Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

Automation Security and Safety System for Smart Home

Khalifa Dai Elnour Mohamed, University of Kordofan,
Iseldin Mohammed Mohammedali,
University of Kordofan, Faculty of Engineering and Technical Studies

ABSTRACT
This paper presents the design and implementation of an automated home security and safety system aimed at minimizing human intervention while enhancing household protection. The proposed system utilizes a network of sensors to collect environmental data, which is processed by a microcontroller to make real-time decisions. A web-based interface allows users to remotely monitor and control the system.
The system architecture integrates various components, including an electrical circuit that interprets sensor inputs and triggers actions based on predefined commands. Safety measures are embedded through sensors that detect unsafe interactions with electrical appliances, as well as gas leakage sensors to mitigate fire risks. Additionally, temperatures sensors help regulate indoor climate conditions, contributing to overall home safety and comfort.

***Keywords*—Home Automation, Gas Leakage Detection, Arduino IDE, Web-Based Monitoring, Embedded Sensors**

1.1 INTRODUCTION

Home automation refers to a range of technological features that can be installed in a home to enhance convenience, security, and efficiency. It enables remote control of various home functions via a computer or smartphone, automates electronic devices to respond to specific conditions, and integrates multiple systems into a centralized control center for seamless management.

1.2 Literature Review on Home AutomationSystem

Several studies have explored home automation platforms utilizing IoT devices. Research on IoT has been continuously evolving, particularly in conjunction with various applications of the Internet of Things. The future growth of IoT largely depends on its adoption and advancements in security, efficiency, and accessibility.
Shaikh Amreen (2017) described the architecture of home automation systems and explained how IoT can be leveraged to monitor home conditions through sensor-based systems. His study highlights the role of sensors in tracking environmental parameters to enhance convenience and efficiency in smart homes.
Anurag Tiwari et al. (2017) examined the challenges and ongoing research in IoT. While IoT systems have gained widespread popularity, security and privacy concerns vulnerable to cyber threats, making security a crucial aspect of IoT implementation. These concerns have hindered IoT from being fully recognized as a trusted and secure technology.
SatishPalaniappan (2015) discussed how smart homes can integrate various sensors, such as motion, light, and temperature sensors, to automate device operation based on real-time conditions. The study emphasized energy efficiency by ensuring devices operate only when necessary—for instance, turning off lights in unoccupied rooms or adjusting brightness levels automatically. Furthermore, the system can be integrated with home security solutions, enhancing both control and safety for homeowners. The next step in this technological evolution is extending automation beyond residential settings to larger environments such as offices and factories.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

1.3 The Problems

This paper addresses several critical home safety and efficiency issues, including:

1. Gas leakage in kitchens
2. Forgetting to turn off electrical appliances when not in use
3. Temperature imbalances within the home
4. Risks associated with direct contact with electrical equipment and unsafe handling of sockets

1.4 Objectives of the paper

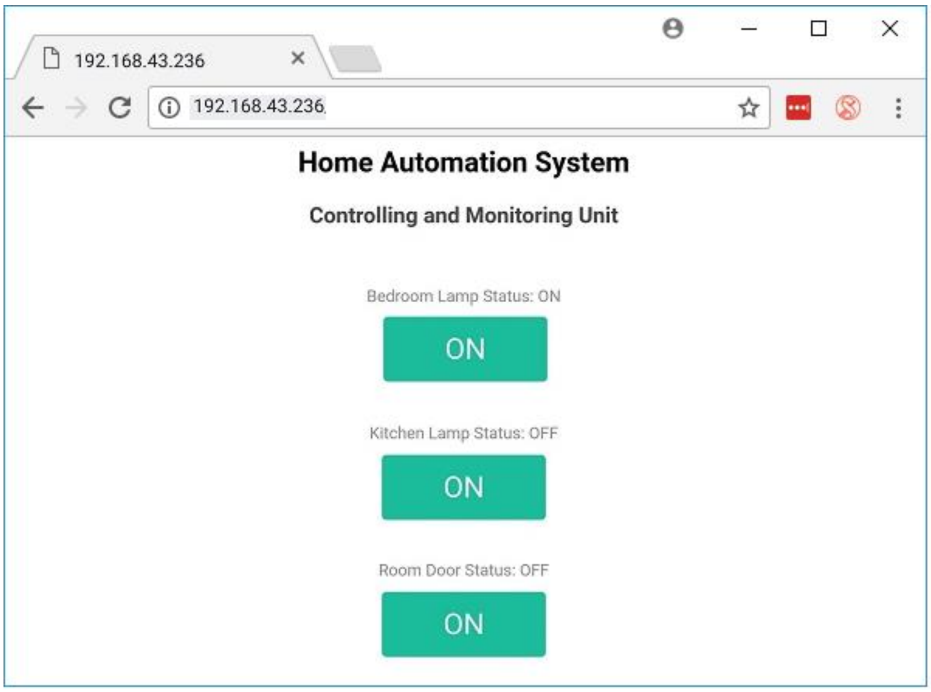
1. Optimize electrical energy usage.
2. Mitigate risks associated with contact with electrical devices.
3. Design a LAN server page for home automation control.
4. Prevent fire hazards within the home.
5. Maintain a balanced room temperature.

1.5 Methodology

Design an electrical circuit that collects and processes data based on predefined commands to execute specific tasks.

2.1 Accessing the Web Server

A network-based web page is used to control room lighting. This web page communicates with the system via the IP protocol and can be accessed through any internet browser on a phone or computer. The web page is developed using the programming code of the primary system and includes interactive buttons for sending commands. These commands are transmitted to the primary system, which then relays them to the secondary system to control the lighting.



Home Automation System

Controlling and Monitoring Unit

Bedroom Lamp Status: ON

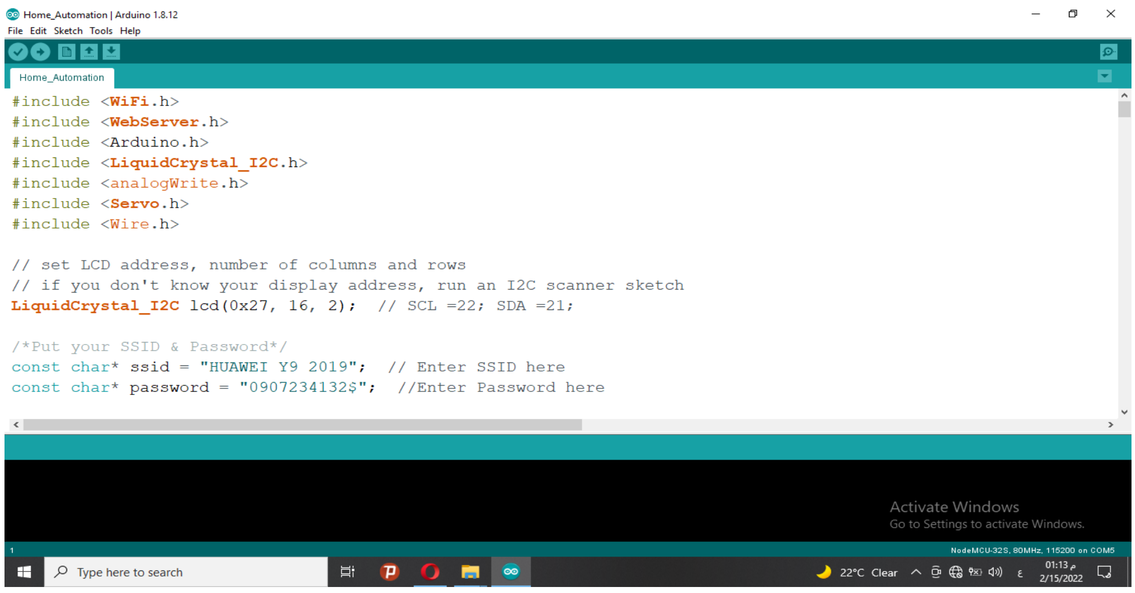
Kitchen Lamp Status: OFF

Room Door Status: OFF

Fig(1) : The web server page

2.2 Code writing interface (Arduino IDE)

In the interface unit in figure (2) below, the programming code for the control panel is written and uploaded, from which commands are also sent and the work of all sensors is monitored.
The codes contain the main program and sub-programs from the Arduino programming language. It also includes libraries specialized in dealing with sensors and other electronic circuit parts such as the circuits for connecting the crystal display (I^2C adapter).


Home_Automation | Arduino 1.8.12
File Edit Sketch Tools Help
Home_Automation
Activate Windows
Go to Settings to activate Windows.
NodeMCU-32S, 80MHz, 115200 on COM5
Type here to search
22°C Clear
01:13
2/15/2022

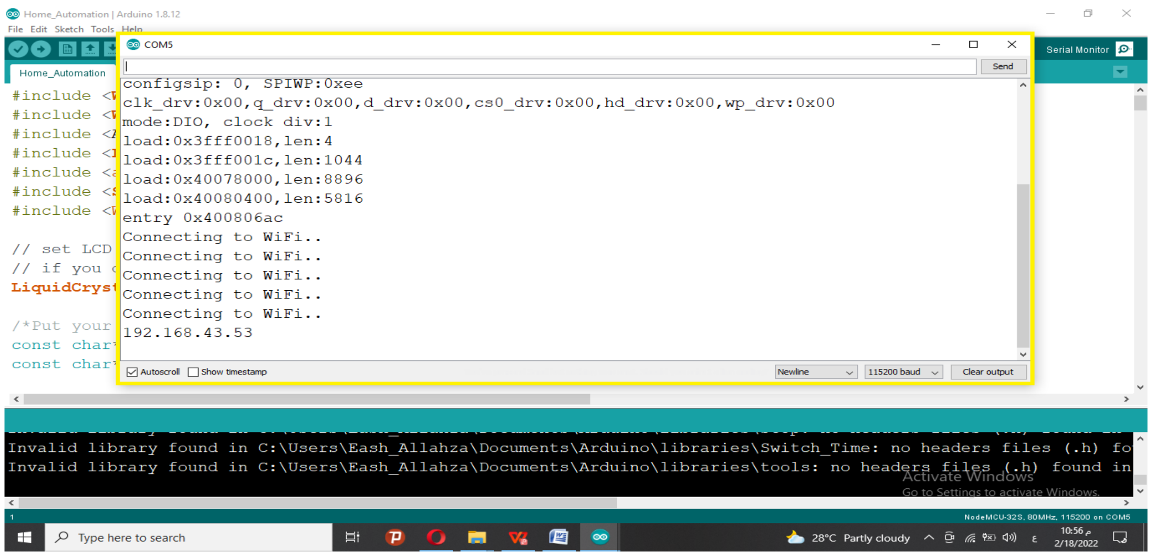
```
#include <WiFi.h>
#include <WebServer.h>
#include <Arduino.h>
#include <LiquidCrystal_I2C.h>
#include <analogWrite.h>
#include <Servo.h>
#include <Wire.h>

// set LCD address, number of columns and rows
// if you don't know your display address, run an I2C scanner sketch
LiquidCrystal_I2C lcd(0x27, 16, 2);  // SCL =22; SDA =21;

/*Put your SSID & Password*/
const char* ssid = "HUAWEI Y9 2019";  // Enter SSID here
const char* password = "0907234132$";  //Enter Password here
```

Fig(2): Graphic interface unit

The interface also contains a window known as the Serial monitor to display the values and graphical charts of the variable values.It is called a serial monitor because it monitors all the input and output values of the microcontroller ,as indicated in the yellow rectangle in figure (3) below.



```
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac
Connecting to WiFi..
Connecting to WiFi..
Connecting to WiFi..
Connecting to WiFi..
Connecting to WiFi..
192.168.43.53
```

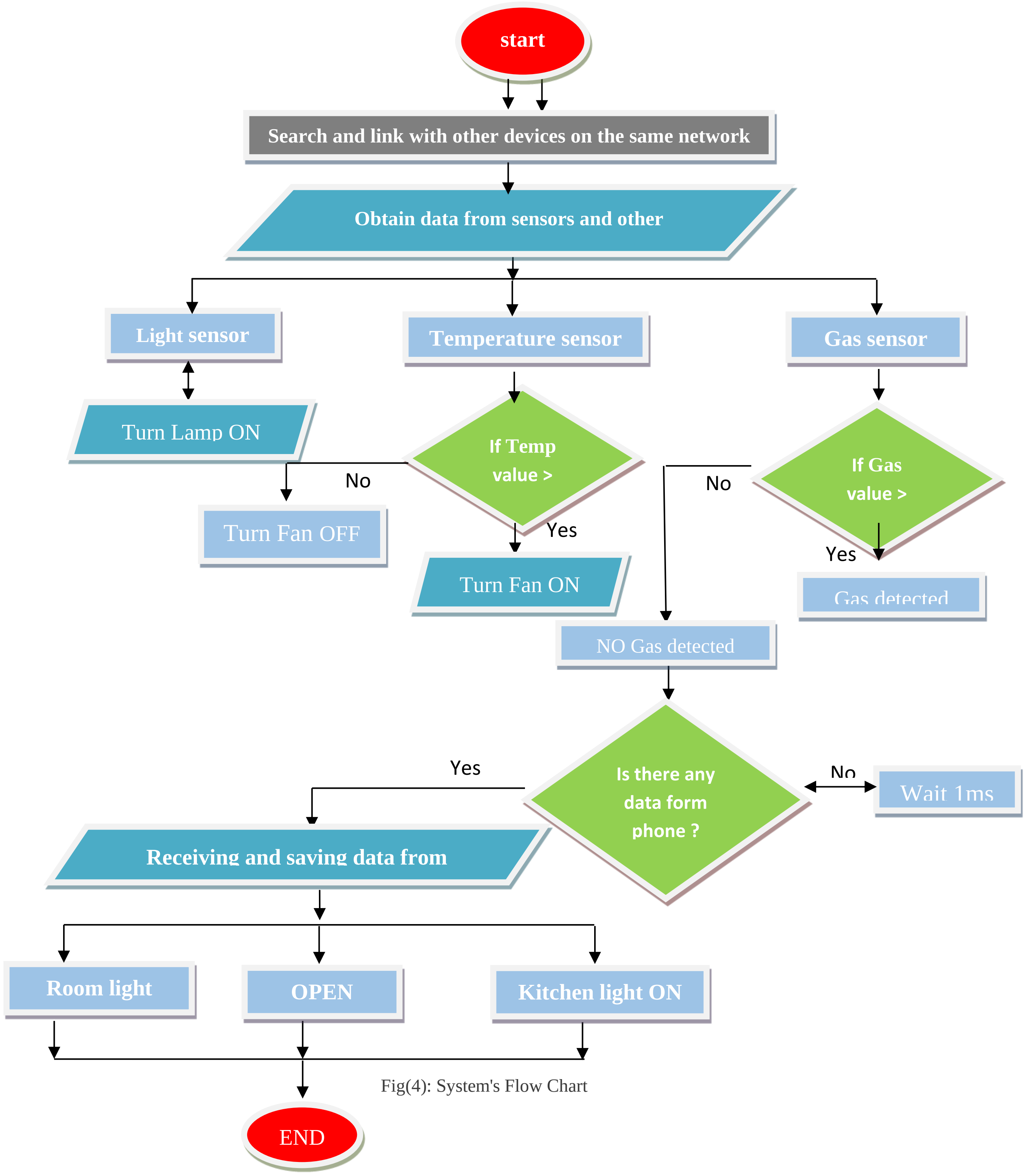
```
Invalid library found in C:\Users\Eash_Allahza\Documents\Arduino\libraries\Switch_Time: no headers files (.h) fo
Invalid library found in C:\Users\Eash_Allahza\Documents\Arduino\libraries\tools: no headers files (.h) found in
```

Fig(3): Serialmonitor

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

2.3 Algorithm

System's Flow Chart in figure (4) shows the transfer algorithm in the system.

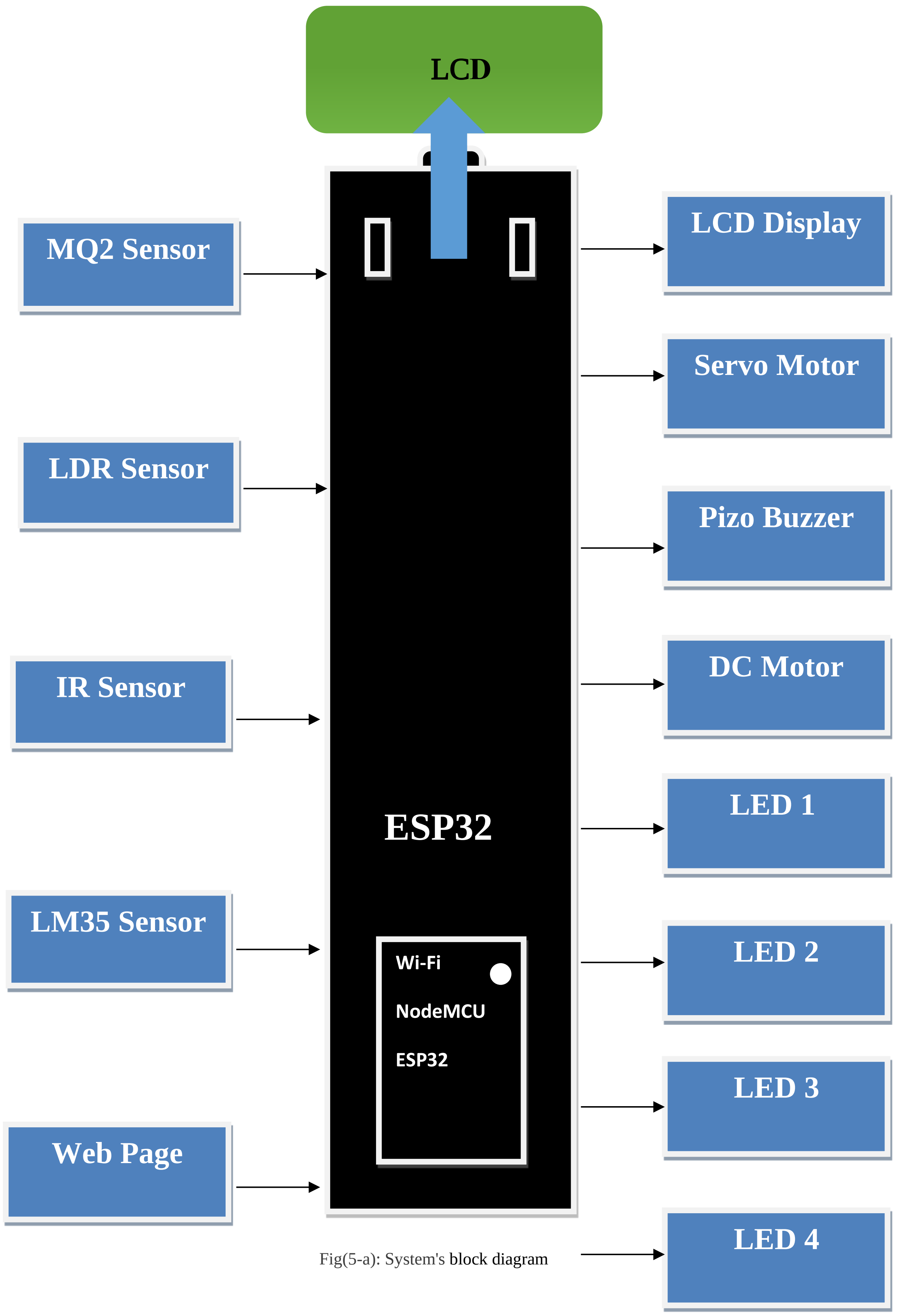


Fig(4): System's Flow Chart

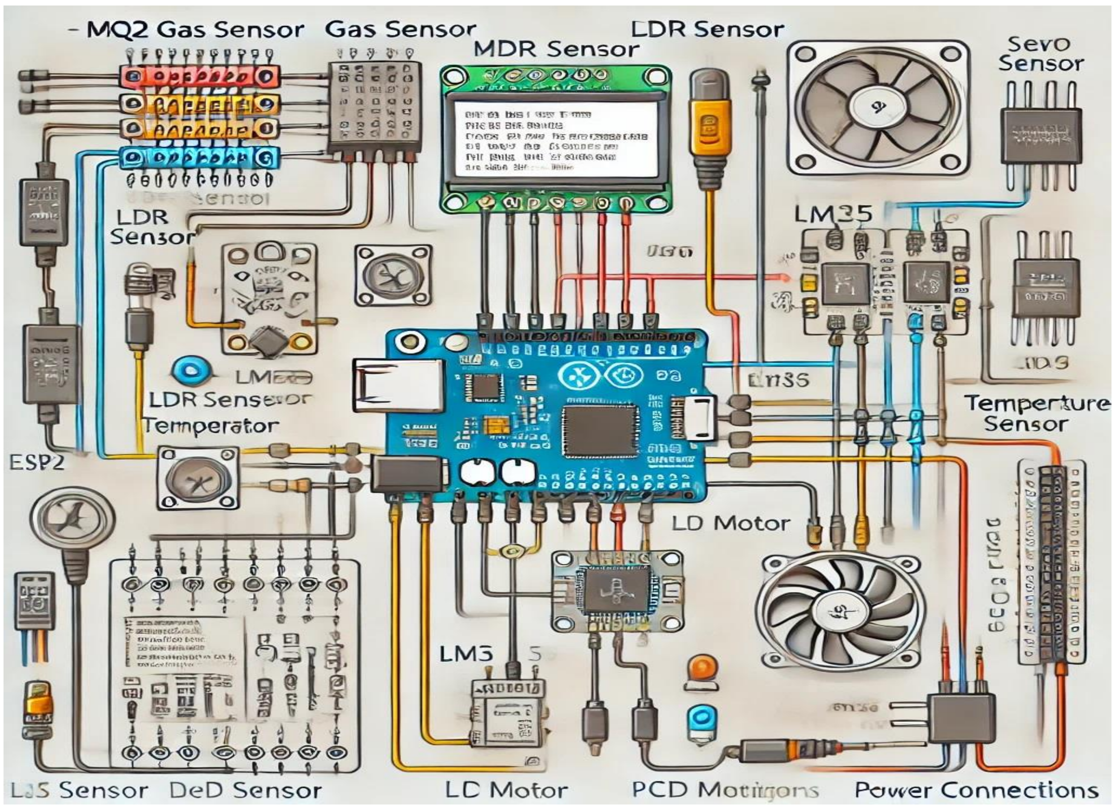
3.1 System's block diagram

The system collects data from the surrounding environment through sensors that are inserted into the controller to make the appropriate decision. There is also a web server page to enter data into the system for control and monitoring. Figure (5-a) shows the system's block diagram. Figure (5-b) shows what the final wiring scheme should look like after all connections are made.

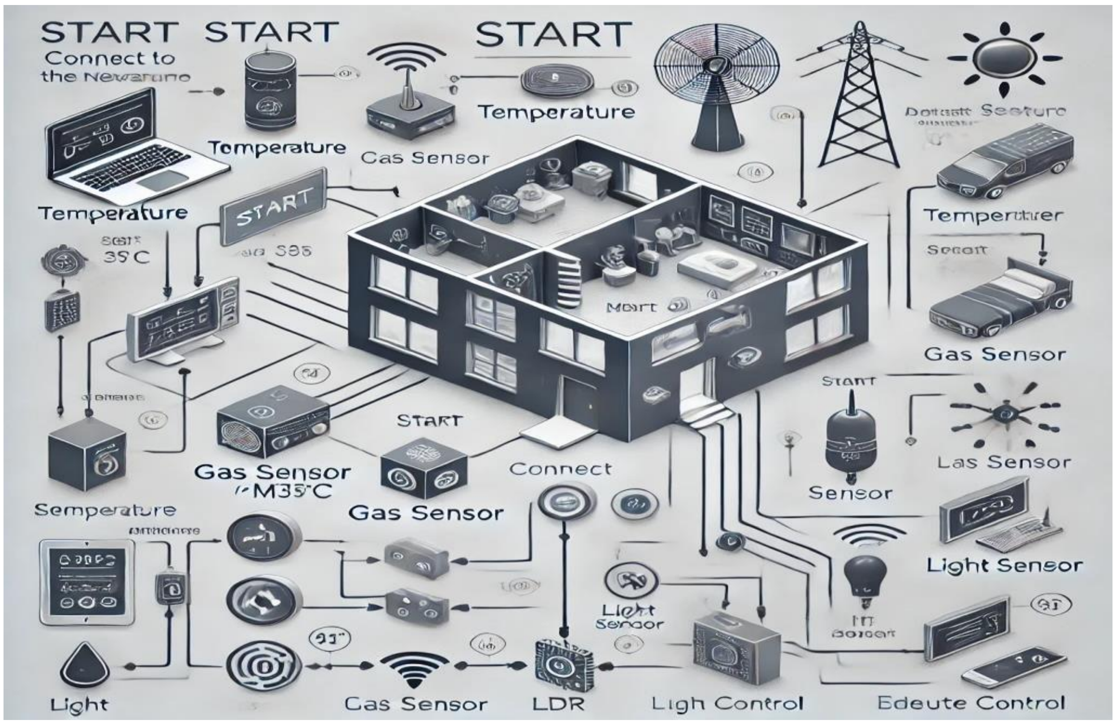
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025



Fig(5-a): System's block diagram

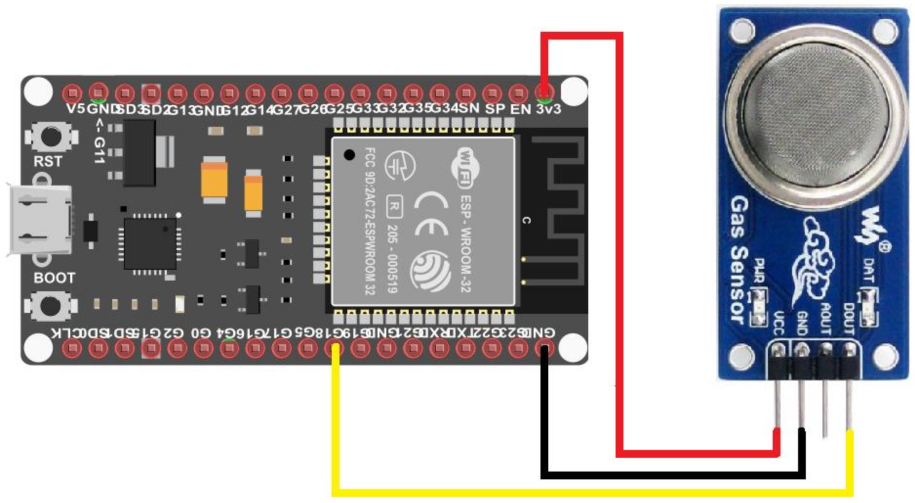


Fig(5-b): System's final wiring scheme



3.2 Connecting the Gas sensor to the ESP

Start by placing the sensor on to your breadboard. Connect VCC pin to the 3.3V pin on the ESP and connect GND pin to the Ground pin on the ESP.
Connect A0 output pin on the module to Analog pin GIOP34 on the ESP. Figure (6) shows the process of connecting the gas sensor to the ESP.


V5 GND SD3 SD2 G13 GND G12 G14 G27 G26 G25 G33 G32 G35 G34 SN SP EN 3v3
RST
BOOT
<- G11
ESP - WROOM-32
FCC 9D:2AC72-ESPWROOM 32
205 - 000519
CLK SD0 SD1 G15 G2 G0 G4 G16 G17 G5 G18 G19 GND G21 RXD TXD G22 G23 GND
Gas Sensor
PWR
DAT
DOUT
AOUT
GND
VCC

Fig(6): Connecting the Gas Sensor to the ESP

3.3 Connecting the LDR sensor to the ESP

Start by placing the sensor on to your breadboard. Connect VCC pin to the 3.3V pin on the ESPand connect GND pin to the Ground pin on the ESP .Connect S pin on the module to Digital pin GIOP39 on the ESP . Figure (7) shows the process of connecting the LDR Sensor to the ESP.

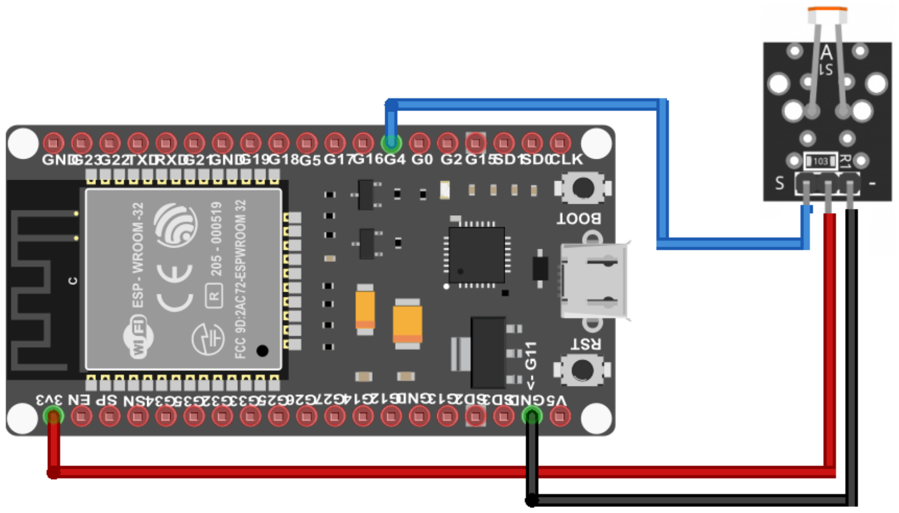

GND G23 G22 TXD RXD G21 GND G19 G18 G5 G17 G16 G4 G0 G2 G15 SD1 SD0 CLK
BOOT
RST
<- G11
ESP - WROOM-32
205 - 000519
FCC 9D:2AC72-ESPWROOM 32
3V3 EN SP SN G34 G35 G32 G33 G25 G26 G27 G14 G12 GND G13 SD2 SD3 GND V5
A
S1
R1
103
S
-

Fig (7): Connecting the LDR Sensor to the ESP

3.4 Connecting the Temperature Sensor to the ESP

There are different types of temperature sensors, including the LM35 sensor.

Start by placing the sensor on to your breadboard. Connect VCC pin to the 3.3V pin on the ESP and connect GND pin to the Ground pin on the ESP . Connect Signal pin on the Temperature Sensor to Analog pin GIOP36 on the ESP . Figure (8) shows the LM35 sensor connected to the ESP, Connecting the LM35 Sensor to the ESP

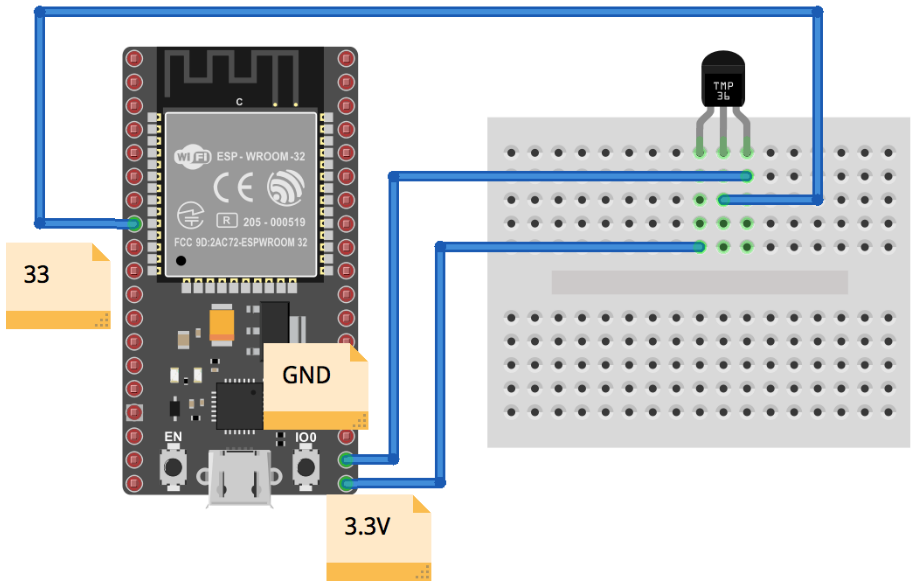


Fig (8): Connecting the LM35 Sensor to the ESP

3.5 Connecting the IR Sensor to the ESP

Start by placing the sensor on to your breadboard. Connect VCC pin to the 3.3V pin on the ESP and connect GND pin to the Ground pin on the ESP. Connect OUT pin on the module to Digital pin GIOP12 on the ESP. Figure (9) shows the IR sensor connection and Connecting the IR Sensor to the ESP.

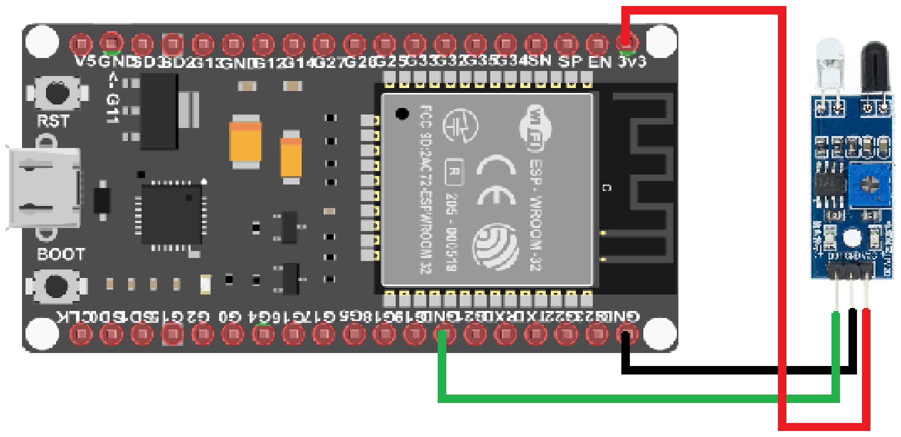


Fig (9): Connecting the IR Sensor to the ESP

3.6 Connecting the LEDs to the ESP

The LEDs is connected to the digital ports of the ESP (GIOP16 ,5,18,19,23) through a resistor to limit the intensity of the current and its value ranges from 220 ohms to 1 kilo ohm as appears in figure (10) below.

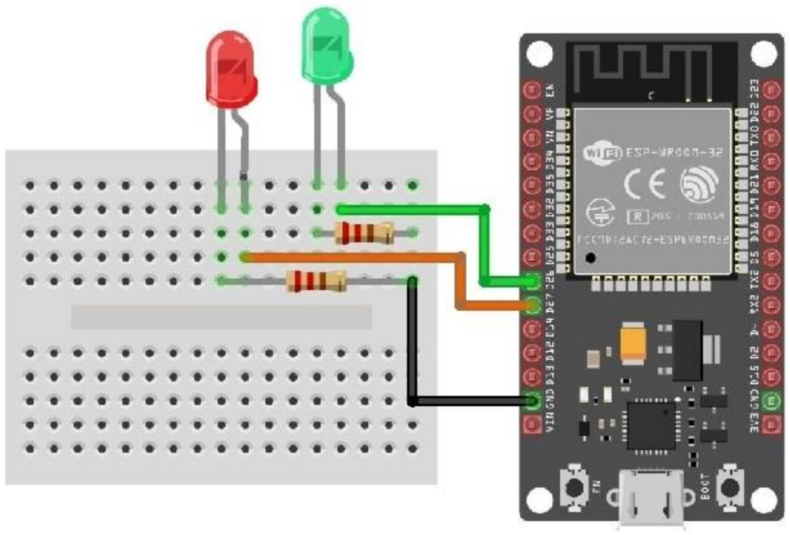


Fig (10): LEDs give only one color

3.7 Connecting DC Motor to the ESP

The base of the transistor is connected to the digital pin GIOP17 of the ESP, while the emitter and collector terminals are connected to the power and ground terminals as appears in figure (11) below.

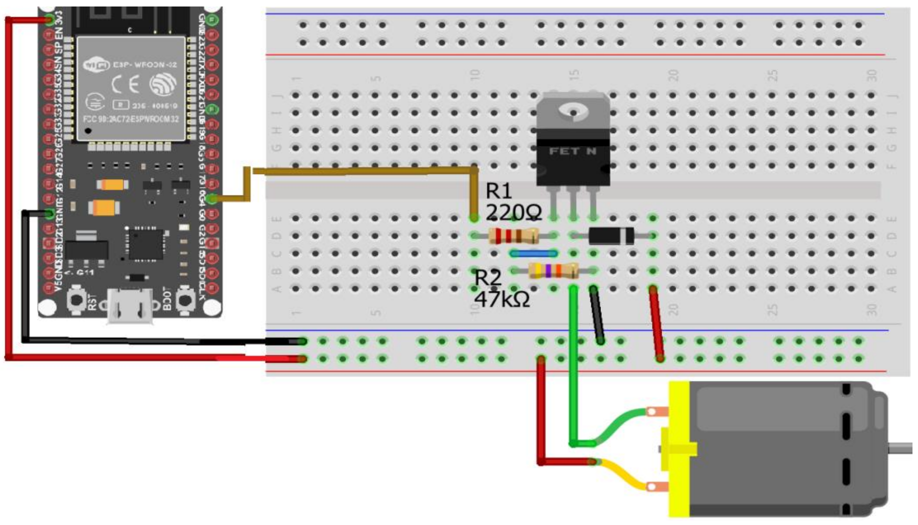


Fig (11): Connecting the DC Motor to the ESP

3.8 Connecting Servo Motor to the ESP

I use the Servo motor here to indicate the movement of doors and windows .The servo motor consists of three terminals, two terminals for power and one terminal for the signal that connects to the analog pin GIOP4 in the ESP. Figure (12) shows how to connect the servo motor to the ESP.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

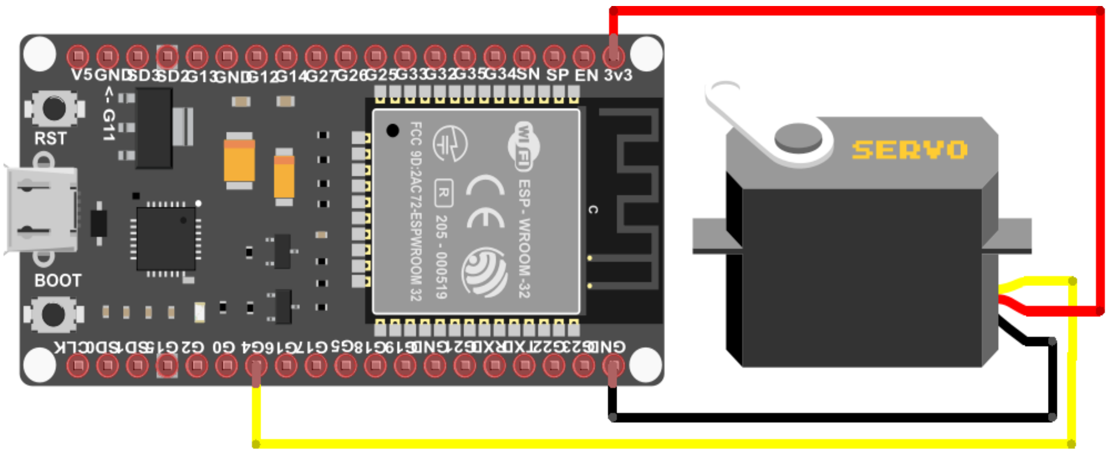
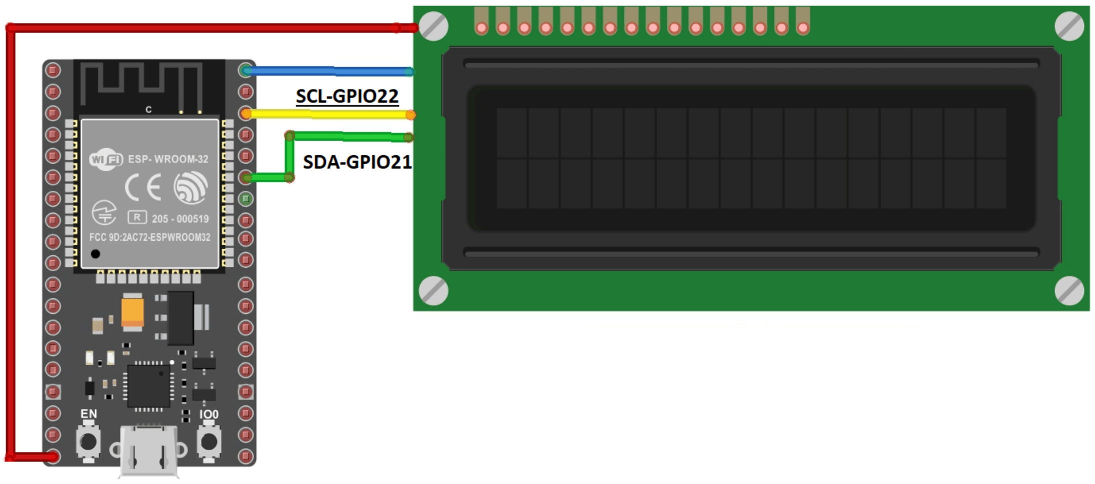


Fig (12): Connecting the Servo Motor to the ESP

3.9 Connecting the LCD Display to the ESP

The crystal display is connected to the ESP through the I^2C adapter , The I^2C adapter connected to the LCD display with 16 pins and outputs only 4 pins, 2 for data and another 2 for power and ground as appears in figure (13) below.



Fig(13): Connecting the LCD Display to the ESP

4.1 Results and analyses

However, when we started uploading the code to the ESP32 board and waited for a while until the upload process was completed, the upload process did not complete and "A fatal error occurred: Connection to ESP32 failed: Time. The error message "Specify" may be displayed. Off ... Connect ... ". This means that the ESP32 is not in flash / upload mode. Having the right board name and COM por selected, follow these steps:

- Hold-down the “**BOOT**” button which appears in figure (14) in your ESP32 board.

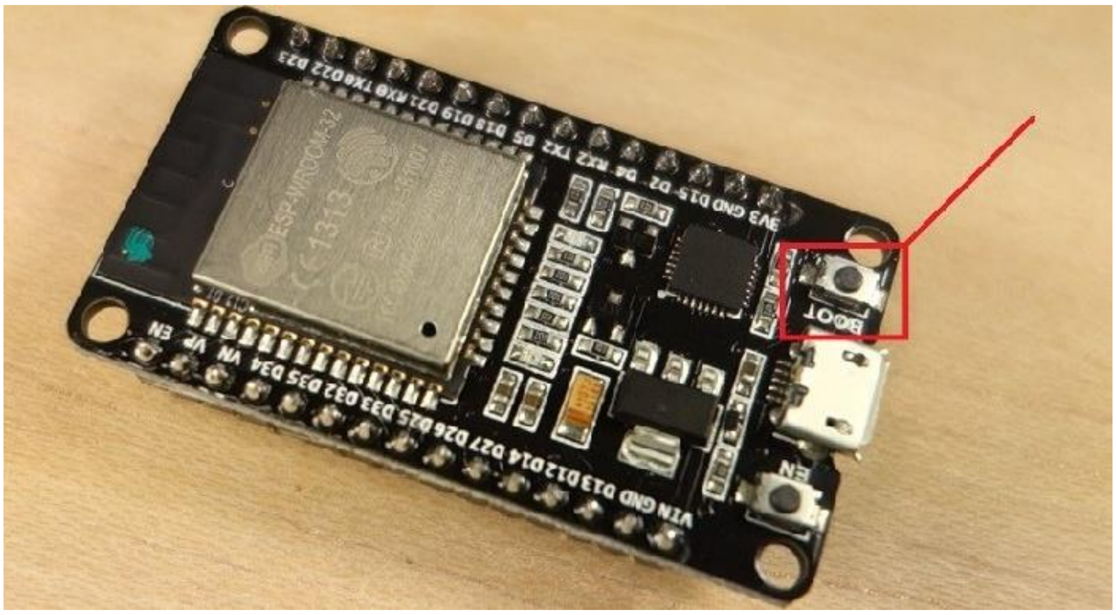
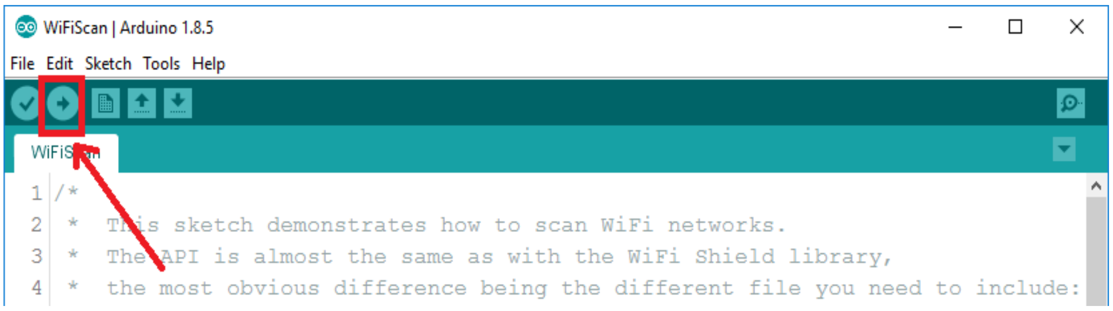


Fig (14): BOOT button

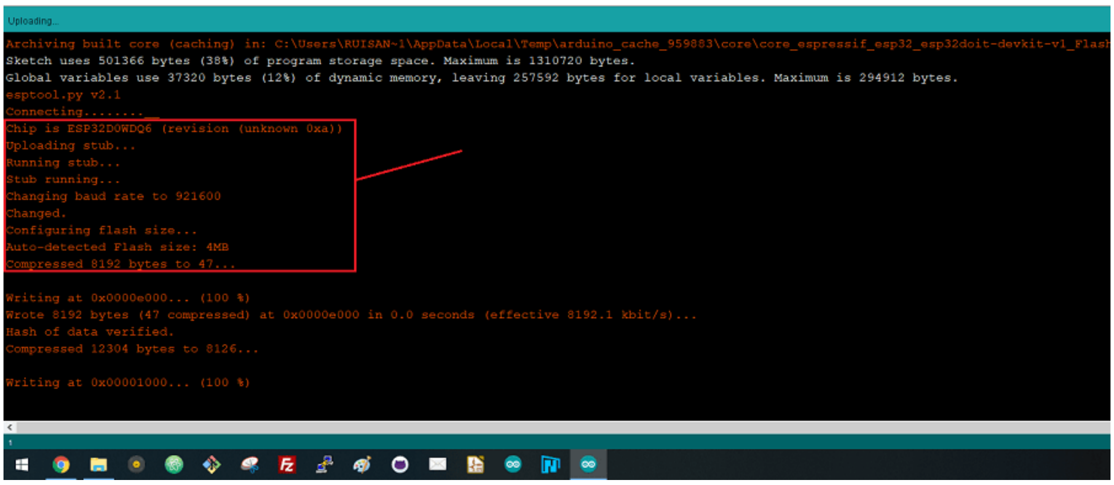
- Press the “Upload” button in the Arduino IDE which appears in figure (15) to upload your sketch:



```
/*
 *  This sketch demonstrates how to scan WiFi networks.
 *  The API is almost the same as with the WiFi Shield library,
 *  the most obvious difference being the different file you need to include:
```

Fig (15): Upload button

- After you see the “Connecting….” message in your Arduino IDE which appears in figure (16), release the finger from the “BOOT” button:



```
Uploading...
Archiving built core (caching) in: C:\Users\RUISAN~1\AppData\Local\Temp\arduino_cache_959883\core\core_espressif_esp32_esp32doit-devkit-v1_Flash
Sketch uses 501366 bytes (38%) of program storage space. Maximum is 1310720 bytes.
Global variables use 37320 bytes (12%) of dynamic memory, leaving 257592 bytes for local variables. Maximum is 294912 bytes.
esptool.py v2.1
Connecting........_
Chip is ESP32D0WDQ6 (revision (unknown 0xa))
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...

Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 8192.1 kbit/s)...
Hash of data verified.
Compressed 12304 bytes to 8126...

Writing at 0x00001000... (100 %)
```

Fig (16): Connecting….” message

- After that, you should see the "Done uploading" message

That's it. We need to run a new sketch on the ESP32. Press the "ENABLE" button to restart the ESP32 and run the newly uploaded sketch.
You'll also have to repeat that button sequence every time you want to upload a new sketch.

4.2 Gas sensor results and analyzes

Now the gas sensor starts to check for gas leaks, and we notice that the gas sensor shows a message in the crystal display (NO Gas leak detected!) as appears in figure (17) below .

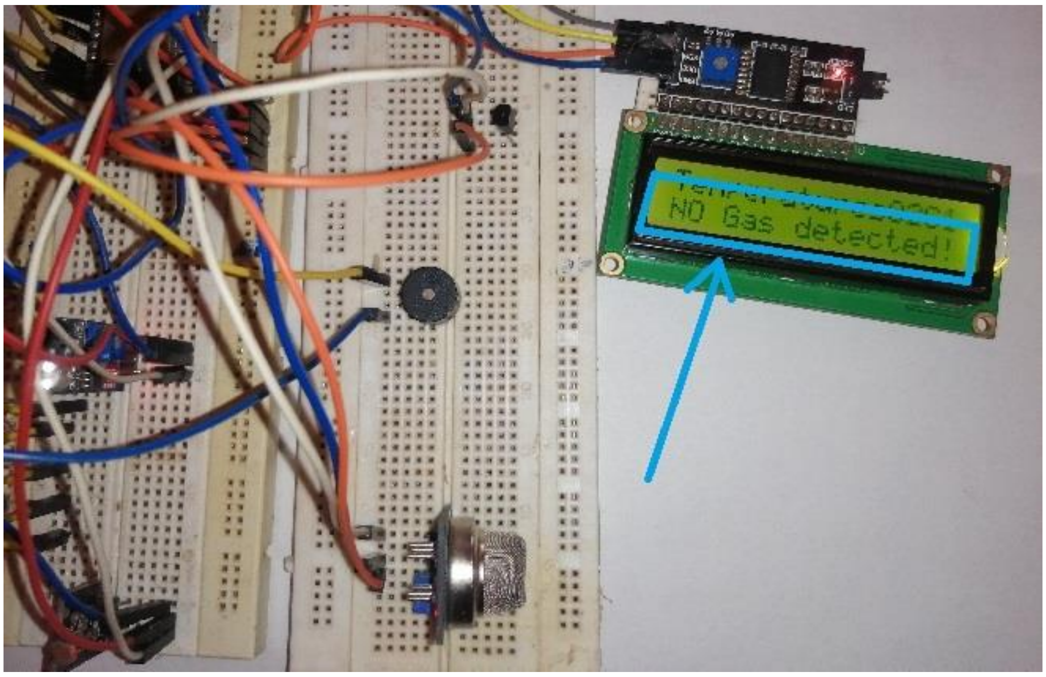


Fig (17): NO Gas leak detected

In the event of a gas leak, the message changes to (Gas leak detected!), as we note from the image below in figure (18) when the sensor was exposed to some gases that result from the burning of papers. The buzzer of the sound referred to in the picture also emits a warning sound in conjunction with the appearance of the gas until its disappearance.

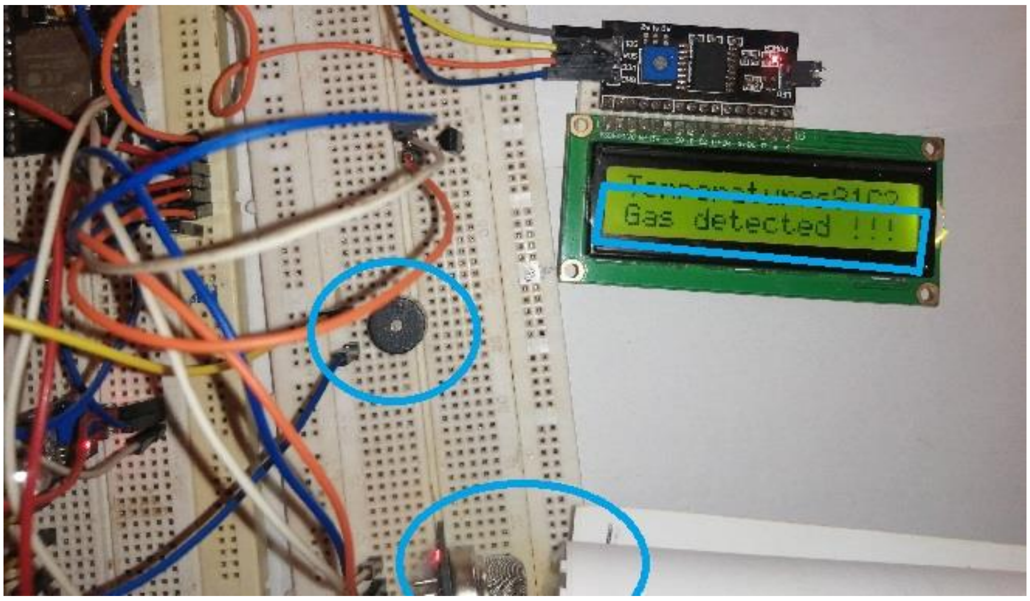


Fig (18): Gas leak detected

I also note during the experiment from the graph of the gas sensor, which represents the analog output values of gas sensor that the analog values in the normal situation are stable in the range of 750 to 850, but when exposed to gases, the symmetric values rise to the limits of 1800 values. Figure (19) shows the graph of analog output values of gas sensor.

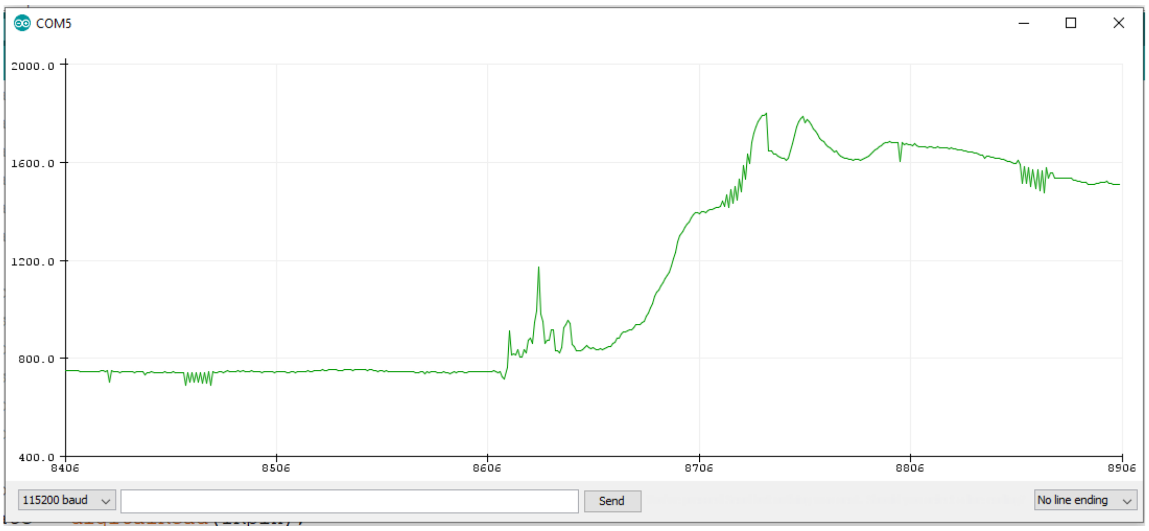


Fig (19): The analog output values of gas sensor

4.3 Light sensors results and analyzes

After testing the IR sensor as appears in figure (20), I noticed that when no object passed, the LED indicated below would be off, and when someone passed a finger in front of the sensor, the blue LED turned on.
I also noticed that the sensor's work during daylight hours is not stable, as a result of the interference of infrared rays emanating from sunlight, so it is the best time to work at night.

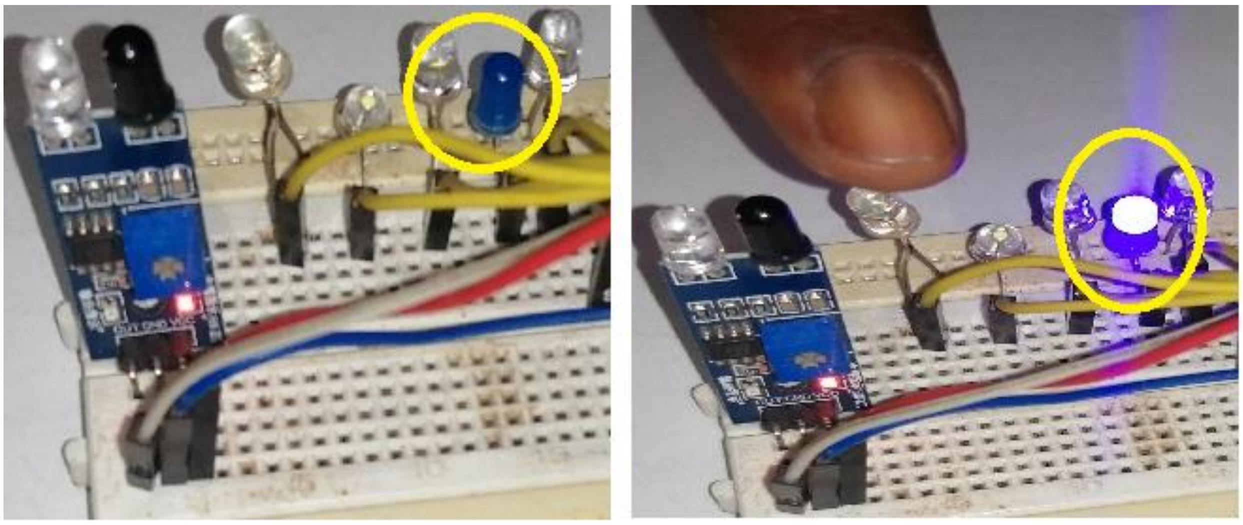


Fig (20): Testing of IR sensor

With this mechanism of work, I solved the problem of direct dealing with electrical switches, which may result in an electric shock, especially in the shower rooms. Also, this mechanism is a solution to the problem of not remembering to turn off the lights in the shower rooms when there is no need for them.
Speaking of lighting, the LDR sensor was used to control the street lamps around the houses and on their roofs. This sensor helps make the lamps turn off during the day and only work in the dark hours. Figure (21) shows the practical test of the sensor in the case of daytime and dark.

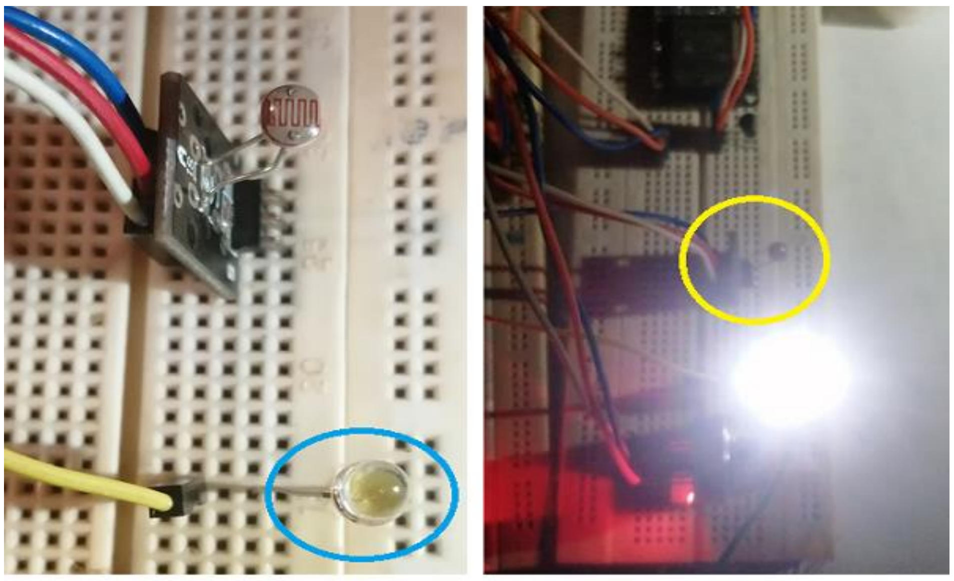


Fig (21): A practical test of the sensor in the case of daytime and dark

The graph in figure (22) below shows the relationship of the analog output produced by the sensor when it is illuminated, as the output values in darkness range from 600 to 1023, and it decreases significantly to less than 200 when exposed to light, as shown by the red rectangle in the figure below.

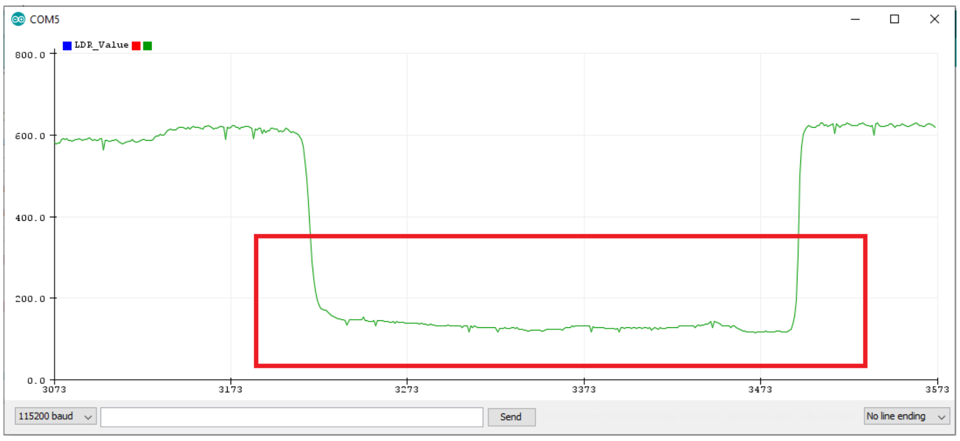
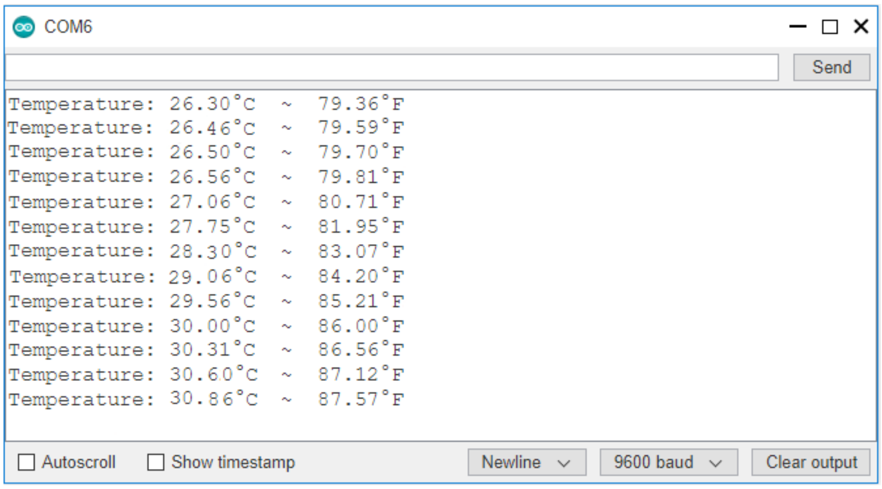


Fig (22): The analog output produced by the sensor

4.4 Temperaturesensor results and analyzes

When testing the temperature sensor, it was observed that it gives temperature values ranging from 26° C to 33° C with an accuracy of +/- 0.4°C at room temperature.Figure (23) shows the results of the temperature sensor readings.



```
Temperature: 26.30°C  ~  79.36°F
Temperature: 26.46°C  ~  79.59°F
Temperature: 26.50°C  ~  79.70°F
Temperature: 26.56°C  ~  79.81°F
Temperature: 27.06°C  ~  80.71°F
Temperature: 27.75°C  ~  81.95°F
Temperature: 28.30°C  ~  83.07°F
Temperature: 29.06°C  ~  84.20°F
Temperature: 29.56°C  ~  85.21°F
Temperature: 30.00°C  ~  86.00°F
Temperature: 30.31°C  ~  86.56°F
Temperature: 30.60°C  ~  87.12°F
Temperature: 30.86°C  ~  87.57°F
```

Fig (23): The results of the temperature sensor readings

By treating the temperature values with the fan's rotation speed, the problem of temperature balance inside the room has been solved, and this reduces unnecessary energy consumption. Figure (24)shows the temperature sensor test.

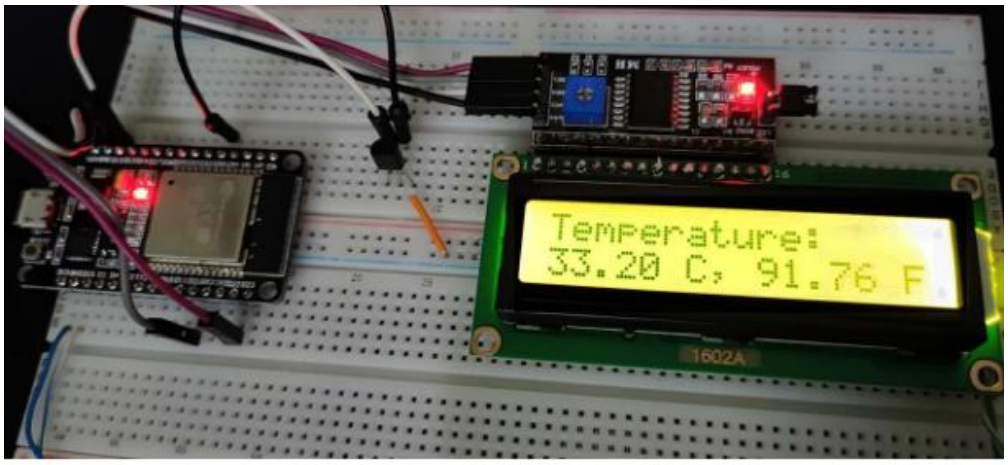
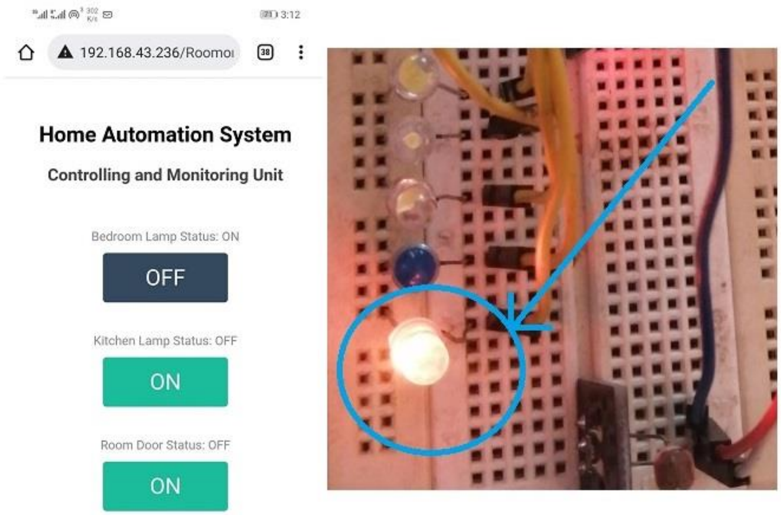


Fig (24): Temperature sensor test

4.5 Web server page to control lights and door

In this part of the project, I used a web server page to control two lamps and a small servo motor. Commands were sent from the control page through the buttons, where I find that when I touch the bedroom button in the browser, the lamp turns on and when I touch again it stops working. Figure (25) shows the test when I touch the bedroom button in the browser.



192.168.43.236/Roomoı

Home Automation System

Controlling and Monitoring Unit

Bedroom Lamp Status: ON

OFF

Kitchen Lamp Status: OFF

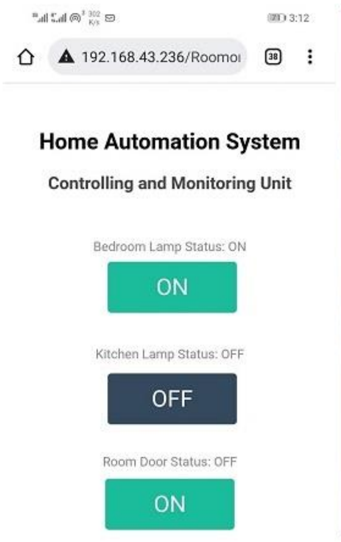
ON

Room Door Status: OFF

ON

Fig (25): The testwhen I touch the bedroom button in the browser

I also find that when you touch the kitchen button in the browser, the lamp turns on, and when you touch it again, it stops working. Figure (26) shows the test when I touch the kitchen button in the browser.



192.168.43.236/Roomoı

Home Automation System

Controlling and Monitoring Unit

Bedroom Lamp Status: ON

ON

Kitchen Lamp Status: OFF

OFF

Room Door Status: OFF

ON

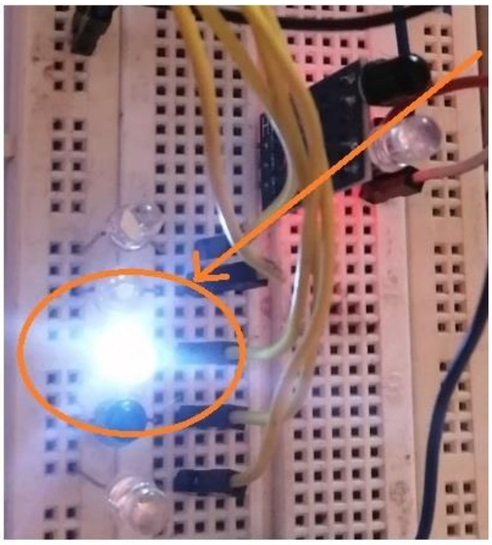


Fig (26): The testwhen we touch the kitchenbutton in the browser

In the normal position of the servo motor, the motor is at 0°, and when you touch the start button in the browser, the servo changes its angle to 90°, and this is useful in controlling the opening and closing of doors and windows, as I use it in this project to control the room door as shown in the image in figure (27) below.

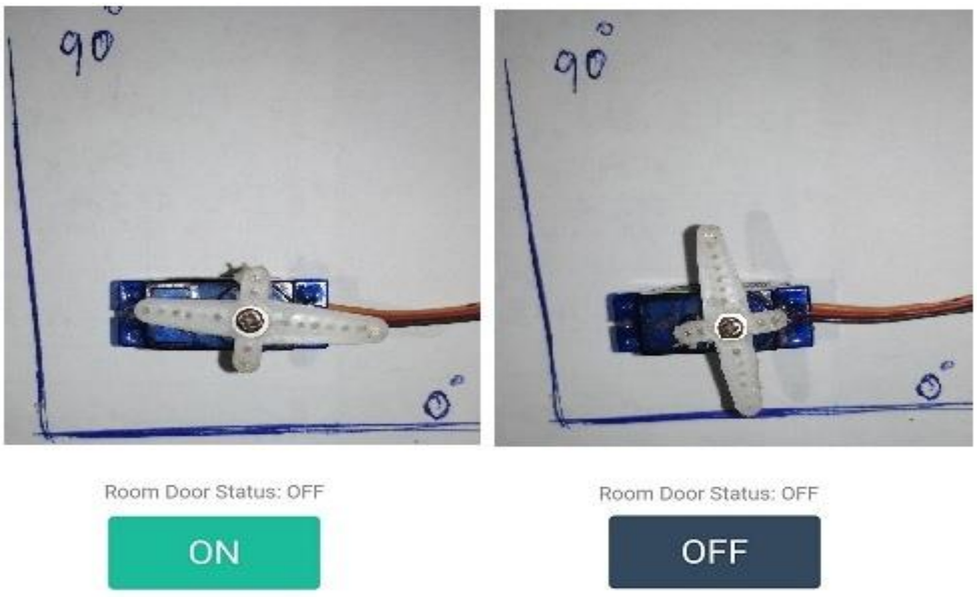


Fig (27): Position of the servo motor

Accordingly, the web server page facilitates dealing with electrical devices remotely with flexibility and safety.
Based on the previous results we obtained, I find that they share the concept of rationalizing electrical energy consumption.

5.1 Conclusion

The project test demonstrated that the system efficiently utilizes sensors to operate only when necessary, contributing to rationalizing electrical energy consumption.

Additionally, the system has sensors that minimize risks associated with direct interaction with electrical sockets and appliances.

Control is made flexible through a web server page, accessible via a phone or computer browser, allowing convenient remote management.

Furthermore, the system includes gas leak detection sensors, enhancing home safety by reducing fire hazards. It is also equipped with temperature-balancing sensors to maintain a comfortable indoor environment.

5.2 Recommendations:

1. Upgrade to a gas sensor that can detect specific types of gases for improved accuracy.
2. Utilize global internet connectivity instead of a local network to expand control capabilities.
3. Integrate surveillance cameras to enhance home security.
4. Connect a 10µF electrolytic capacitor between the EN pin and GND to enable automatic flashing/uploading mode on the ESP32 board.
5. Install a lightning rod to protect home systems from electrical surges and breakdowns.
6. Develop a dedicated network page or mobile application for remote home control from anywhere.
7. Implement a universal remote control system to manage all home appliances, eliminating the need for separate remotes.

DECLARATIONS

Availability of Data and Materials

The experiments and results presented in this paper were conducted at the University of Kordofan, Faculty of Engineering and Technical Studies, Sudan. The data used and analyzed during the current study are available from the author upon reasonable request. This work is original, has not been published elsewhere, and is not under consideration for publication in any other venue.

Competing Interests

The author declares that there are no competing interests.

Funding

No funding was obtained for this study.

Author Contributions

The author read and approved the final manuscript.

Acknowledgements

Not applicable.

REFERENCES

[1] Adriansyah, A., &Dani, A. W. (2014, August). Design of small smart home system based on Arduino. In 2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS) (pp. 121-125). IEEE. https://doi.org/10.1109/EECCIS.2014.7042706

[2] Li, M., & Lin, H. J. (2014). Design and implementation of smart home control systems based on wireless sensor networks and power line communications. IEEE Transactions on Industrial Electronics, 62(7), 4430-4442. https://doi.org/10.1109/TIE.2014.2379586

[3] Cameron, N. (2021). *Electronics projects with the ESP8266 and ESP32*. Elektor.

[4] Eichhorn, D. (2017, February 10). *ESP8266 weather station*. https://shop.squix.org/esp8266-weather-station

[5] Kurniawan, A. (2015). *NodeMCU development workshop* (1st ed.). PE Press.

[6] Seneviratne, P. (2015). *Internet of things with Arduino blueprints*. Packt Publishing.

[7] Li, Y. (2013, June). Design of a key establishment protocol for smart home energy management system. In 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks (pp. 88-93). IEEE. https://doi.org/10.1109/CICSYN.2013.50

[8] Random Nerd Tutorials. (n.d.). *ESP-NOW with ESP32 using Arduino IDE*. https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/

[9] Last Minute Engineers. (n.d.). *Getting started with ESP32 using Arduino IDE*. https://lastminuteengineers.com/esp32-arduino-ide-tutorial/

[10] Elprocus. (n.d.). *Arduino sensor types and their applications*. https://www.elprocus.com/arduino-sensor-types-and-applications/