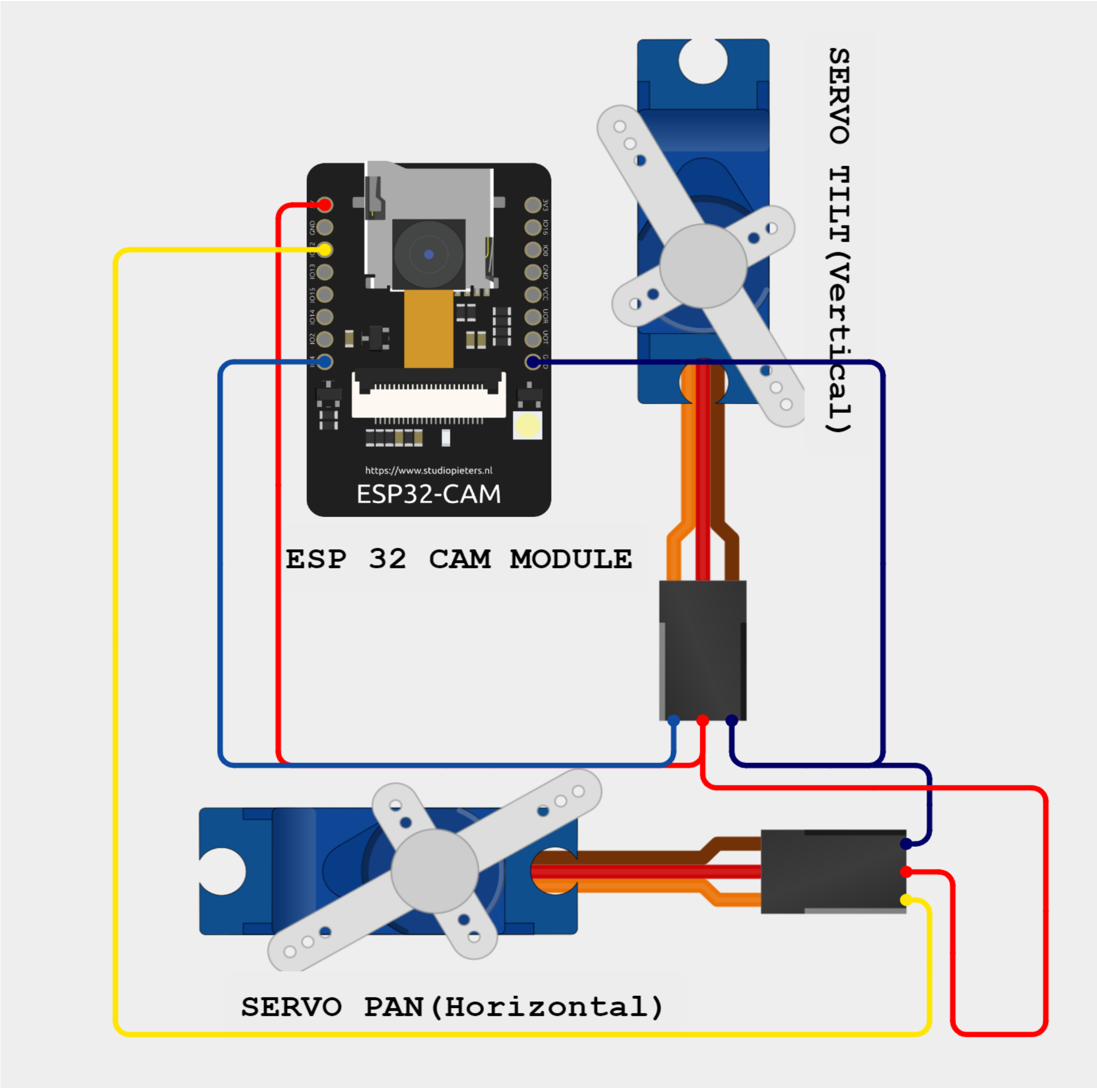


Fig 3.11: Connection of servo with ESP 32 cam

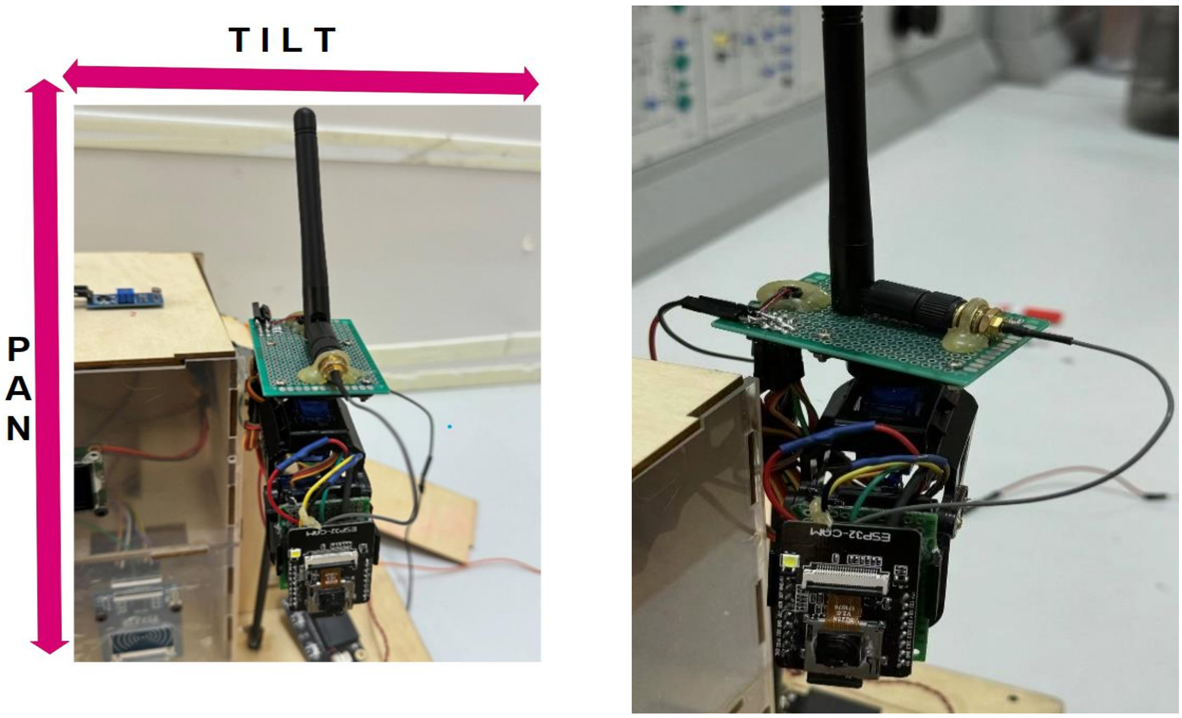


Fig 3.12: PAN/ TILT MOVEMENT OF ESP 32 via webserver

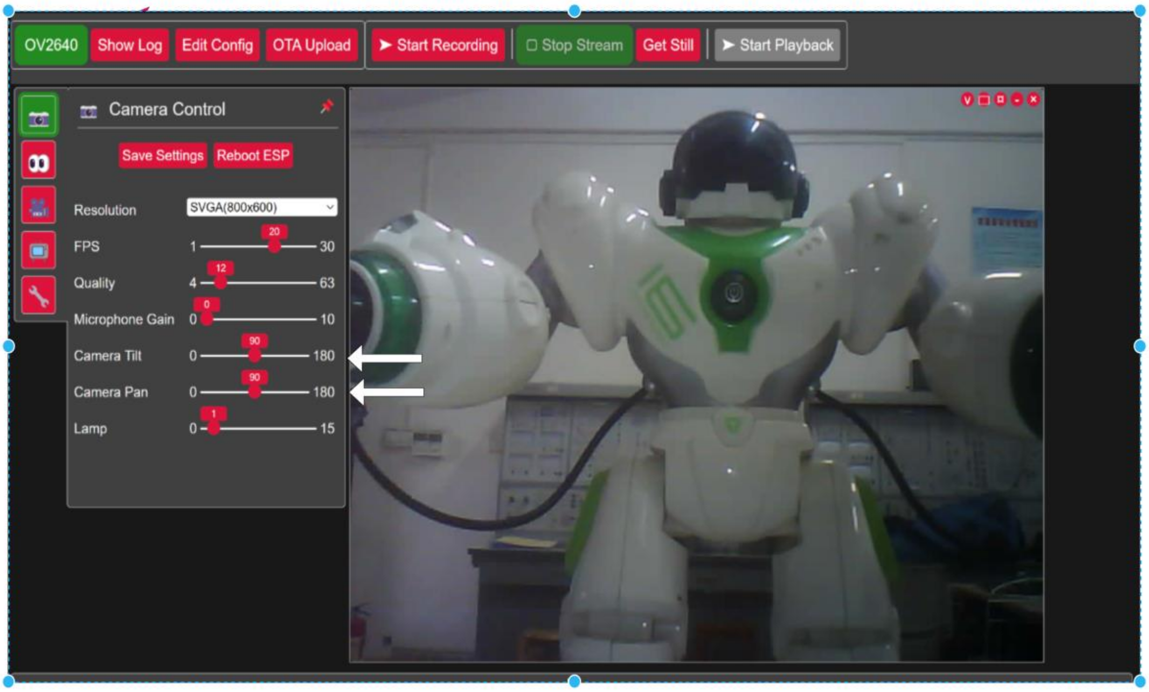


Fig 3.13: Web Interface for Surveillance

Indeed, the setup where the ESP32-CAM hosts a web server for video streaming and servo motor control via buttons offers remarkable versatility for surveillance applications. By accessing the web page remotely, users can effortlessly manipulate the camera's position, enabling real-time monitoring of various areas according to the camera's orientation.

This functionality provides a dynamic surveillance solution, allowing users to adapt to changing scenarios and focus on specific areas of interest as needed. Whether it's monitoring different sections of a property, observing specific points of entry, or scanning the surroundings for any unusual activity, the ability to control the camera remotely via the web interface enhances situational awareness and responsiveness.

Overall, the integration of a web server with video streaming and servo motor control buttons offers a convenient and effective means of conducting surveillance, making it an ideal solution for various security and monitoring applications.

3.4 Results and Discussion:

3.4.1 Evaluation of Door Locking System and Outdoor Surveillance System:

The door locking system, incorporating RFID card reader and fingerprint sensor, showed reliable performance in granting access to authorized users while keeping unauthorized individuals out. The integration of these components provided multiple layers of security.

The outdoor surveillance system, utilizing the ESP32-CAM mounted on a pan-and-tilt stand, offered flexibility in monitoring different areas. The web server hosting live video streaming and servo motor control buttons facilitated remote surveillance with ease.

3.4.2 Security Implications and User Experience:

- The security implications are significant, as the implemented systems enhanced the overall security of the intelligent home. Users experiences improved peace of mind knowing their property was equipped with advanced security features.

- The residents will be satisfied with intuitive interfaces for both the door locking system and outdoor surveillance. Remote access capabilities provided convenience, allowing users to monitor and control security aspects remotely.

3.4.3 Effectiveness in Detecting and Deterring Intruders:

- The systems demonstrated effectiveness in detecting and deterring intruders. The door locking system prevented unauthorized access, while the outdoor surveillance system allowed for real-time monitoring and response to suspicious activities.
- The integration of multiple security layers, including biometric authentication and live video streaming, enhanced the overall effectiveness of the security setup.

3.4.4 Privacy Considerations:

- Privacy considerations were taken into account during the implementation of the systems. Data encryption and secure communication protocols were utilized to protect sensitive information, such as fingerprint templates and live video feeds.
- Users were provided with options to configure privacy settings and restrict access to certain functionalities, ensuring their privacy was respected while maintaining security.

3.4.5 Future Enhancements:

- Future enhancements could include integrating artificial intelligence algorithms for advanced motion detection and object recognition in the outdoor surveillance system.
- Implementing additional security features, such as facial recognition and voice authentication, could further enhance the door locking system's capabilities.
- Integration with smart home automation systems could enable seamless coordination between security devices and other home appliances, enhancing overall functionality and convenience.

Overall, the evaluation of the door locking system and outdoor surveillance system demonstrated their effectiveness in enhancing home security while providing a positive user experience. By addressing security implications, privacy considerations, and potential future enhancements, the intelligent home security setup can continue to evolve and adapt to emerging threats and user needs.

4. Hazard Detection:

In this section, we explore the implementation of hazard detection mechanisms within the intelligent home project. The primary objective is to enhance safety by identifying potential hazards such as gas leaks or fire incidents using the MQ-135 gas sensor.

4.1. Introduction to Hazard Detection:

The critical need for integrating hazard detection mechanisms into the intelligent home project is to prioritize occupant safety during the hazardous situations like fire or gas leaks. We emphasize the importance of early detection of potential hazards such as gas leaks and fire incidents to prevent property damage and ensure the well-being of occupants. Additionally, we underscore the pivotal role of the MQ-135 gas sensor in identifying various harmful gases commonly found in indoor environments.

Moreover, we discuss the necessity of implementing comprehensive alert systems, including audible alarms and emergency red lighting, to promptly notify occupants of detected hazards. Furthermore, we emphasize the importance of initiating proactive measures such as automatically opening windows and doors to facilitate quick evacuation in the event of an emergency, thereby minimizing potential risks and ensuring swift response and evacuation procedures.

4.2. System Design:

Firstly, we meticulously consider the requirements and factors involved in selecting suitable sensors, prominently featuring the MQ-135 gas sensor, known for its reliability in detecting various harmful gases commonly found in indoor environments. These sensors are strategically positioned within the home environment to ensure comprehensive coverage and efficient detection of potential hazards.

One crucial aspect of the system design is the incorporation of alarm alerts and emergency red lightings. Upon detecting a hazardous situation, the system triggers audible and visual alarms to alert occupants, while emergency red lightings are activated to provide clear visual cues for evacuation routes. Additionally, the system initiates automated actions such as opening windows and doors to facilitate quick evacuation and ensure the safety of occupants.

By combining effective hazard detection with swift response mechanisms, the system aims to minimize potential risks and enhance the overall safety and well-being of the intelligent home occupants.

- **Description of the components used in this section:**

Table 4.1: Lists of components used for Hazard Detection System.

HAZARD DETECTION SYSTEM			
• MQ-135 gas sensor	• Piezo Buzzer	• LED (WS2812B RGB)	• Servo(Door and Window)

4.2.1 MQ-135 gas sensor:

An MQ135 air quality sensor is one type of MQ gas sensor used to detect, measure, and monitor a wide range of gases present in air like ammonia, alcohol, benzene, smoke, carbon dioxide, etc. It operates at a 5V supply with 150mA consumption. Preheating of 20 seconds is required before the operation, to obtain the accurate output.

Fig 4.1: MQ135 gas sensor

This sensor is particularly sensitive to gases such as NH3, NOx, CO2, benzene, and smoke, making it suitable for monitoring air quality in various applications. It is cost-effective and widely used for detecting and monitoring harmful gases in the atmosphere. When the concentration of gases exceeds a predefined threshold in the air, the digital output pin of the sensor becomes high. The threshold value can be adjusted using the potentiometer present on the sensor. Additionally, the sensor provides an analog output voltage from its analog pin, offering an approximate indication of the gas level present in the air.

4.2.2 Piezo Buzzer:

A piezo buzzer is an electronic device employed for generating tones, alarms, or sounds. It possesses a lightweight and straightforward construction, often available at a low cost. Despite its simplicity, depending on the specifications of the piezo ceramic buzzer, it offers reliability and can be manufactured in various sizes to accommodate different frequencies, resulting in diverse sound outputs.

Fig 4.2: Piezo Buzzer

The utilization of piezo ceramic buzzers stems from an application of the piezoelectric principle, initially discovered by **Jacques and Pierre Curie** in 1880. Their findings revealed that certain materials could generate electricity when subjected to mechanical pressure, and conversely, could produce mechanical deformation when exposed to an electric field.

Piezo buzzers, typically composed of synthetic piezoceramic materials, operate on this principle. When these materials encounter an alternating electric field, they undergo sequential stretching and compression in line with the field's frequency, resulting in audible sound production.

In contrast to magnetic buzzers, which operate within a narrow voltage range of approximately one to 16 volts, piezo buzzers exhibit a broader operational voltage spectrum ranging from three to 250 volts. Moreover, magnetic buzzers tend to have higher power consumption, typically ranging from 30 to 100 milliamperes, whereas piezo buzzers generally consume less than 30 milliamperes, even at higher frequency rates. Although piezo buzzers necessitate a larger physical footprint compared to magnetic buzzers, they offer a higher sound pressure level.

4.3. Implementation:

The hazard detection system in the intelligent home design aims to enhance safety by identifying potential hazards such as gas leaks or fire incidents. The implementation involves integrating sensors, actuators, and indicators to detect, alert, and respond to hazardous conditions effectively.

```
const int Air= A2,buzzer= 3;
int air;
void firehazards(){
  lcd.clear();
  digitalWrite(buzzer,LOW);
  lcd.setCursor(0, 0);
  lcd.print("FIRE!!!EVACUATE!!");
   for (int i = LED.numPixels() - 1; i >= 0; i--) {
    LED.setPixelColor(i, LED.Color(255, 0, 0));
    LED.show();
    delay(10);
    LED.setPixelColor(i, 0);}
  digitalWrite(buzzer,HIGH);
  delay(250);
  lcd.clear();}
void loop() {
air=analogRead(Air);
if (air > 200)// HAZARD DETECTION FOR POSSIBLE GAS LEAKS OR FIRE{
  firehazards();  OpenDoor(); windowopen();}

}
```

In the provided code snippet, the hazard detection system is implemented using an MQ135 gas sensor, a piezo buzzer, an RGB LED (WS2812B), and servo motors for controlling the door and window.

- **MQ135 Gas Sensor:**
  The MQ135 gas sensor is used to detect hazardous gases in the air, such as ammonia, carbon dioxide, and smoke. In the code, the **air** variable represents the air quality reading obtained from the MQ135 sensor. When the air quality reading exceeds a predefined threshold (**200** in this case), it indicates the presence of hazardous gases.
- **Piezo Buzzer:**
  The piezo buzzer connected to **digital pin 3** is employed to provide an audible alarm when a hazardous condition is detected. In the **firehazards()** function, the buzzer is activated by setting the **buzzerPin** to **HIGH**, generating an audible alarm signal. The buzzer remains active for a specified duration (**1000 milliseconds**) to alert occupants of the hazardous situation.
- **RGB LED (WS2812B):**
  The RGB LED is used to provide visual indication of the hazardous condition.
  When the **fireHazard()** function is triggered, the LED is set to **red** color **(LED.Color(255, 0, 0)) to** signify danger. The LED remains illuminated in red for the same duration as the buzzer alarm.
- **Servo Motors (Door and Window Control):**
  Servo motors are employed to automate the opening of the door and window to facilitate evacuation in case of fire or gas leaks. Separate functions (**OpenDoor()** and **windowopen()**) control the servo motors connected to the door and window, respectively. When the hazardous condition is detected, both the door and window servo motors are activated, causing them to open for quick evacuation.

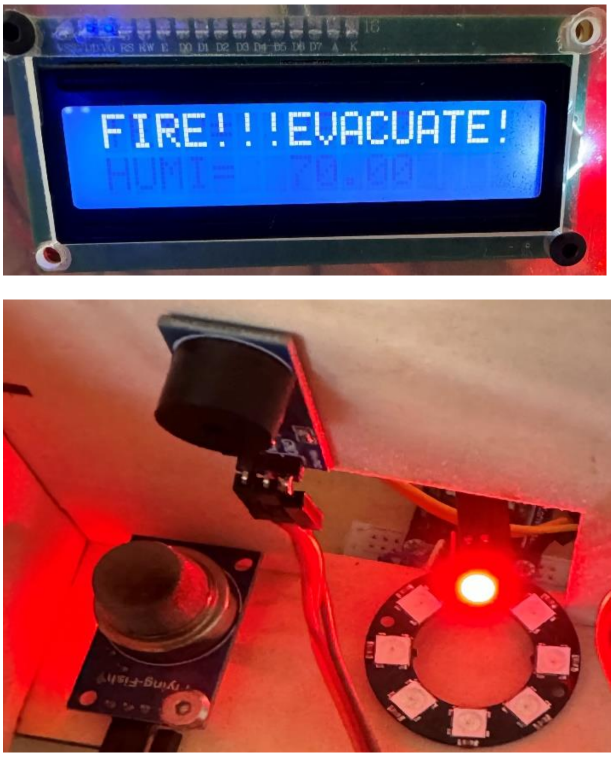


Fig 4.3: Function of Buzzer and WS2818B LED in response to gas hazard detection.

4.3.1 Hazard Detection system overall functionality and designed circuit.

The overall operation for the hazard detection system is illustrated with the flowchart provided along with the circuit for this system.

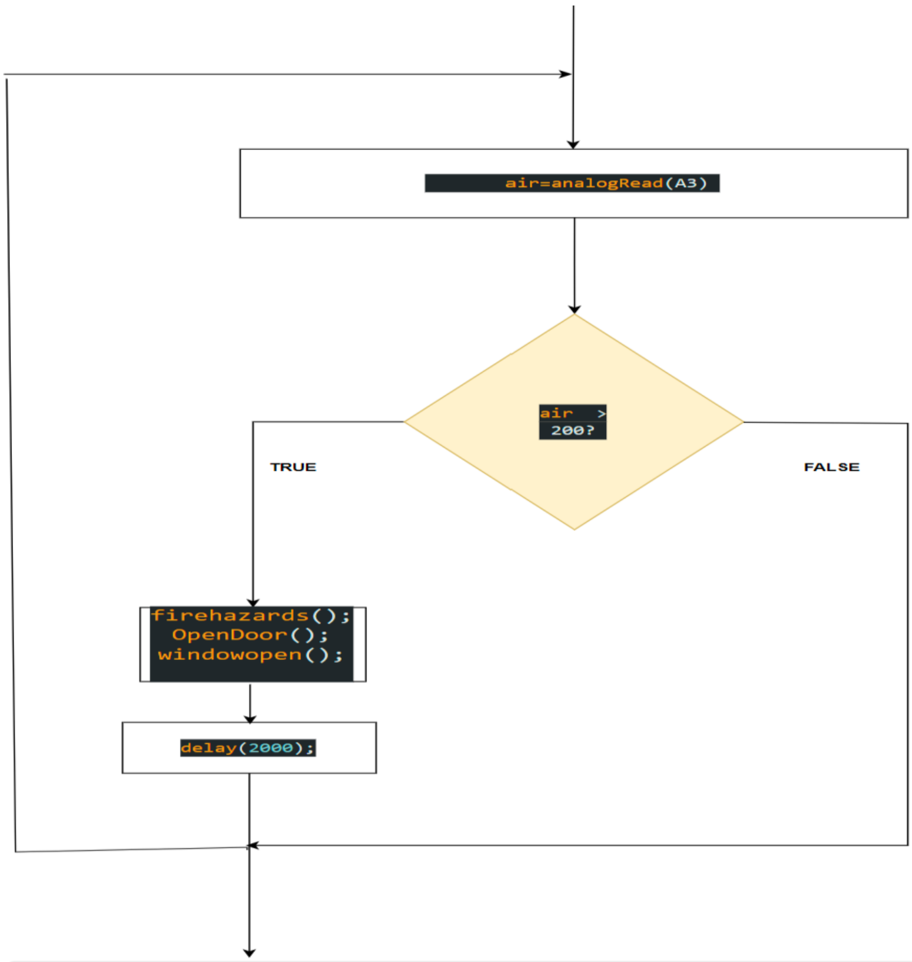


Fig 4.4: Operation of overall Hazard Detection System

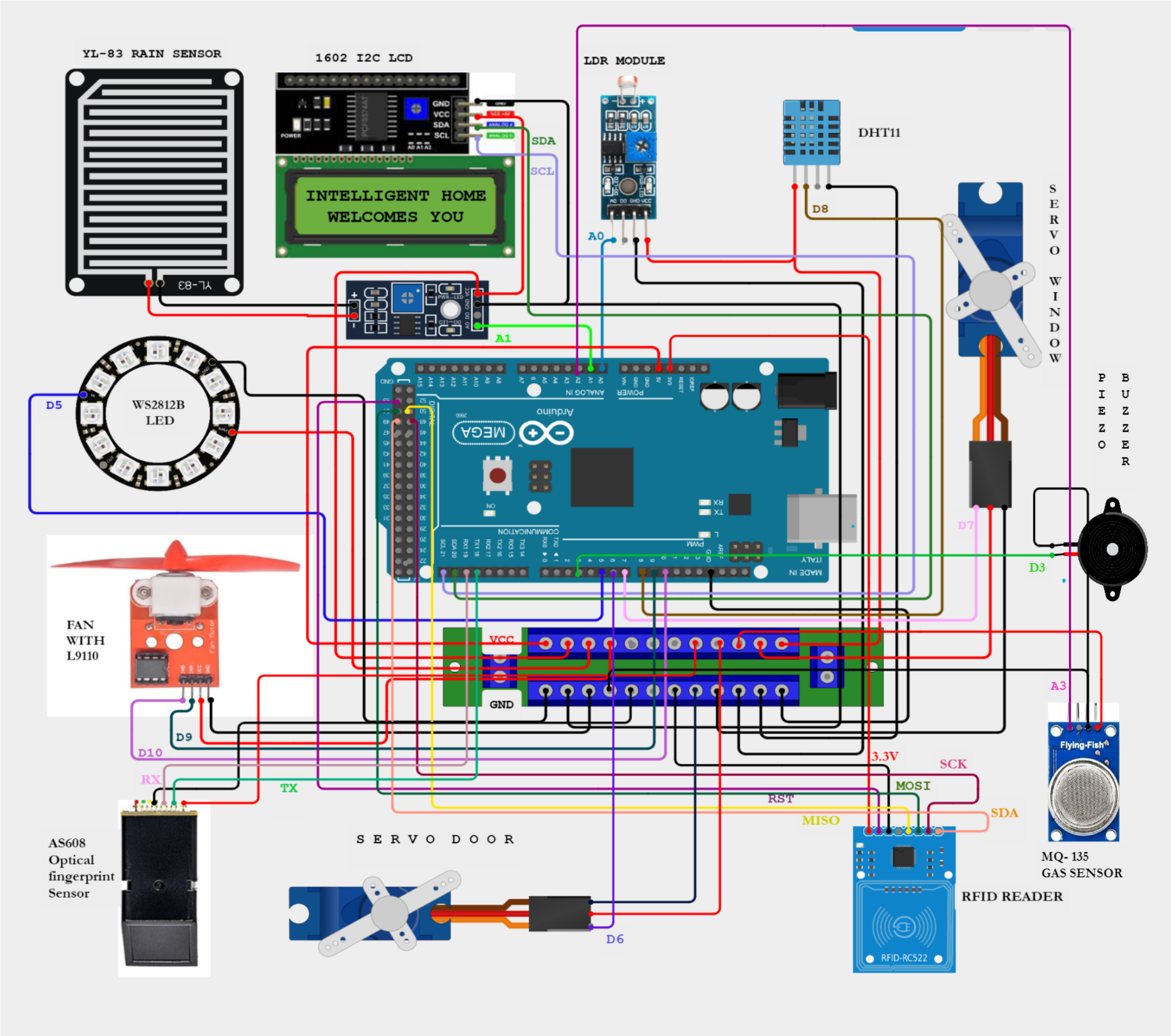


Fig 4.5: Circuit Diagram with Hazard Detection System.

4.4. Results and Discussion:

The hazard detection system implemented in the intelligent home design plays a crucial role in enhancing safety by identifying and mitigating potential hazards such as gas leaks or fire incidents. In this section, we evaluate the performance of the hazard detection system, discuss its effectiveness in identifying and mitigating hazards, and explore potential future enhancements.

4.4.1. Evaluation of Hazard Detection System:

The performance of the hazard detection system was evaluated based on its ability to accurately detect hazardous conditions and trigger appropriate responses. Testing scenarios included simulated gas leaks and fire incidents to assess the system's responsiveness and reliability.

Overall, the hazard detection system demonstrated reliable operation, effectively detecting hazardous conditions, and activating response mechanisms in a timely manner. The MQ135 gas sensor provided accurate readings of air quality, enabling the system to identify elevated levels of harmful gases indicative of potential hazards.

4.4.2. Effectiveness in Identifying and Mitigating Hazards:

The hazard detection system proved to be effective in identifying various hazards, including gas leaks and fire incidents. When hazardous conditions were detected, the system promptly activated audible and visual alarms using the piezo buzzer and RGB LED, respectively. Additionally, servo motors were deployed to automatically open the door and window, facilitating quick evacuation of occupants.

The coordinated response of the hazard detection system contributed to mitigating the identified hazards, allowing occupants to take appropriate action to ensure their safety. The audible and visual alarms provided timely alerts, while the automated opening of the door and window facilitated swift evacuation, minimizing potential risks to occupants.

4.4.3. Future Enhancements:

While the current hazard detection system performed satisfactorily, several enhancements could be implemented to further improve its functionality and effectiveness. Some potential future enhancements include:

Integration of additional sensors: Incorporating additional sensors, such as smoke detectors and heat sensors, could enhance the system's capability to detect a broader range of hazards, including fires of varying intensity.

Enhanced communication and notification features: Integrating the hazard detection system with a centralized monitoring platform or mobile application could enable real-time alerts and notifications to be sent to occupants, emergency services, or designated contacts in the event of a hazard.

Intelligent decision-making algorithms: Implementing intelligent algorithms to analyze sensor data and prioritize response actions based on the severity and type of hazard could optimize the system's effectiveness in mitigating risks while minimizing false alarms.

Redundancy and failover mechanisms: Implementing redundant sensor arrays and failover mechanisms could enhance the reliability and robustness of the hazard detection system, ensuring continuous operation even in the event of sensor or component failures.

Overall, the hazard detection system represents a critical component of the intelligent home design, contributing to the safety and well-being of occupants. Continual refinement and enhancement of the system's capabilities will be essential to address evolving safety requirements and ensure optimal performance in various hazardous scenarios.

5. Control Design:

In the context of the intelligent home project, the utilization of **IR remote** and **DFRobot Gravity voice recognition module** represents a manual control approach to interact with the home appliances. Unlike the automated control system based on weather analysis and other sensors, which **autonomously** adjusts settings and activates devices based on predefined conditions, the **manual control** mechanisms empower users with direct control over individual appliances and systems.

5.1 Introduction to Control Design:

Control design plays a crucial role in home automation by providing the means to manage and regulate various components and systems within an intelligent home. It involves designing algorithms, interfaces, and mechanisms to monitor, control, and automate devices such as lights, appliances, security systems, and environmental controls.

In this section, we will explore the use of IR remote and DFrobot Gravity voice recognition module to control the home appliances in the intelligent home project. These control mechanisms provide convenient and intuitive ways for users to interact with the automation system, enabling them to manage various devices and systems with ease.

The integration of IR remote technology enables users to effortlessly manipulate home appliances with the press of a button, offering a tactile and familiar interface for controlling functions such as lighting, entertainment systems, and climate control. Meanwhile, the DFrobot Gravity voice recognition module introduces a futuristic dimension to home automation, allowing users to issue commands and execute tasks through simple voice prompts, enhancing convenience and accessibility.

By harnessing the power of these advanced control mechanisms, homeowners can seamlessly orchestrate their intelligent home environment according to their preferences and needs. Whether it's adjusting lighting levels, setting room temperatures, arming security systems, or initiating entertainment modes, the IR remote and voice recognition technologies empower users to manage their smart home ecosystem with unparalleled ease and efficiency.

5.2 User Interfaces:

In our intelligent home design, we have adopted two primary methods for user interaction and control: Infrared (IR) remote control and Voice Control, facilitated by the DFrobot Gravity voice recognition module. These methods provide intuitive and convenient ways for users to manage and interact with various components and systems within the intelligent home environment.

5.2.1 Infrared (IR) Remote Control:

IR Remote control uses Infrared light to transmit data. Infrared light is below the visible spectrum. IR subjects to interference. Some common sources of interference include sunlight, lighting system in our home (LEDs are less interference prone than the incandescent lights).

IR remote works by sending the pulses of IR light as receiving source. To prevent interference, these pulses are modulated usually at the frequency of **38KHz.** Each button on the remote sends a **unique code** to the receiving sources. This code differs between the manufacturers. As a receiver of our IR remote, we use the device called as **1838IR** sensor and it has three pins **VCC, GND and Data pin**.

As initial state (without receiving signal) the high signal is output from data pin, and whenever the receiver detects pulses from any button of the IR remote, the signal is sent in the number of pulses. 1838IR is demodulating the modulated signal and hence determines which key in the remote is pressed.

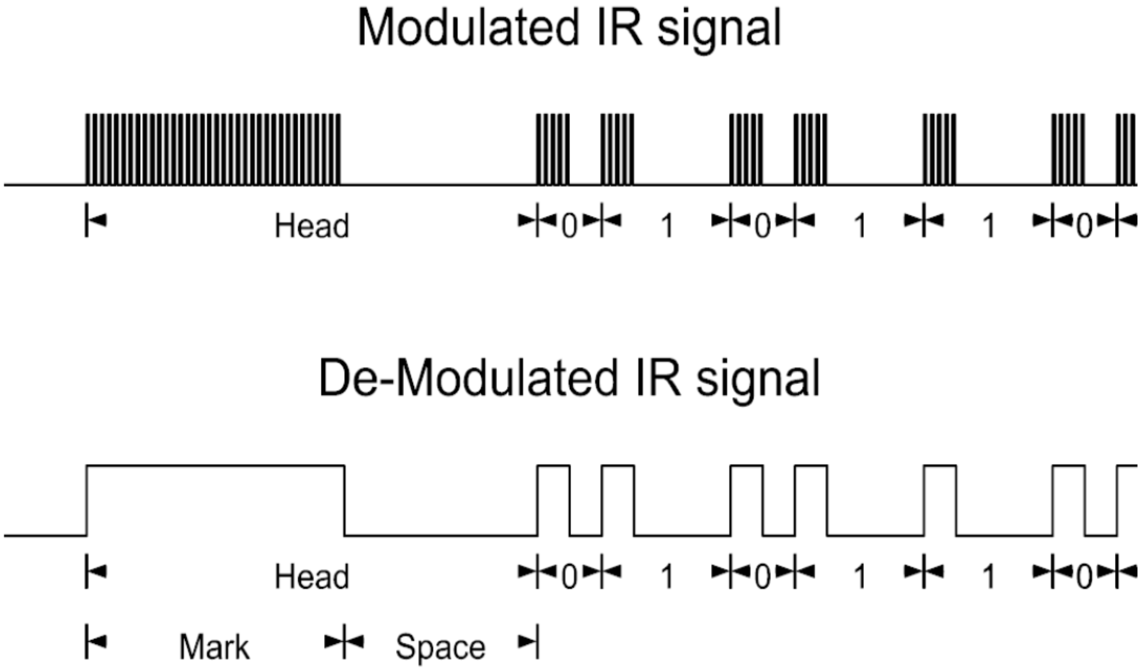


Fig 5.1: Demodulation of the modulated IR signal

IR remote control is a familiar and widely used method for controlling electronic devices. It allows users to send commands to devices using infrared signals emitted by the remote-control unit. In our intelligent home setup, users can utilize IR remote controls to adjust lighting, control entertainment systems, operate appliances, and perform other functions with ease. This method offers tactile feedback and is particularly suitable for users who prefer physical buttons for control.

- **Components Used for IR Remote Control:**

Table 5.1: Lists of components used for IR Remote Control.

IR Remote Control		
• IR Receiver	• IR remote	• DF Mini MP3 player

5.2.1.1 IR Receiver:

In our intelligent home design, the IR receiver serves as a crucial component for facilitating IR remote control functionality. The IR receiver module detects infrared signals emitted by remote control units and translates them into electrical signals that can be interpreted by the microcontroller unit (MCU), such as the Arduino Mega 2560, which serves as the brain of our automation system.

Fig 5.2: Infrared IR sensor Receiver

Here's how the IR receiver works within our intelligent home setup:

Detection of Infrared Signals: When a user presses a button on the IR remote control unit, it emits a series of infrared signals corresponding to the specific command or function.
Reception by IR Receiver: The IR receiver module, typically connected to one of the digital input pins of the MCU, receives these infrared signals. It consists of a photodiode that detects the incoming infrared radiation.
Signal Demodulation: The IR receiver module demodulates the received infrared signals, converting them into electrical signals that represent the digital data transmitted by the remote control.
Interpretation by MCU: The MCU, equipped with appropriate firmware and libraries, interprets the electrical signals received from the IR receiver module. It identifies the specific command or function encoded in the signal and initiates the corresponding action or response.
Control of Home Devices: Based on the interpreted command, the MCU triggers the appropriate control logic to interact with various devices and systems within the intelligent home environment. This may include adjusting lighting, controlling appliances, activating security measures, or performing other predefined tasks.

By incorporating an IR receiver into our intelligent home design, we enable users to interact with the automation system using conventional IR remote control units, providing a familiar and user-friendly interface for managing home devices and systems. This enhances the accessibility and convenience of our automation solution, catering to a wide range of user preferences and lifestyles.

5.2.1.2 IR remote:

The IR remote serves as a convenient and familiar method for users to interact with various devices and systems within the home automation setup. The IR remote control unit typically consists of a handheld transmitter with buttons corresponding to different functions or commands.

Fig 5.3: Infrared Remote

Here's how the IR remote is utilized in our intelligent home system:

Transmission of Infrared Signals: When a user presses a button on the IR remote, it emits a series of infrared signals corresponding to the specific command or function selected by the user. Each button on the remote is associated with a unique infrared signal pattern.
Reception by IR Receiver: The infrared signals emitted by the IR remote are detected by an IR receiver module installed within the intelligent home environment. The IR receiver module translates the received infrared signals into electrical signals that can be interpreted by the microcontroller unit (MCU) or other control devices.

Interpretation by MCU: The MCU, such as the Arduino Mega 2560, interprets the electrical signals received from the IR receiver module. It decodes the signal patterns to identify the specific command or function selected by the user.
Execution of Commands: Based on the interpreted command, the MCU initiates the corresponding action or response within the intelligent home system. This may include turning on/off lights, adjusting room temperature, controlling entertainment devices, or activating security features, among other tasks.
User Interaction and Control: The IR remote provides users with a simple and intuitive interface for controlling various aspects of the intelligent home environment. Users can conveniently operate different devices and systems from a distance using the handheld remote, enhancing convenience and accessibility.

By integrating IR remote functionality into our intelligent home design, we offer users a familiar and user-friendly method for interacting with the automation system. This promotes ease of use and enhances the overall user experience, making home automation more accessible and convenient for occupants of the intelligent home.

5.2.1.3 DF Mini MP3 player:

The DF Mini MP3 player serves as an integral component in our intelligent home design, facilitating the playback of recorded voice prompts or audio feedback corresponding to specific actions or events within the automation system.

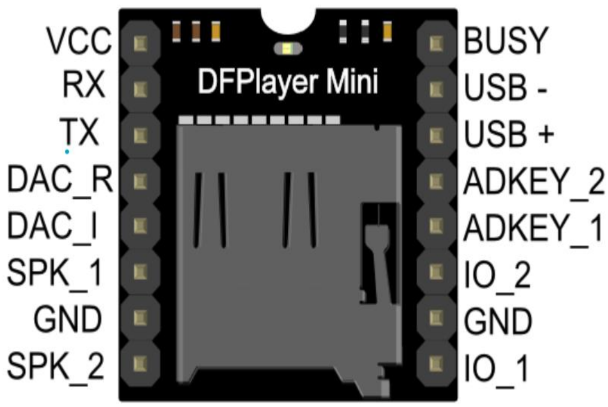


Fig 5.4: DFMini MP3 Player

Here's how it is utilized:

Voice Prompt Storage: Recorded voice prompts or audio feedback are stored on an SD card inserted into the DF Mini MP3 player. These audio files are pre-recorded and programmed to correspond to specific actions, events, or commands within the intelligent home system.
Button Trigger: The DF Mini MP3 player is configured to respond to button presses or commands received from external input devices such as IR remotes or sensors. When a button is pressed or a command is received, the DF Mini MP3 player is triggered to play the corresponding audio file stored on the SD card.
Integration with Device Control: The playback of audio prompts by the DF Mini MP3 player is synchronized with the execution of actions or commands within the intelligent home system. For example, when a user presses a button to turn on the lights, the lights are activated, and simultaneously, the DF Mini MP3 player plays the audio prompt "LIGHTS TURNED ON" to provide auditory feedback to the user.

Enhanced User Experience: By incorporating audio feedback through the DF Mini MP3 player, we enhance the user experience and provide intuitive feedback to users regarding the status of their actions or the operation of the intelligent home system. This auditory feedback mechanism improves accessibility and user engagement, especially for individuals with visual impairments or in situations where visual feedback is not practical.

Customization and Expansion: The DF Mini MP3 player allows for the customization of audio prompts and the expansion of functionality to accommodate additional voice commands or feedback messages as needed. Users can record and upload their voice prompts to personalize the system and enhance the overall user experience.

Overall, the DF Mini MP3 player serves as a versatile and user-friendly audio playback device, seamlessly integrating with our intelligent home design to provide auditory feedback and enhance user interaction and engagement. Its integration adds a layer of convenience and accessibility to the automation system, making it more intuitive and user-friendly for occupants of the intelligent home.

5.2.2 Voice Control with DFRobot Gravity Module:

Voice recognition is a computerized technology that identifies and transforms speech signals into editable text or executable commands through analysis. It enables individuals to engage with computers solely by speaking, eliminating the need for traditional input devices. Voice control adds a layer of convenience and hands-free operation to the intelligent home environment.

By integrating the DFrobot Gravity voice recognition module, users can interact with the home automation system using voice commands. This enables users to control devices, activate routines, and perform various tasks simply by speaking commands aloud. Voice control enhances accessibility and allows for natural interaction with the intelligent home, making it an ideal choice for users seeking a more seamless and futuristic control experience.

Fig 5.5: DFRobot Gravity Voice Recognition Module

Here are some Key features of this module:

- **Self-Learning Capability:**
  The voice recognition module features a self-learning function, enabling users to add up to 17 custom command words. Such versatility empowers users to create interactive audio projects with enhanced flexibility and creativity.

- **User-Friendly Interface:**
  The offline speech recognition sensor is designed with user convenience in mind. With 121 pre-installed fixed command words, users are relieved of the task of recording their own voices. For example, in an intelligent window system, when faced with rainfall or thunder, manual intervention to close the window

becomes unnecessary. By recognizing the preset command "close the window," the offline voice recognition module initiates automatic window closure, effectively responding to abrupt weather shifts.

- **Instant Voice Response:**
  The voice recognition module features dual microphones for noise reduction and extended detection range, ensuring accurate performance in noisy environments. It includes built-in and external speaker options for immediate feedback on recognition outcomes. Users can activate the system with a trigger phrase, and the assistant promptly responds to commands, providing real-time feedback.
- **Specifications**:
  The voice recognition module operates within a voltage range of 3.3 to 5V, with a maximum operating current of 370 mA at 5V, supporting communication via I2C/UART protocols and featuring an I2C address of 0x64, providing 121 fixed commands, one fixed wake-up command, and the capability for 17 custom commands, alongside a single learning activation command, boasting an onboard microphone sensitivity of -28db, compactly sized at 49×32 mm (1.93×1.26 inches), and designed to operate within a temperature range of 0 to 70°C.

5.3 Control Logic and Implementation:

In the control logic and implementation section, we detail the specific algorithms and procedures utilized to integrate IR remote control and DFrobot voice recognition modules into the intelligent home system, elucidating how these control mechanisms facilitate user interaction and device management within the automation framework.

In our Intelligent Home design project, we have implemented a sophisticated control mechanism to enhance user interaction and provide flexibility in managing home automation tasks. This control mechanism, governed by the “control” variable, which allows users to seamlessly switch between different modes of operation, including autonomous control or manual control via IR remote, and voice command.

Table 5.2: Categorization of different control system methods.

control=1	Automatic Control: Weather Analysis and Control
control=2	Manual Control1: Infrared (IR) Remote Control:
control=3	Manual Control2: Voice Control with DFRobot Gravity Module

```
control=1;// Automatic Control: Weather Analysis and Control
control=2;//Manual Control1: Infrared (IR) Remote Control:
control=3; //Manual Control2: Voice Control with DFRobot Gravity Module
```

The **“control”** variable serves as the linchpin of this mechanism, enabling the system to adapt its behavior based on user preferences and inputs. By leveraging this variable, users can effortlessly transition between different control modes, each tailored to their specific needs and preferences.

In the following sections, we will explore in detail how the **“control”** variable operates and its role in shaping the functionality of our Intelligent Home system. We will delve into the specific modes of operation enabled by this mechanism, highlighting how users can utilize it to effortlessly manage their home automation tasks with ease and convenience.

5.3.1 Weather Analysis and Control (control = 1):

In this mode, the system operates autonomously, leveraging sensors to analyze environmental parameters like light intensity, precipitation levels, temperature, and humidity.

Based on the analyzed data, the system autonomously controls various home devices such as lights, windows, doors, and fans to maintain desired conditions within the home. For example, if the light intensity decreases below a threshold, the system automatically turns on lights. Similarly, if the temperature rises above a set limit, the system activates fans or air conditioning to maintain a comfortable temperature.

The overall functioning stays the same as what we discussed in **Section 2: Weather Analysis and Control**. This control method is enabled by defaults.

5.3.2 IR Remote Control Mode (control = 2):

When the user presses the power button on the IR remote control, the system transitions to manual remote-control mode (control = 2). In this mode, the user gains direct control over home devices using the IR remote control, overriding the autonomous control of the system.

The user can toggle devices such as lights, windows, doors, and fans on or off using specific buttons on the remote control manually.

For this control mechanism, we have the different buttons configuration for each home appliances along with the voice over notification or audio feedback for device turned on/off. The buttons configuration for IR remote along with its HEX code are given below:

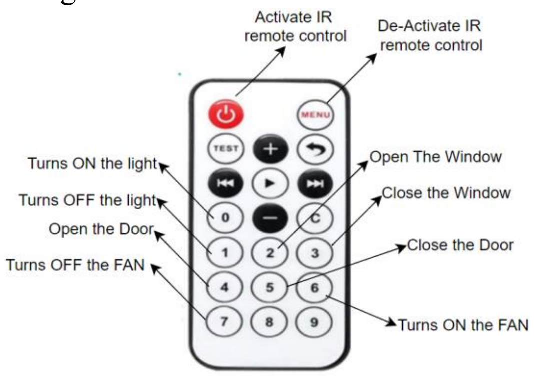
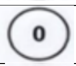
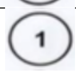
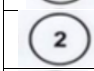
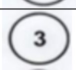
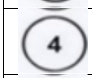
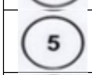
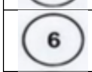
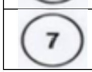


Fig 5.6: IR remote button configurations for controlling home appliances.

Table 5.3: Buttons with their corresponding hex code and function.

BUTTON	HEX CODE	FUNCTION:
	0xFFA25D	Enable IR remote control
	0xFFE21D	Disable IR remote control
	0xFF6897	Turn On the lights
	0xFF30CF	Turn Off the lights
	0xFF18E7	Open the window
	0xFF7A85	Close the window
	0xFF10EF	Open the Door
	0xFF38C7	Close the Door
	0xFF5AA5	Turn On the Fan
	0xFF442BD	Turn Off the Fan

As we have also used the MP3 player for the audio feedback ; the voice recordings for each of the appliance activity are saved with specific numeric file names in a SD card which is mounted on MP3 player. For an example the voice recording for **“Lights turned On”** is saved with file name **“0002”** so that we can play this specific audio whenever the lights is turned on). The file names for each of the saved recordings are given below:

Table 5.4: File names of Audios in SD card and their respective audios.

FILE NAME	AUDIO DESCRIPTION:
002	“Lights turned On”
003	“Lights turned Off”
004	“Window Opened”
005	“Window Closed”
006	“Door Opened”
007	“Door Closed”
008	“FAN turned On”
009	“FAN turned Off”
010	“Remote Control Activated”
011	“Remote Control Deactivated”
012	“Voice Control Activated”
013	“Voice Control De-Activated”

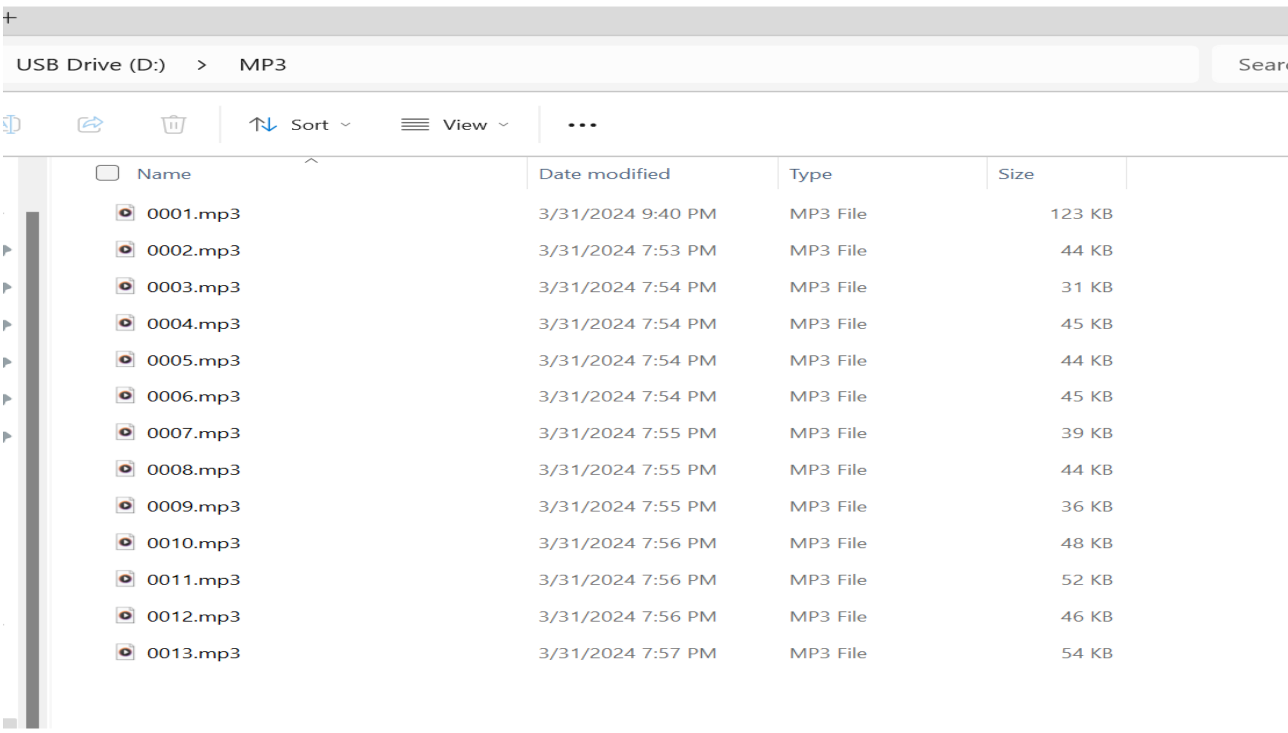


Fig 5.7: File names of the Audio in SD card .

```
if (irrecv.decode(&results)) {
   Serial.println(results.value, HEX);
   if (results.value == 0xFFA25D) //ACTIVATE REMOTE CONTROL with power btn {
    control=2;//remote control
    mp3_play(10);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL1:RMT CTRL");
    lcd.setCursor(0,1);
    lcd.print("ACTIVATED");
    delay(1500);
    lcd.clear();}
   if (results.value == 0xFFE21D) //DEACTIVATE REMOTE CONTROL with MENU btn  {
    control=1;
    mp3_play(11);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL1:RMT CTRL");
    lcd.setCursor(0,1);
    lcd.print("DEACTIVATED");
    delay(1500); }
   irrecv.resume(); // Receive the next value
   if (control==2) {
    switch(results.value){
      case  0xFF6897:  lighton();      mp3_play (2);             break;//btn 0
      case  0xFF30CF:  lightoff();     mp3_play(3);              break;//btn 1
      case  0xFF18E7:  windowopen();   mp3_play(4);              break;//btn 2
      case  0xFF7A85:  windowclose();  mp3_play(5);              break;//btn 3
      case  0xFF10EF:  OpenDoor();     mp3_play(6);              break;//btn 4
      case  0xFF38C7:  CloseDoor();    mp3_play(7);              break;//btn 5
      case  0xFF5AA5:  fanon();        mp3_play(8);             break;//btn 6
      case  0xFF42BD:  fanoff();       mp3_play(9);             break;//btn 7
    } } }
```

This code snippet is part of the main loop of the program and is responsible for handling IR remote control inputs. Here's a breakdown of how it works:

- **IR Reception:**
  The condition **if (irrecv.decode(&results))** checks if there is any incoming IR signal.
- **Remote Control Activation:**
  If the received IR signal corresponds to the power button **(0xFFA25D),** the control variable is set to 2, indicating that manual remote control mode is activated. Additionally, an auditory confirmation is played using the **mp3_play(10)** function, and a message is displayed on **the LCD** indicating that **manual remote control is activated.**
- **Remote Control Deactivation:** If the received IR signal corresponds to the menu button **(0xFFE21D)**, the control variable is set back to **1**, indicating that autonomous control mode is restored. Similarly, an auditory confirmation is played, and a message is displayed on **the LCD** indicating that **manual remote control is deactivated.**
- **IR Signal Processing in Manual Remote-Control Mode:** If the control variable is set to **2** (indicating manual remote-control mode), the program enters a switch-case block to handle different button presses on the IR remote.
- **Action Execution:** Depending on the specific IR signal received (e.g., button 0, button 1, etc.), corresponding actions are executed. For example, pressing button 0 might turn on the lights (**lighton()**

function) and play a specific sound (**mp3_play(2)** function). Similarly, other button presses trigger different actions, such as turning off lights, opening or closing windows, controlling the door, and managing the fan.

- **IR Receive Buffer Clearing:** Finally, **irrecv.resume()** is called to clear the receive buffer and prepare for the next IR signal.

Overall, this code snippet demonstrates how the program switches between autonomous control and manual remote-control modes based on IR signals received, and how it executes different actions depending on the specific IR signals received during manual remote control mode.

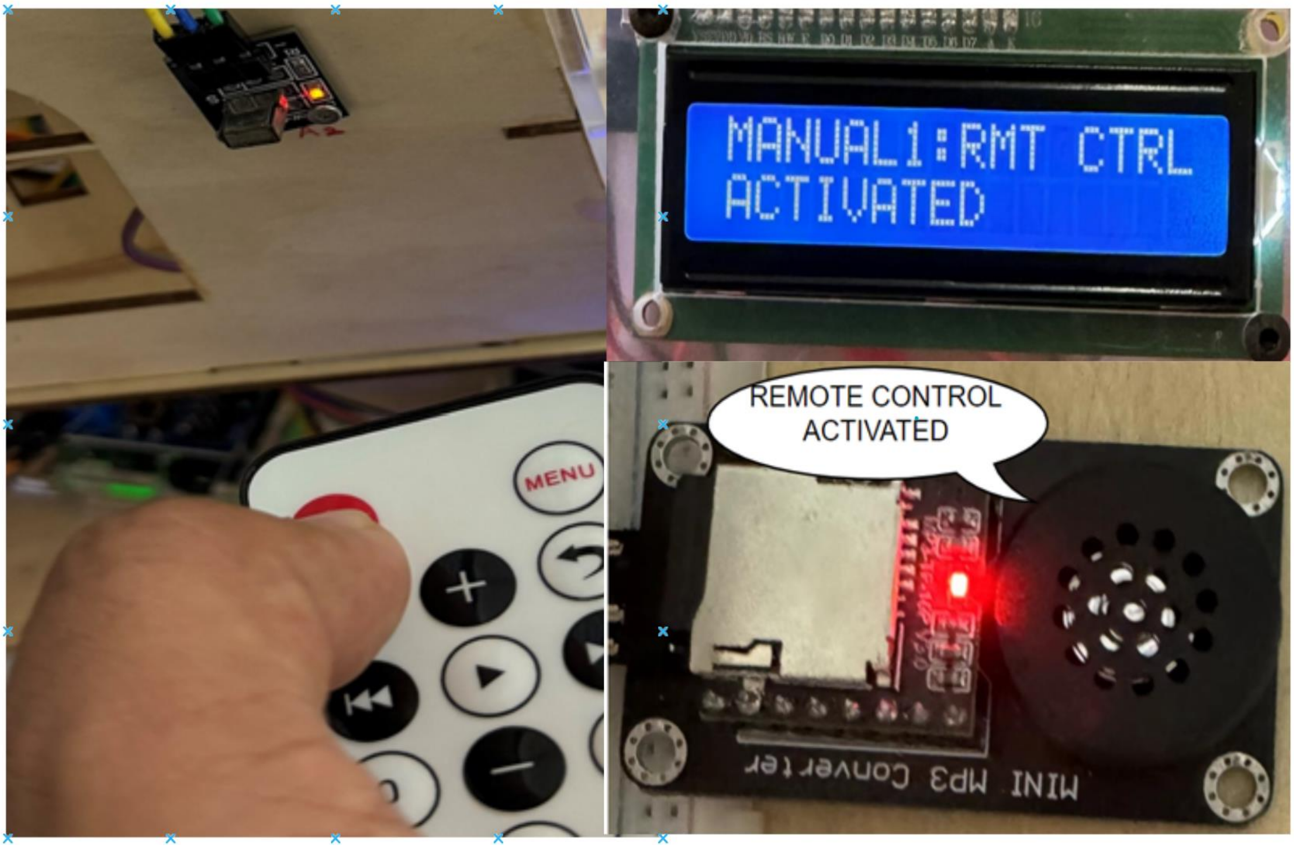
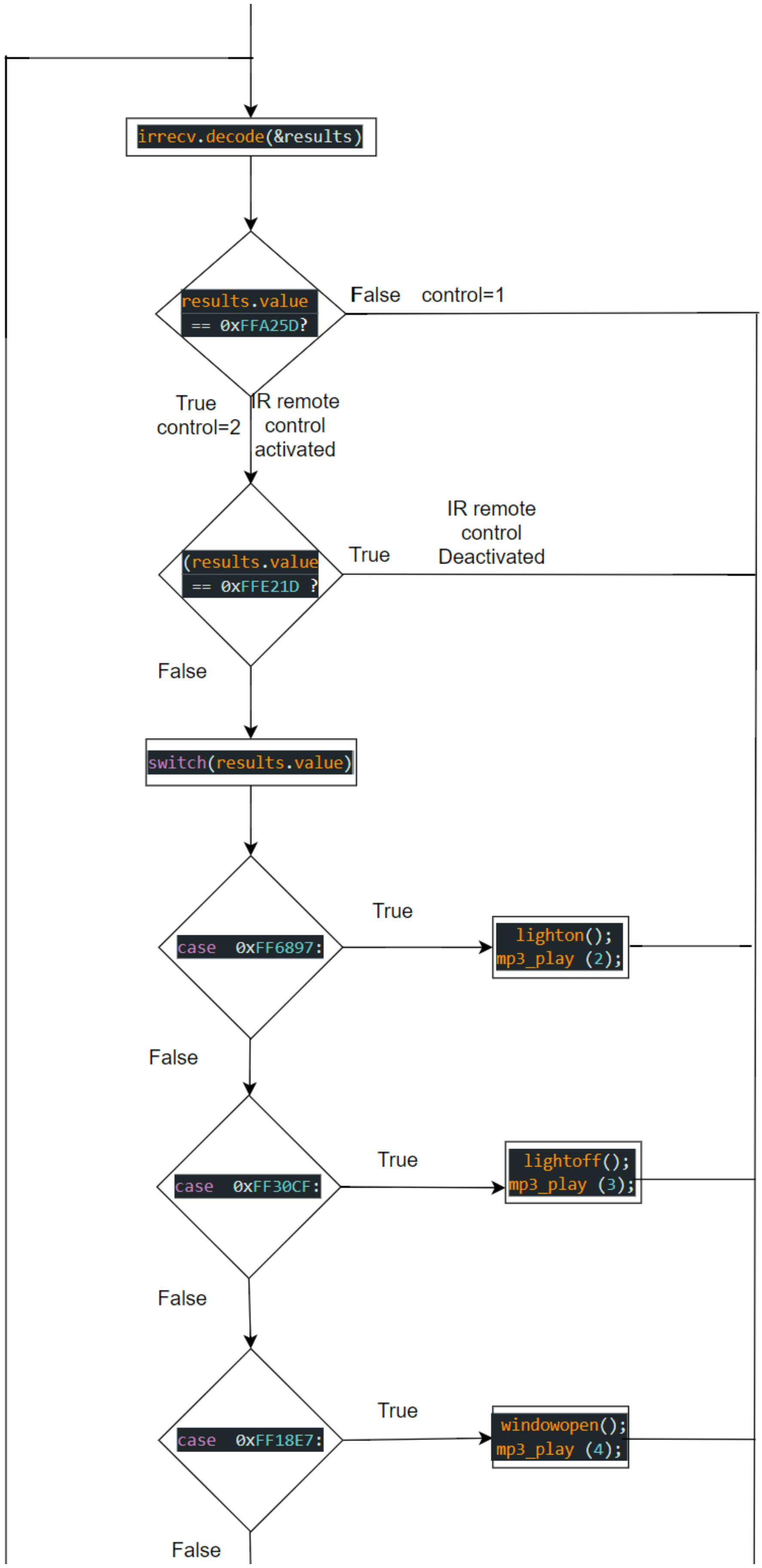


Fig 5.8: Activation of IR remote Control

5.3.3 Overall operation of IR remote control:

(PLEASE TURN OVER)


irrecv.decode(&results)
results.value == 0xFFA25D?
False control=1
True control=2
IR remote control activated
(results.value == 0xFFE21D ?
True
IR remote control Deactivated
False
switch(results.value)
case 0xFF6897:
True
lighton(); mp3_play (2);
False
case 0xFF30CF:
True
lightoff(); mp3_play (3);
False
case 0xFF18E7:
True
windowopen(); mp3_play (4);
False

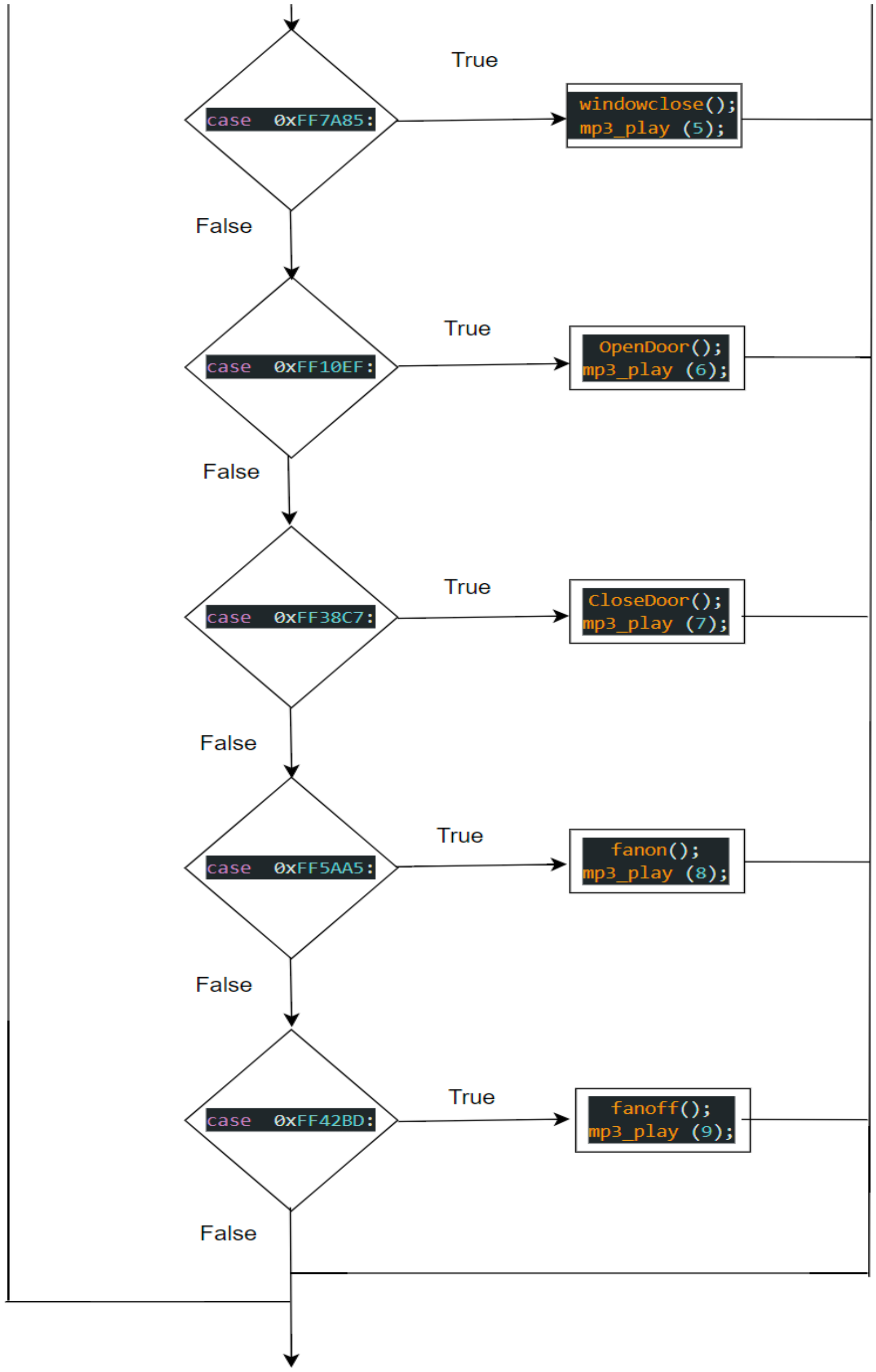


Fig 5.9: Overall operation of IR remote control.

5.3.4 Voice Control Mode (control = 3):

For using DFrobot voice recognition module, we need to be familiar with **wake-up words** and **Commands Words**.

5.3.4.1 Wake-up words:

The term "wake-up word" denotes the specific word or phrase used to transition a device from standby mode to active mode, initiating its functionality. It's akin to uttering "Alexa," "Hey Siri," or "OK Google." In this context, the default wake-up word is "Hello Robot," serving as the initial command to activate the module. Additionally, users have the option to incorporate an alternative wake-up phrase, such as "**Intelligent Home System**."

To add a new wake-up word we need to follow the given procedure:

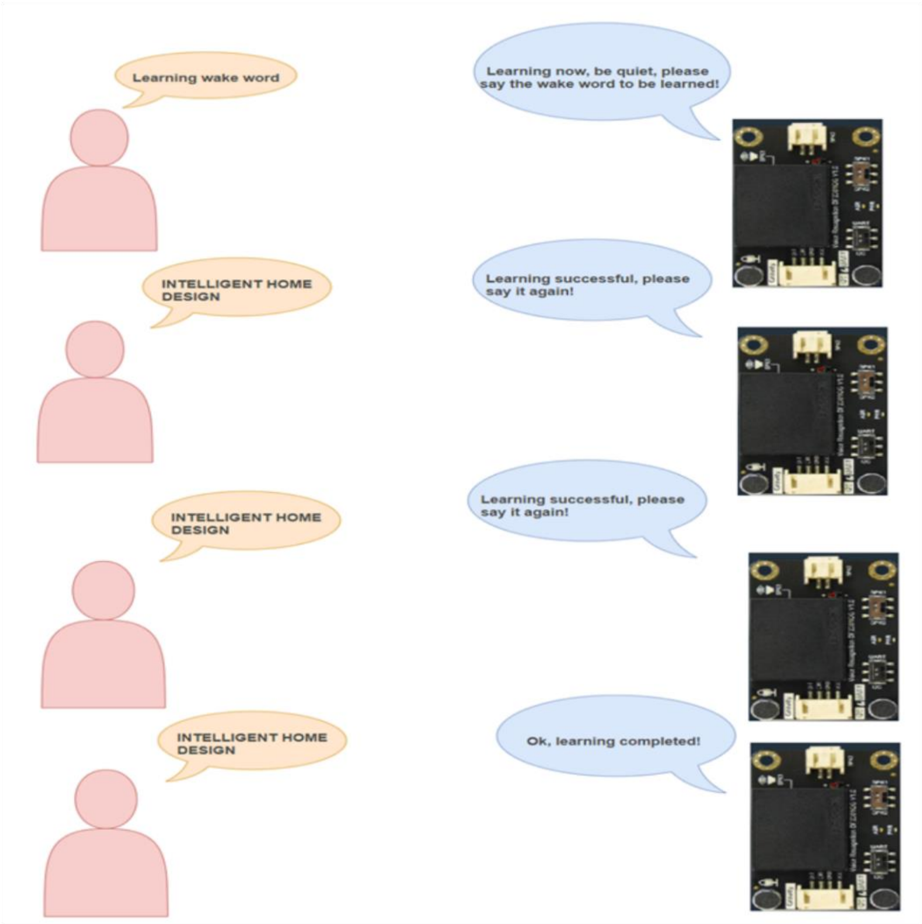


Fig 5.10: New Wake-word Learning Procedure.

After the module says "OK! Learning completed". The new wake word is converted into speech signals and stored in ID 1.

Table 5.5: Wake-up words and their corresponding IDs

Wake-Up Words	**ID**
INTELLIGENT HOME SYSTEM	1
Hello robot	2

5.3.4.2 Command words:

Fixed command words are predefined terms utilized by users to provide precise commands. They are crucial in the context of an offline module, as it operates independently without external internet data or servers. These words have been preprocessed and trained, stored within the module's memory. When activated, they generate corresponding IDs, facilitating the execution of various actions. In total, there are 121 predefined words allocated to specific functions across different IDs. Like Wake-up words we can also add some user defined words by following the same procedure.

(PLEASE TURN OVER)

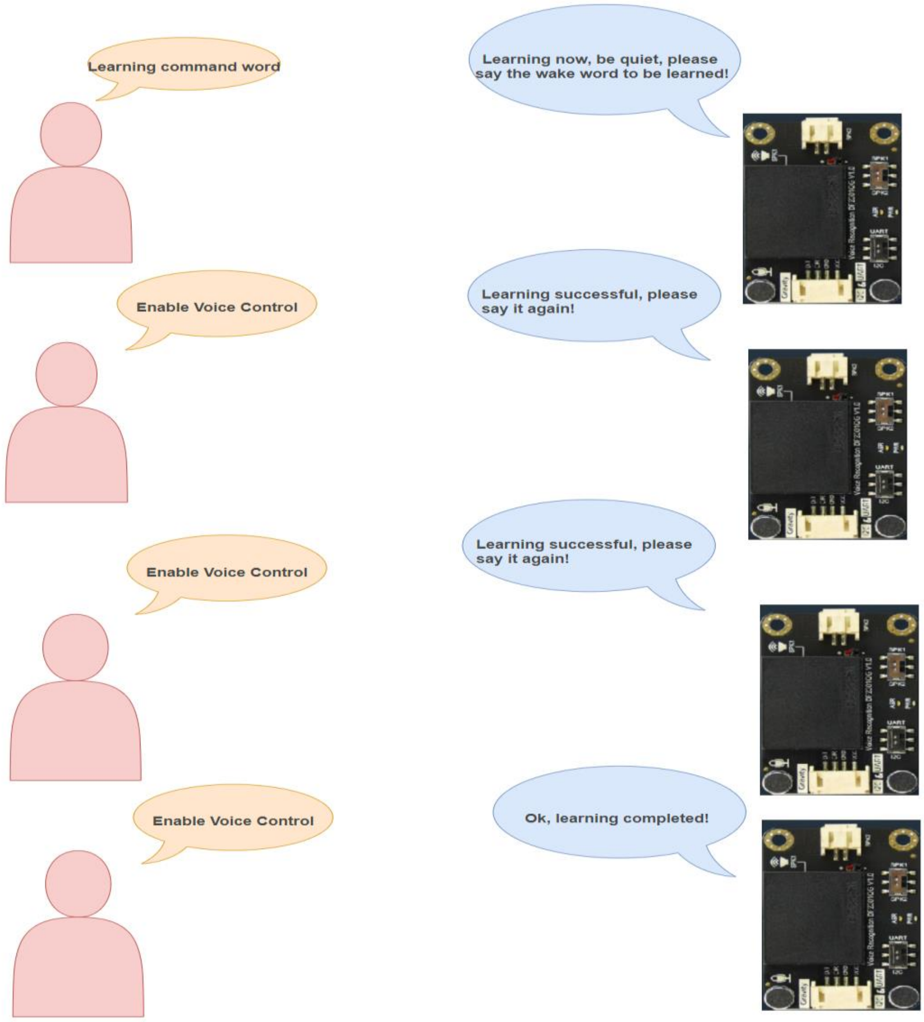


Fig 5.11: New command word Learning Procedure.

After the module says “OK! Learning completed”. The new wake word is converted into speech signals and stored in ID 5 and if the next customized words are added their IDs are added consecutively. We can add up to 17 customized command words.

Table 5.6: Customized command word and their corresponding IDs.

Commands for learning		**ID**	**Commands for learning**	**ID**
The first custom command “Enable Voice Control”		5	The second custom command “Disable Voice Control”	6
……………………………….		…	………………………..	…
………………………………..		…	………………………..	…
…………………………………..		…	……………………………….	…
…………………………………..		…	The seventeenth custom command “…………………”	21

Out of the 121 built-in command words available, we will selectively utilize those relevant to operating household appliances for our project. The relevant commands are highlighted red.

Table 5.7: List of pre-defined Command Words of DFRobot Voice Recognition Module.

Fixed Command Words	ID	Fixed Command Words	ID	Fixed Command Words	ID
Go forward	22	Retreat	23	Park a car	24
Turn left ninety degrees	25	Turn left forty-five degrees	26	Turn left thirty degrees	27
Turn right forty-five degrees	29	Turn right thirty degrees	30	Shift down a gear	31
Line tracking mode	32	Light tracking mode	33	Bluetooth mode	34
Obstacle avoidance mode	35	Face recognition	36	Object tracking	37
Object recognition	38	Line tracking	39	Color recognition	40
Tag recognition	41	Object sorting	42	Qr code recognition	43
General settings	44	Clear screen	45	Learn once	46
Forget	47	Load model	48	Save model	49
Take photos and save them	50	Save and return	51	Display number zero	52
Display number one	53	Display number two	54	Display number three	55
Display number four	56	Display number five	57	Display number six	58
Display number seven	59	Display number eight	60	Display number nine	61
Display smiley face	62	Display crying face	63	Display heart	64
Turn off dot matrix	65	Read current posture	66	Read ambient light	67
Read compass	68	Read temperature	69	Read acceleration	70
Reading sound intensity	71	Calibrate electronic gyroscope	72	Turn on the camera	73
Turn off the camera	74	Turn on the fan	75	Turn off the fan	76
Turn fan speed to gear one	77	Turn fan speed to gear two	78	Turn fan speed to gear three	79
Start oscillating	80	Stop oscillating	81	Reset	82
Set servo to ten degrees	83	Set servo to thirty degrees	84	Set servo to forty-five degrees	85
Set servo to sixty degrees	86	Set servo to ninety degrees	87	Turn on the buzzer	88
Turn off the buzzer	89	Turn on the speaker	90	Turn off the speaker	91
Play music	92	Stop playing	93	The last track	94
The next track	95	Repeat this track	96	Volume up	97
Volume down	98	Change volume to maximum	99	Change volume to minimum	100
Change volume to medium	101	Play poem	102	Turn on the light	103
Turn off the light	104	Brighten the light	105	Dim the light	106
Adjust brightness to maximum	107	Adjust brightness to minimum	108	Increase color temperature	109
Decrease color temperature	110	Adjust color temperature to maximum	111	Adjust color temperature to minimum	112
Daylight mode	113	Moonlight mode	114	Color mode	115
Set to red	116	Set to orange	117	Set to yellow	118
Set to green	119	Set to cyan	120	Set to blue	121
Set to purple	122	Set to white	123	Turn on ac	124
Turn off ac	125	Increase temperature	126	Decrease temperature	127
Cool mode	128	Heat mode	129	Auto mode	130
Dry mode	131	Fan mode	132	Enable blowing up & down	133
Disable blowing up & down	134	Enable blowing right & left	135	Disable blowing right & left	136
Open the window	137	Close the window	138	Open curtain	139
Close curtain	140	Open the door	141	Close the door	142

To Enable voice control, the user requires to turn the voice recognition module into active mode by reciting the activation word or wake-up word as **“INTELLIGENT HOME DESIGN”** and then recite **“Enable voice control ”** to make the system transitions to manual Voice control mode (control = 3). In this mode, the user gains direct control over home devices using voice command, overriding the autonomous control of the system. The user can toggle devices such as lights, windows, doors, and fans on or off using specific voice commands to control manually.

```
CMDID = DF2301Q.getCMDID();
  if(0 != CMDID) {
    Serial.print("CMDID = ");
    Serial.println(CMDID);
  }
  if (CMDID==5)//ACTIVATE VOICE CONTROL {
    control=3;
    mp3_play(12);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL2:VCE CTRL");
    lcd.setCursor(0,1);
    lcd.print("ACTIVATED");
    delay(1500);
    lcd.clear();}
  if (CMDID==6)//DEACTIVATE VOICE CONTROL{
    control=1;
    mp3_play(13);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL2:VCE CTRL");
    lcd.setCursor(0,1);
    lcd.print("DEACTIVATED");
    delay(1500);
    lcd.clear();}
  if(control==3)
  {
  switch (CMDID)
  {
    case 103:     lighton();                mp3_play(2);                break;
    case 104:     lightoff();               mp3_play(3);                break;
    case 137:     windowopen();                                         break;
    case 138:     windowclose();                                        break;
    case 141:     OpenDoor();                                           break;
    case 142:     CloseDoor();                                          break;
    case 75:      fanon();                                              break;
    case 76:      fanoff();                                             break;

  }

}
```

This code snippet demonstrates the integration of the DFRobot Gravity voice recognition module (DF2301Q) within our overall project. Here's a breakdown of how it works:

- **Get Voice Command ID:** The CMDID = DF2301Q.getCMDID(); line retrieves the ID of the recognized voice command using a function from the DFRobot Gravity module's library. If a command is recognized (CMDID is not zero), the code prints the ID for debugging.
- **Activate/Deactivate Voice Control:**
  The code checks the value of CMDID. If CMDID is 5, it sets the control variable to 3 (indicating activation of voice control) and plays an audio file. If CMDID is 6, it sets the control variable to 1 (indicating deactivation) and plays another audio file.
- **Execute Actions based on Voice Commands (when Voice Control is Active):**
  If voice control is active (control is 3), the code checks the value of CMDID again. This time, it likely corresponds to recognized voice commands. Based on CMDID, different functions like lighton(), lightoff(), windowopen(), etc., are executed.

Overall, this code segment allows for the dynamic control of an intelligent home system based on voice commands, enabling users to activate or deactivate voice control mode and execute various actions through voice commands.

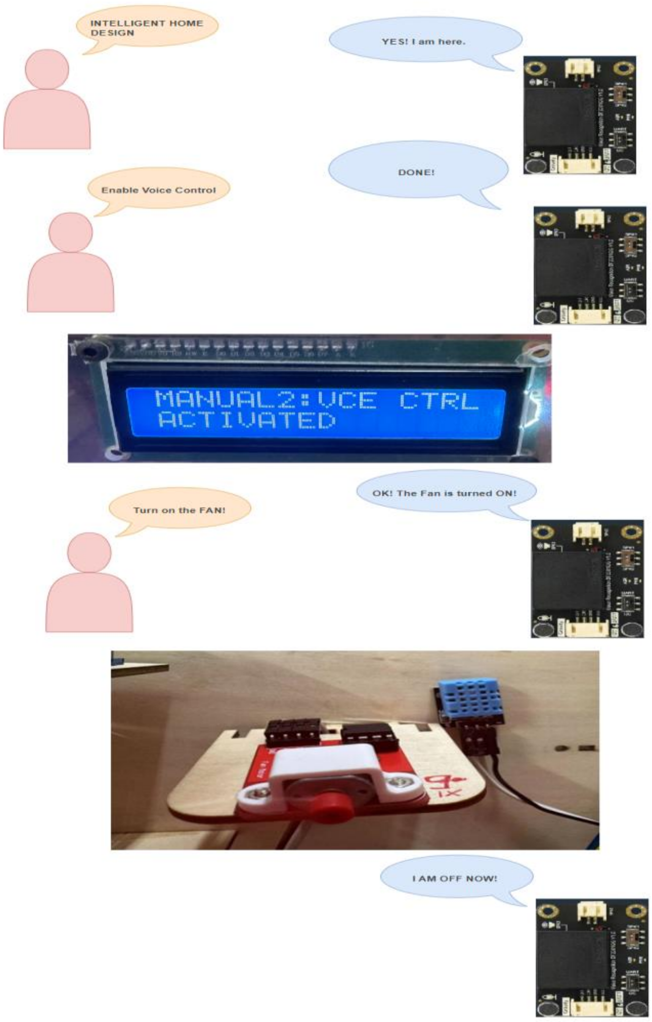
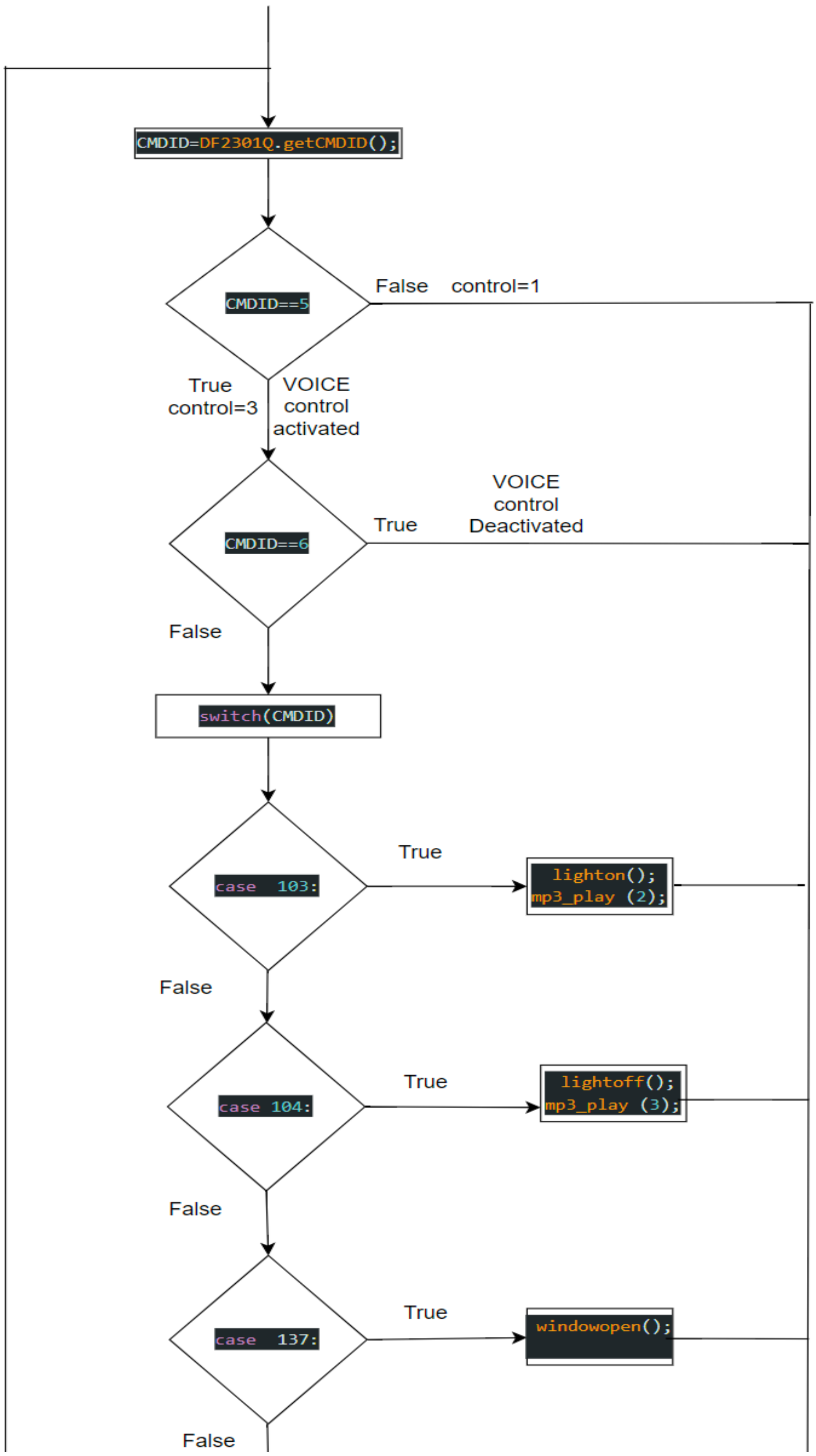


Fig 5.12: Voice Control Activation to control the Intelligent Home System

5.3.5 Overall operation of Voice Control System:

(PLEASE TURN OVER)


CMDID=DF2301Q.getCMDID();
CMDID==5
False control=1
True control=3
VOICE control activated
CMDID==6
True
VOICE control Deactivated
False
switch(CMDID)
case 103:
True
lighton();
mp3_play (2);
False
case 104:
True
lightoff();
mp3_play (3);
False
case 137:
True
windowopen();
False

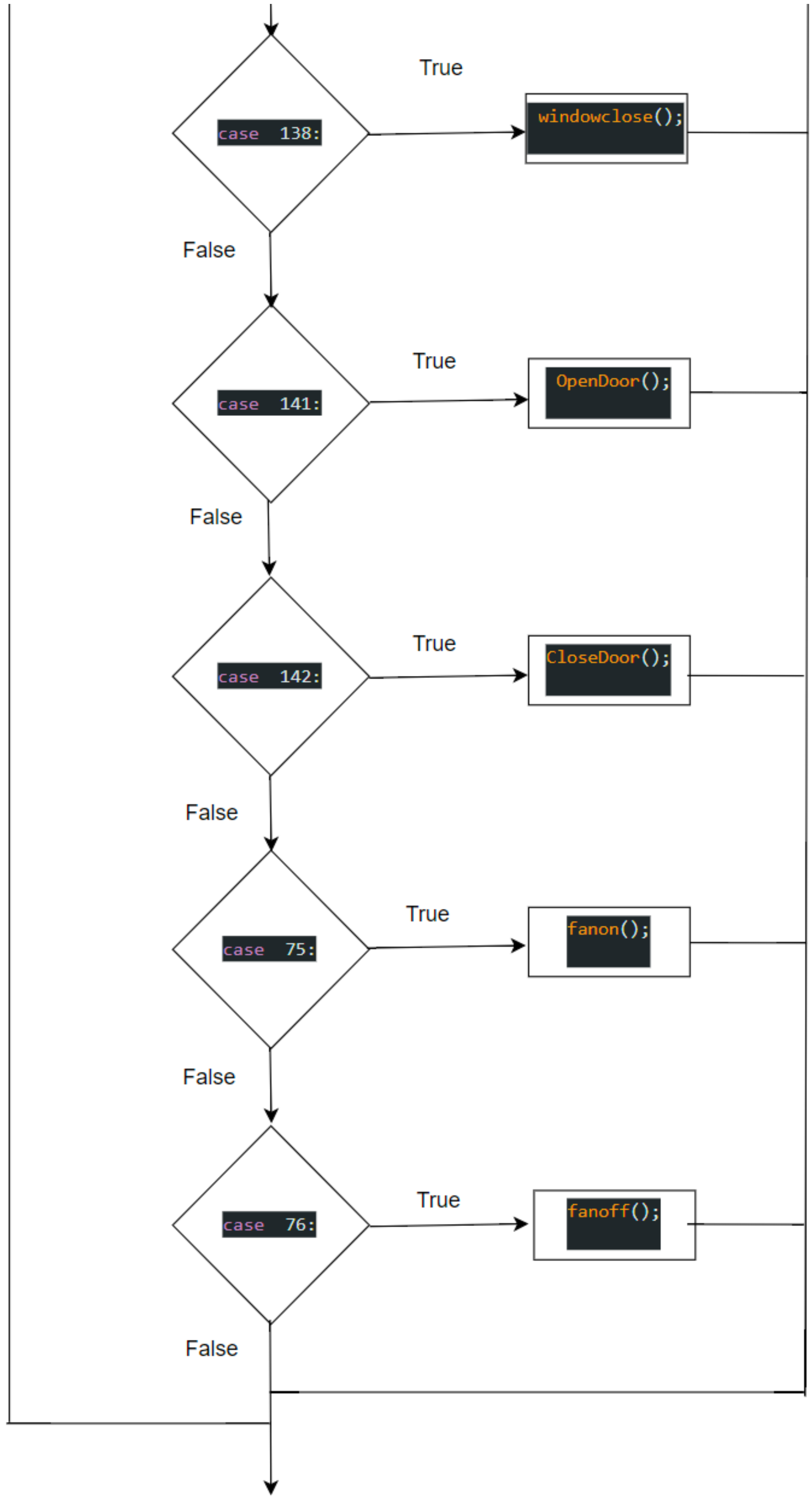


Fig 5.13: Overall operation of Voice Control System

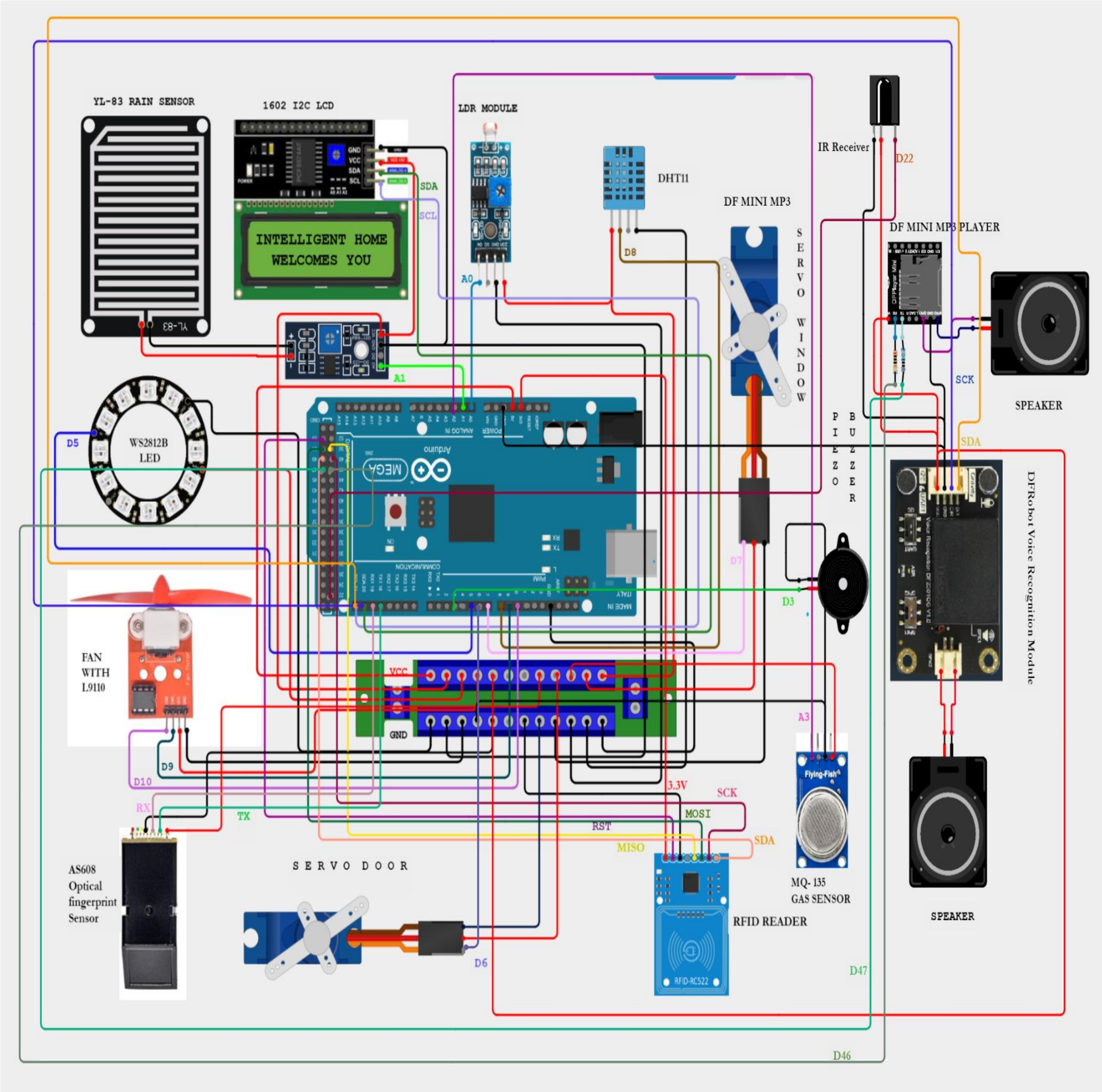


Fig 5.14: Circuit Diagram for Control System

5.4 Testing and Validation:

Testing and validation are crucial phases in the development of any system, including a home automation setup. These processes ensure that the system functions as intended, meets user requirements, and operates reliably under various conditions.Testing involves executing predefined procedures to identify defects, errors, or discrepancies in the system's behavior. It encompasses different types of tests, such as functional testing, performance testing, and integration testing, each focusing on specific aspects of system functionality.

Validation, on the other hand, aims to confirm that the system meets user needs and requirements effectively. It involves assessing the system's overall performance, usability, and compliance with specifications and standards.Together, testing and validation ensure that the home automation system functions correctly, provides a positive user experience, and meets the desired objectives of convenience, efficiency, and safety.

5.4.1 Sensor Integration Testing and Validation:

Sensor integration testing is a critical aspect of validating the functionality and performance of sensors within a home automation system. This testing phase focuses on ensuring that sensors accurately detect and report environmental data, such as temperature, humidity, light levels, and precipitation. By subjecting sensors to various environmental conditions and scenarios, the effectiveness of their integration into the overall system can be evaluated. This process helps verify that sensors provide reliable data inputs for decision-making and control actions, ultimately contributing to the system's overall functionality and user experience. Sensor integration validation ensures that the sensors used in the home automation system accurately detect environmental parameters and trigger appropriate actions. Here's how it can be conducted:

- **Sensor Calibration:**
  a) Sensors such as temperature, humidity, and rain sensors are calibrated to ensure accurate readings.
  b) Calibration involves adjusting sensor parameters to match known reference values or standards.
  c) Calibration curves or correction factors may be established to compensate for sensor drift or inaccuracies.

- **Environmental Testing:**
  a. Sensors are exposed to various environmental conditions (e.g., different light levels, temperature, humidity) to validate their performance across different scenarios.
  b. Temperature sensors are tested in environments with stable temperatures and subjected to temperature changes to verify responsiveness.
  c. Humidity sensors are tested in different humidity levels, from dry to humid conditions, to assess accuracy across the entire range.
  d. Light sensors are tested under different lighting conditions, including low light, bright sunlight, and artificial lighting, to evaluate sensitivity and response.

- **Data Validation:**
  a. Sensor data is compared against reference measurements or standards to validate accuracy.
  b. Data consistency and integrity are verified by analyzing trends, patterns, and correlations between different sensor readings.
  c. Anomalies or outliers in sensor data are identified and investigated to determine their cause and address any issues with sensor performance.

- **Fault Tolerance Testing:**
  a. The system's response to sensor failures or malfunctions is evaluated to assess fault tolerance and resilience.
  b. Redundancy mechanisms or backup sensors may be activated to maintain essential functions in case of sensor failure.
  c. Failure recovery procedures, such as sensor recalibration or automatic fault detection, are tested to ensure timely mitigation of issues.
- **User Feedback Collection:**
  a. Users interact with the system and provide feedback on sensor accuracy, responsiveness, and reliability.
  b. Feedback may include observations of sensor behavior under different conditions and suggestions for improvement.
  c. User feedback is valuable for identifying areas of improvement and optimizing sensor integration for enhanced user experience.

By conducting thorough sensor integration validation, the reliability and effectiveness of the home automation system in accurately sensing and responding to environmental conditions can be confirmed, ensuring optimal performance and user satisfaction.

5.4.2 IR Remote Control Testing and Validation:

Assess the IR receiver's capability to detect signals from the remote control across varied distances and angles to ensure consistent signal reception.

- **Button Functionality Testing:**
  Verify each button on the remote control to ensure it triggers the intended actions within the home automation system.
- **Signal Interference Testing:**
  Evaluate the IR remote control's performance in environments prone to signal interference, such as the presence of other IR devices or exposure to sunlight, to gauge its resilience against interference.
- **Battery Life Testing:**
  Measure the battery life of the IR remote control under typical usage conditions to ensure prolonged operation without frequent battery replacements.

Validation Procedures:

- **Functionality Verification:**
  Validate the IR remote control's ability to initiate actions like adjusting settings, activating/deactivating features, etc., within the home automation system.
- **Compatibility Testing:**
  Verify compatibility with various devices and subsystems in the home automation setup to ensure smooth integration and control.
- **User Satisfaction:**
  Gather user feedback on aspects such as ease of use, responsiveness, and overall satisfaction with the IR remote control to confirm its effectiveness in enhancing user experience.
- **Reliability Assessment:**
  Assess the IR remote control's reliability in consistently transmitting signals and initiating actions without errors or delays through extended usage and stress testing.

Testing and validation ensure that the IR remote control meets performance standards, seamlessly integrates with the home automation system, and satisfies user expectations for functionality, compatibility, and reliability.

5.4.3 Voice Recognition Testing and Validation:

To ensure the reliability and accuracy of voice recognition, various testing procedures are conducted:

- **Accuracy Testing and Command Recognition:**
  a) Predefined voice commands are spoken into the system multiple times to verify consistent recognition.
  b) The system's ability to accurately interpret and execute commands is assessed by comparing the recognized commands with the intended ones.
  c) The accuracy percentage is calculated based on the number of correctly recognized commands out of the total attempts.
- **Noise Tolerance Testing:**
  a) Voice commands are tested in different noise environments to assess the system's ability to filter out background noise.
  b) Various types of noise, such as chatter, music, or ambient sounds, are introduced to simulate real-world scenarios.

c) The system's performance in noisy environments is compared against its performance in quiet conditions to assess noise tolerance.

- **Response Time Evaluation:**
  a) The time taken by the system to respond to voice commands is measured and analyzed.
  b) Response times for different commands and under various conditions (e.g., system load, latency) are recorded.
  c) Response time benchmarks are established based on user expectations and system requirements.
- **False Positive/Negative Analysis:**
  a) Instances of false positives (incorrectly recognized commands) and false negatives (missed commands) are documented and analyzed.
  b) Patterns or common causes of false recognition are identified and addressed to improve accuracy.
  c) Adjustments to voice recognition thresholds or algorithms may be made based on the analysis.
- **User Feedback Collection:**
  a) Users interact with the voice recognition system and provide feedback on their experience.
  b) Feedback may include perceptions of accuracy, ease of use, and overall satisfaction with the voice interface.
  c) Surveys, interviews, or usability tests may be conducted to gather qualitative feedback from users.
- **Validation against Use Cases:**
  a) The voice recognition system is validated against specific use cases or scenarios defined during the system design phase.
  b) For example, commands related to controlling lights, adjusting thermostat settings, or playing music are tested to ensure functionality and accuracy in real-world applications.
- **Continuous Improvement:**
  a) Ongoing monitoring and feedback collection are essential for continuously improving the voice recognition system.
  b) Updates and enhancements, such as improved algorithms or expanded vocabulary support, may be implemented based on user feedback and performance metrics.

The voice recognition system consistently recognizes predefined commands with high accuracy (>95%) and responds within acceptable response times (typically <1 second) if used inside distance radius as 3-4 meters. By conducting comprehensive validation procedures, the effectiveness and reliability of the voice recognition system in the home automation project can be confirmed, ensuring a positive user experience and seamless integration with other control interfaces.

6. Conclusion

6.1 Weather Analysis and Control:

- **Key Findings:**
  Effective weather analysis enables the home automation system to respond proactively to changing environmental conditions, optimizing energy usage and enhancing user comfort.
- **Contribution:**
  Integration of weather analysis enhances the efficiency and adaptability of home automation systems, improving energy conservation and user satisfaction.
- **Future Directions:**
  Further research could explore advanced machine learning algorithms for more accurate weather prediction and optimization strategies for energy management based on forecasted weather patterns.

6.2 Security System:

- **Key Findings:**
  Robust security systems provide peace of mind by detecting and mitigating potential threats, enhancing home safety and security.
- **Contribution:**
  Implementation of comprehensive security measures improves the overall safety and protection of residents and their property.
- **Future Directions:**
  Future research could focus on integrating emerging technologies such as AI-driven surveillance and facial recognition for more advanced threat detection and prevention.

6.3 Hazard Detection:

- **Key Findings:**
  Hazard detection systems play a crucial role in identifying and mitigating potential risks, safeguarding occupants and property from harm.
- **Contribution:**
  Integration of hazard detection technologies enhances the safety and resilience of home automation systems, minimizing the impact of emergencies and accidents.
- **Future Directions:**

Further research could explore the integration of IoT devices and real-time monitoring systems for more accurate and timely hazard detection, as well as the development of automated response mechanisms for rapid mitigation of hazards.

6.4 Control Design:

- **Key Findings:**
  Well-designed control interfaces facilitate intuitive interaction and management of home automation systems, enhancing user experience and convenience.
- **Contribution:**
  Implementation of user-friendly control designs improves accessibility and usability, empowering users to easily monitor and control various aspects of their home environment.
- **Future Directions:**
  Future research could focus on the development of personalized and adaptive control interfaces tailored to individual user preferences and behaviors, as well as the exploration of immersive technologies such as augmented reality for enhanced control and visualization of home automation systems.

References:

[1]. Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G., ... & Gavras, A. (2014, March). Enabling reliable and secure IoT-based smart city applications. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS) (pp. 111-116). IEEE.

[2]. Ma, Y., Yang, S., Huang, Z., Hou, Y., Cui, L., & Yang, D. (2014, December). Hierarchical air quality monitoring system design. In 2014 International Symposium on Integrated Circuits (ISIC) (pp. 284-287). IEEE.

[3]. Marques, G., & Pitarma, R. (2016). An indoor monitoring system for ambient assisted living based on internet of things architecture. International journal of environmental research and public health, 13(11), 1152.

[4]. S. Chitnis, N. Deshpande and A. Shaligram, “An investigative study for smart home security: issues, challenges and countermeasures,” Wireless Sensor Network, vol. 8, no. 04, pp. 61-68, 2016.

[5]. A. Pnevmatikakis, “Recognising Daily Functioning Activities in Smart Homes,” Wireless Personal Communications, vol. 96, no. 3, pp. 3639-3654, 2017

[6]. A. C. Jose and R. Malekian, “Improving Smart Home Security: Integrating Logical Sensing Into Smart Home,” IEEE Sensors Journal, vol. 17, no. 13, pp. 4269-4286, 2017.

[7].Shailin Desai Sapan Khanna Home Automation System Volume -2(October -2013). International Journal of Engineering Research & Technology (IJERT)

[8]. D. Greaves, "Control Software for Home Automation, Design Aspects and Position Paper", The AutoHan project at the University of Cambridge Computer Laboratory

[9]. Satish Palaniappan, Naveen Hariharan ,etall , “Home Automation Systems - A Study”, International Journal of Computer Applications, Volume 116 – No. 11, April 2015

[10]. Qela B, Mouftah HT. Observe, Learn, and Adapt (OLA) - An algorithm for energy management in smart homes using wireless sensors and artificial intelligence. IEEE Transactions on Smart Grid. 2012 Sep; 3(4):2262–72.

[11]. EA, “Fact Sheet: Standby Power Use and the IEA “1 Watt Plan”,” Apr.2007. http://www.iea.org/ textbase/papers/2007/standby_fact.pdf

[12]. Calboun B.H., Chandrakasan A.P., “Standby Power Reduction Using Scaling and Carry Flip - Flop Structures,” IEEE Journal of Solid-State Circuits, vol.39, no.9, pp.1504-1511, Sep. 2004

[13]. Armando Roy Delgado, Rich Picking and Vic Grout “Remote- Controlled Home Automation Systems with Different Network Technologies” Centre for Applied Internet Research (CAIR), University of Wales, NEWI, Wrexham, UK.

[14]. Folea, S. ; Autom. Dept., Tech. Univ. of Cluj-Napoca, Cluj-Napoca, Romania ; Bordencea, D. ; Hotea, C. ; Valean, H – “Smart home automation system using Wi-Fi low power devices” Published in: Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on Date of Conference: 24- 27 May 2012 Page(s): 569 – 574 Print ISBN:978-1- 4673-0701-7 ; INSPEC Accession Number: 12853582 Conference Location :Cluj-Napoca ; DOI:10.1109/AQTR.2012.6237775 ; Publisher: IEEE

[15]. Obaid, T. et al. 2014. "ZIGBEE BASED VOICE CONTROLLED WIRELESS SMART HOME SYSTEM", International Journal of Wireless & Mobile Networks (IJWMN), Vol. 6, No. 1, Pg. 47-5

[16]. San Hlaing, N. N., & San Lwin, S. (2019). Electronic Door Lock using RFID and Password Based on Arduino. International Journal of Trend in Scientific Research and Development, 3(2), 799-802.

[17]. Zhang, L., "An Improved Approach to Security and Privacy of RFID application System", Wireless Communications, Networking and Mobile Computing. International Conference. pp 1195-1198, 2005.

[18]. K. Karu,A.K. Jain, Fingerprint classification, PatternRecognition 29(3)(1996) 389-404

[19]. D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted MMI for model and feature-space discriminative training. In Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2008.

[20]. https://www.dfrobot.com/product-2665.html

[21]. https://github.com/s60sc/ESP32-CAM_MJPEG2SD

Acknowledgments:

I extend my hearty gratitude to my supervisor, **Associate Professor Wang Yingpin**, for graciously allowing me to conduct my thesis in his lab for instrumentation. His unwavering support, valuable insights, and regular check-in on my work have been beneficial throughout this journey. With his guidance and mentorship, I have been able to navigate challenges, stay focused, and ultimately complete my thesis on time. I am sincerely grateful for his patience, encouragement, and dedication to my academic growth.

Additionally, I would like to acknowledge the importance of providing the necessary components to conduct my experiments. Without access to these essential tools and materials, my research endeavors would not have been possible. These components enabled me to design and execute experiments, collect valuable data, and draw meaningful conclusions. Their availability ensured the success and efficiency of my research efforts, and for that, I am truly thankful.

Appendices:

THE FINAL PRODUCT:

The program files used in this project are as accordance to the provided screenshot:

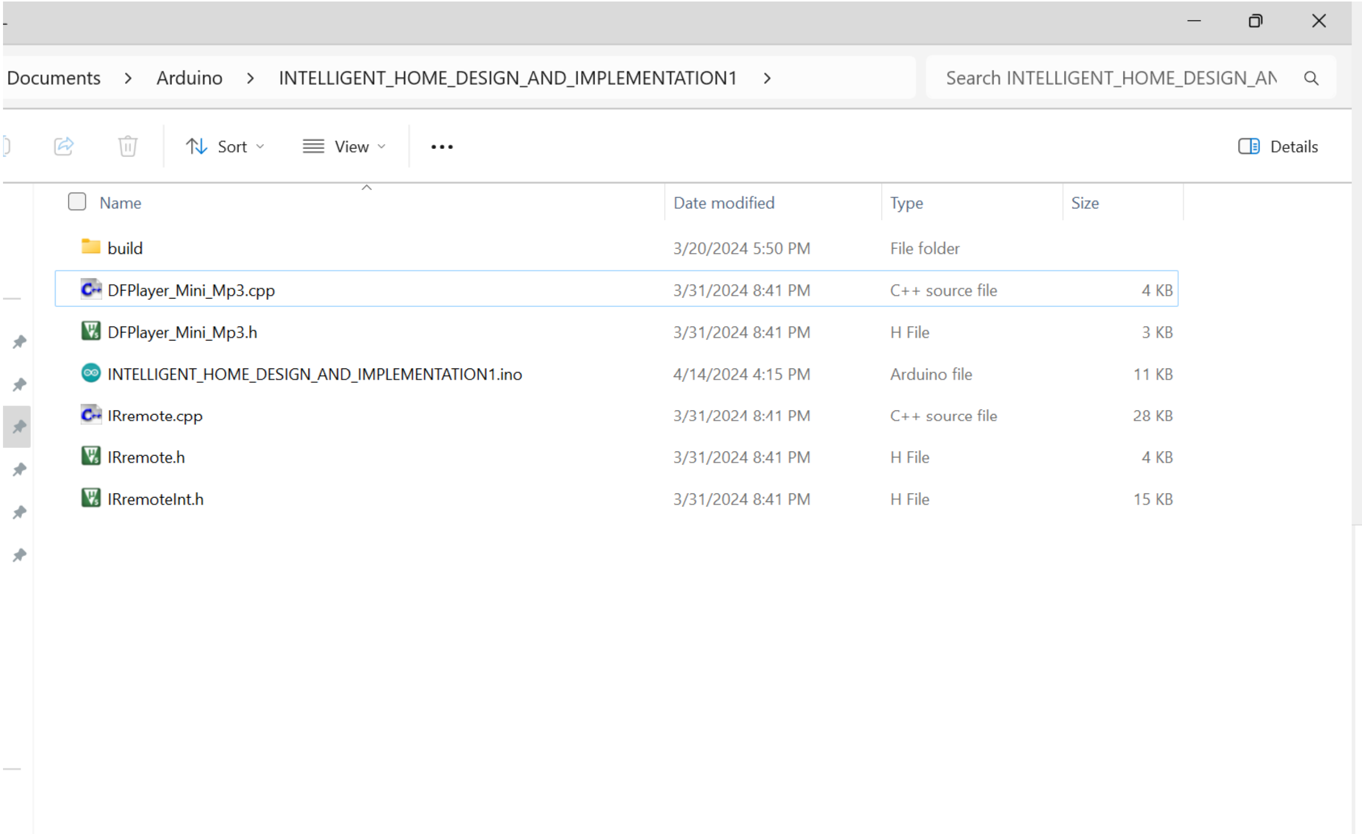


Fig : Program files for this project.

INTELLIGENT_HOME_DESIGN_AND_IMPLEMENTATION1.ino

```
// Library functions call
#include<Servo.h>
#include<DHT.h>    //DHT11 Tempandhumid
#include<Wire.h>  //I2C LCD
#include<LiquidCrystal_I2C.h>  //LCD
#include <Adafruit_NeoPixel.h>  //LED
#include <Adafruit_Fingerprint.h>
#include <SPI.h> // RFID
#include <MFRC522.h> //RFID
#include <SoftwareSerial.h>
#include "DFPlayer_Mini_Mp3.h"
#include "IRremote.h"
#include "DFRobot_DF2301Q.h"

// I/O pins declarations

const int LDR= A0;
const int rain= A7;
const int Air= A2;
const int buzzer=3;
const int led=5;
const int door=6;
const int window=7;
const int dht=8;
```

```
const int IN1=9;
const int IN2=10;
const int mp3rx=46;
const int mp3tx=47;
const int rst=48;
const int spi=49;
const int IrPin =22;

// variable declarations:
int light;
int precipation;
int air;
float humid,tempC;
int windowcloseangle=10;
int windowopenangle=175;
int doorcloseangle=10;
int dooropenangle=175;
int noleds=8;
int control;
byte knownID1[] = {0xF3, 0x1D, 0x68, 0xA6}; // TFID CARD 1
byte knownID2[] = {0x2A, 0xE3, 0x0F, 0x0B}; // TFID TAG  2
uint8_t CMDID ;// variable for speech recognition inputs


// Object declarations:
Servo Window;
Servo Door;
#define Type DHT11
DHT  ht(dht,Type);
LiquidCrystal_I2C lcd(0x27,16,2);
Adafruit_NeoPixel LED(noleds, led, NEO_GRB + NEO_KHZ800);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);
MFRC522 mfrc522(spi, rst);
SoftwareSerial mySerial(mp3rx, mp3tx);
IRrecv irrecv(IrPin);   //IR sensor
decode_results results;
DFRobot_DF2301Q_I2C DF2301Q;

//function declarations:
void lighton(void);
void lightoff(void);
void windowopen(void);
void windowclose(void);
void fanon(void);
void fanoff(void);
void OpenDoor(void);
void CloseDoor(void);
bool checkID1(byte[],byte[]);
bool checkID2(byte[],byte[]);
void firehazards(void);
void rainbowBlink(uint16_t, uint16_t );


void setup()
```

```
{
  Serial.begin(9600);
  control=1;          // Automatic control by sensors
  digitalWrite(buzzer,HIGH);
  pinMode(LDR,INPUT);
  LED.begin();
  LED.setBrightness(255);
  pinMode(rain,INPUT);
  Window.attach(window);
  Door.attach(door);
  ht.begin();
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(air,INPUT);
  pinMode(buzzer,OUTPUT);
  lcd.begin(16,2);
  lcd.init();
  lcd.backlight();
  finger.begin(57600);
  finger.getParameters();
  SPI.begin();        // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522
  mySerial.begin (9600);
  mp3_set_serial (mySerial);
  mp3_set_volume (30);
  mp3_play ();
  delay(500);
  irrecv.enableIRIn();
  mp3_play (1);

  //Check if the DF ROBOT voice module is working or not.
  // while( !( DF2301Q.begin() ) ) {
  //   Serial.println("Communication with device failed, please check connection");
  //   delay(1000);
  // }
  DF2301Q.setVolume(7);
  DF2301Q.setMuteMode(0);
  DF2301Q.setWakeTime(15);
  Serial.println("Begin ok!");
  uint8_t wakeTime = 0;
  wakeTime = DF2301Q.getWakeTime();
  Serial.print("wakeTime = ");
  Serial.println(wakeTime);
  DF2301Q.playByCMDID(23);

  lcd.setCursor(0,0);
  lcd.print("INTELLIGENT HOME");
  lcd.setCursor(1,1);
  lcd.print(" WELCOMES YOU");
  rainbowBlink(1000, 1000);
  delay(4500);
  lcd.clear();

}
```

```
void loop()
{
  light=analogRead(LDR);
  precipation=analogRead(rain);
  humid=ht.readHumidity();
  tempC=ht.readTemperature();
  air=analogRead(Air);
  Serial.print("LDR VALUE: ") ;
  Serial.println(light);
  Serial.print("rain VALUE: ");
  Serial.println(precipation);
  Serial.print("TEMP in C= ");
  Serial.println(tempC);
  Serial.print("Humidity= ");
  Serial.println(humid);
  lcd.setCursor(0,0);
  lcd.print("TEMP=  ");
  lcd.print(tempC);
  lcd.setCursor(0,1);
  lcd.print("HUMI=  ");
  lcd.print(humid);

  if (control==1)// WEATHER ANALYSIS AND IMPLEMENTATIONS
  {
    light > 500 ? lighton() : lightoff();
    precipation > 600 ? windowopen() : windowclose();
    tempC > 28 ? fanon() : fanoff();

  }

  //FINGERPRINT SENSOR FOR DOOR
  if (finger.getImage() == FINGERPRINT_OK) {
    if (finger.image2Tz() == FINGERPRINT_OK) {
      if (finger.fingerSearch() == FINGERPRINT_OK) {
        OpenDoor();
        delay(2000);
        CloseDoor();
      }
    }
  }
  //Look for TFID FOR DOOR
  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {

    // Check if the detected card/tag matches any of the known IDs
    if (checkID1(mfrc522.uid.uidByte, mfrc522.uid.size) || checkID2(mfrc522.uid.uidByte,
mfrc522.uid.size))
    {
      OpenDoor();
      delay(2000);// open door for 2 seconds and then close
      CloseDoor();
    }
    // Halt PICC to stop further communication
    mfrc522.PICC_HaltA();
```

```
    // Stop encryption on PCD
    mfrc522.PCD_StopCrypto1();
  }
 if (air > 200)// HAZARD DETECTION FOR POSSIBLE GAS LEAKS OR FIRE
 {
  firehazards();
 }

if (irrecv.decode(&results)) {
   Serial.println(results.value, HEX);
   if (results.value == 0xFFA25D) //ACTIVATE REMOTE CONTROL with power btn
   {
    control=2;//remote control
    mp3_play(10);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL1:RMT CTRL");
    lcd.setCursor(0,1);
    lcd.print("ACTIVATED");
    delay(1500);
    lcd.clear();
   }
   if (results.value == 0xFFE21D) //DEACTIVATE REMOTE CONTROL with MENU btn
   {
    control=1;
    mp3_play(11);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL1:RMT CTRL");
    lcd.setCursor(0,1);
    lcd.print("DEACTIVATED");
    delay(1500);
    lcd.clear();
   }
   irrecv.resume(); // Receive the next value

   if (control==2)
   {
    switch(results.value){
      case  0xFF6897:  lighton();       mp3_play (2);             break;//btn 0
      case  0xFF30CF:  lightoff();      mp3_play(3);              break;//btn 1
      case  0xFF18E7:  windowopen();    mp3_play(4);              break;//btn 2
      case  0xFF7A85:  windowclose();   mp3_play(5);              break;//btn 3
      case  0xFF10EF:  OpenDoor();      mp3_play(6);              break;//btn 4
      case  0xFF38C7:  CloseDoor();     mp3_play(7);              break;//btn 5
      case  0xFF5AA5:  fanon();         mp3_play(8);             break;//btn 6
      case  0xFF42BD:  fanoff();         mp3_play(9);             break;//btn 7
      case  0xFF4AB5:  Serial.println("B");                         break;
      case  0xFF52AD:  Serial.println("B");                         break;
      case  0xFF22DD:  Serial.println("B");                         break;
      case  0xFFC23D:  Serial.println("C");                         break;
    }
   }
```

```
  }
  CMDID = DF2301Q.getCMDID();
  if(0 != CMDID) {
    Serial.print("CMDID = ");
    Serial.println(CMDID);
  }
  if (CMDID==5)//ACTIVATE VOICE CONTROL
  {
    control=3;
    mp3_play(12);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL2:VCE CTRL");
    lcd.setCursor(0,1);
    lcd.print("ACTIVATED");
    delay(1500);
    lcd.clear();

  }
  if (CMDID==6)//DEACTIVATE VOICE CONTROL
  {
    control=1;
    mp3_play(13);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MANUAL2:VCE CTRL");
    lcd.setCursor(0,1);
    lcd.print("DEACTIVATED");
    delay(1500);
    lcd.clear();

  }
  if(control==3)
  {
  switch (CMDID)
  {
    case 103:     lighton();                    mp3_play (2);          break;
    case 104:     lightoff();                   mp3_play(3);           break;
    case 137:     windowopen();                                        break;
    case 138:     windowclose();                                       break;
    case 141:     OpenDoor();                                          break;
    case 142:     CloseDoor();                                         break;
    case 75:      fanon();                                             break;
    case 76:      fanoff();                                            break;

  }

}
delay(1000);
}


// Functions Definations
void rainbowBlink(uint16_t onTime, uint16_t offTime) {
```

```
  uint16_t i, j;

  // rainbow on
  for(j=0; j<256*5; j++) { // 5 cycles of all colors on wheel
    for(i=0; i< LED.numPixels(); i++) {
      LED.setPixelColor(i, Wheel(((i * 256 / LED.numPixels()) + j) & 255));
    }
    LED.show();
    delay(2);
  }

  // wait for onTime milliseconds
  delay(onTime);

  // rainbow off
  for(i=0; i<LED.numPixels(); i++) {
    LED.setPixelColor(i, 0);
  }
  LED.show();

  // wait for offTime milliseconds
  delay(offTime);
}

uint32_t Wheel(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  if(WheelPos < 85) {
    return LED.Color(255 - WheelPos * 3, 0, WheelPos * 3);
  }
  if(WheelPos < 170) {
    WheelPos -= 85;
    return LED.Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
  WheelPos -= 170;
  return LED.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}
void lighton()
{
  LED.fill(LED.Color(255,255,255));
  LED.show();
}
void lightoff()
{
  LED.fill(LED.Color(0,0,0));
  LED.show();
}
void windowopen()
{
  Window.write(windowopenangle);
}
void windowclose()
{

  Window.write(windowcloseangle);
```

```
}
void fanon()
{

    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
}
void fanoff()
 {

    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
 }
 void OpenDoor()
 {
  Door.write(dooropenangle);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DOOR OPEN!");
  delay(1000);
  lcd.clear();
 }
 void CloseDoor()
 {
  Door.write(doorcloseangle);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("DOOR CLOSED!");
  delay(1000);
  lcd.clear();

 }
 bool checkID1(byte uid[], byte uidSize) {
  // Check if the detected ID matches the known ID
  if (uidSize != sizeof(knownID1)) {
    return false; // Different size, not the same card
  }
  for (byte i = 0; i < uidSize; i++) {
    if (uid[i] != knownID1[i]) {
      return false; // Different ID
    }
  }
  return true; // Matched known ID
}
bool checkID2(byte uid[], byte uidSize) {
  // Check if the detected ID matches the known ID
  if (uidSize != sizeof(knownID2)) {
    return false; // Different size, not the same card
  }
  for (byte i = 0; i < uidSize; i++) {
    if (uid[i] != knownID2[i]) {
      return false; // Different ID
    }
  }
```

```
  return true; // Matched known ID
}
void firehazards()
{
  lcd.clear();
  digitalWrite(buzzer,LOW);
  lcd.setCursor(0, 0);
  lcd.print("FIRE!!!EVACUATE!!");

   for (int i = LED.numPixels() - 1; i >= 0; i--) {
    LED.setPixelColor(i, LED.Color(255, 0, 0));
    LED.show();
    delay(10);
    LED.setPixelColor(i, 0);

  }
  digitalWrite(buzzer,HIGH);
  delay(250);
  lcd.clear();
}
```

The overall circuit diagram remains the same as **Fig 39:**