

# Home Automation System

Kaustubh | 21ucs108

Kushal Singh | 21ucs114

The LNM Institute of Information Technology

Course Name: Device Programming for IoT

Performed Under :

1. Dr. Rajbir Kaur

2. Dr. Sunil Kumar

# Table of Contents

Table of Contents.....	2
Objective.....	3
• Home Automation System.....	4
• Data Flow Diagram .....	7
• Circuit Diagram .....	9
• Conclusion.....	10
• Future Scope.....	11
<u>Circuit Setup .....</u>	12
Blink App Setup and ThingSpeak.....	13
Bibliography .....	14

## Objective

The Home Automation System is a project that aims to automatically control the windows of a home using sensor data. The system uses sensors such as LDR and DHT to gather information about the environment and communicates this information through a node for decision-making. Based on the current sensor readings, the decision-making node uses predefined threshold values to determine whether the window should be open or closed. Additionally, the system includes a manual override feature that allows users to control the windows regardless of the sensor data. The system also includes communicating with a cloud-based application to provide real-time updates on the sensor data and allow users to control the windows remotely. Overall, the Home Automation System provides a convenient and efficient solution for managing window ventilation in a home.

## Home Automation System

### Implementation

In this project, we have developed a home automation system using the NodeMCU ESP8266 as a sensor system with LDR and DHT sensors for monitoring the environment. The system sends data to the NodeMCU ESP32, which acts as an actuation system for altering the state of the window and communicates the data to the user via the Blynk app.

#### Hardware Setup:

The hardware setup consists of the following components:

1. NodeMCU ESP8266: It acts as a sensor system and connects to the LDR and DHT sensors. It also communicates with the ESP32 and the Blynk app.
2. ESP32: It acts as an actuation system and receives data from the NodeMCU ESP8266. Based on the data received and the threshold value of the data, it alters the state of the window. The state of the window can be controlled manually and is depicted by the state of the LED.
3. LDR Sensor: It measures the ambient light level outside the window.
4. DHT22 Sensor: It measures the temperature and humidity levels outside the window.
5. LED: It is used to show manual control of the state of the window.
6. Jumper Wires: It is used for various connections across the devices.
7. Servo Motor:- It is used to replicate the motor actuating the function of Opening and Closing of Window.

8. Buzzer-: It is used for emergency Window opening, in case of some smoke detection or other kind of Harmful gas.
9. Smoke Sensor: It is used to detect the presence of any harmful gas in the atmosphere inside the House.

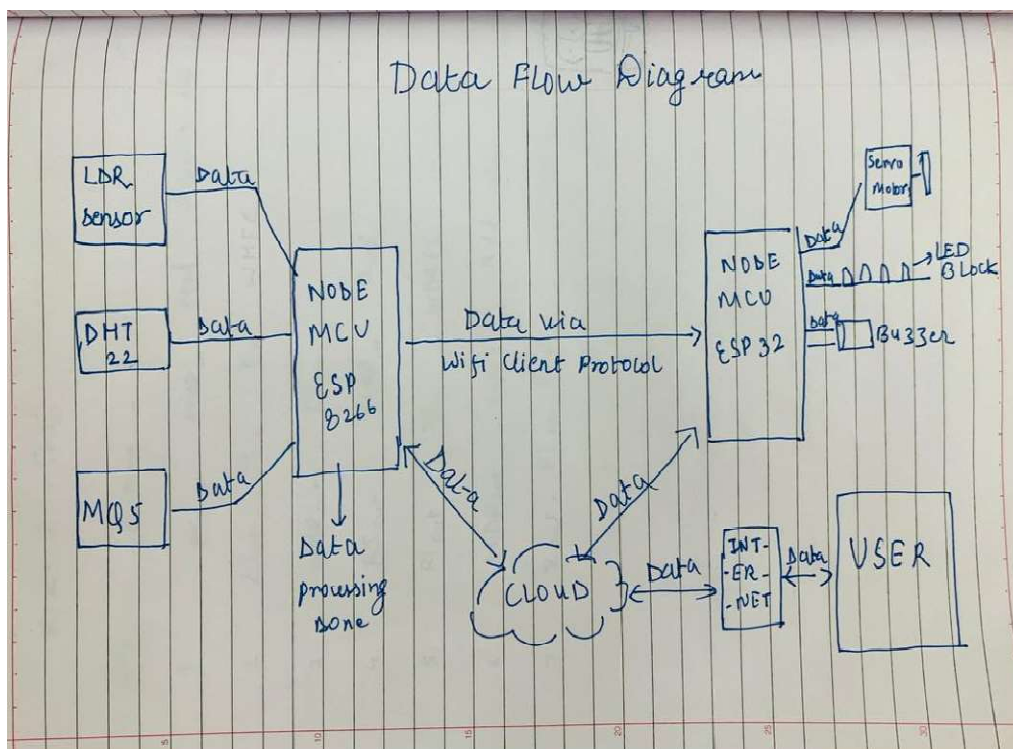
#### Software Implementation:

The software implementation involves the following steps:

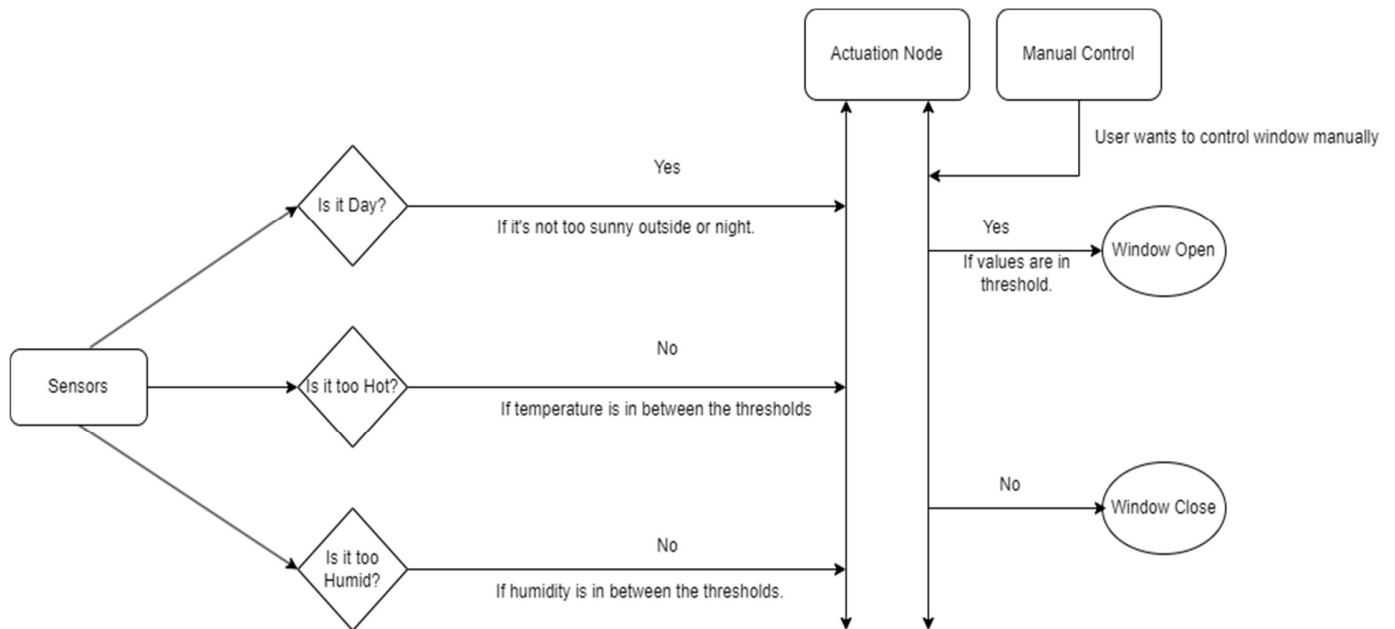
1. Setting up the NodeMCU ESP8266: We have used the Arduino IDE for programming the NodeMCU ESP8266. We have used the Blynk library for communication between the NodeMCU ESP8266 and the Blynk app. We have also used the Adafruit DHT library for interfacing with the DHT sensor and the `analogRead()` function for interfacing with the LDR sensor.
2. Setting up the ESP32: We have used the Arduino IDE for programming the ESP32. Based on the data received and the threshold value of the data, the ESP32 controls the alteration of the window via the state of the LED. Though LEDs can also be used to alter the state of the window manually,
3. Setting up the Blynk app: We have used the Blynk app for monitoring the home automation system remotely. The app displays the temperature, humidity, and light level in the room via widgets. It also displays the state of the window, and the user can also alter the state of the window via a switch OFF/ON button from the app.

4. Node-to-Node: There is node to node communication between the actuation and sensor devices i.e., NodeMCU ESP32 and NodeMCU ESP8266 respectively, using the Wi-Fi Client module of the NodeMCU ESP.
5. ThingSpeak: For Analysis purpose and future prospect, we have used the ThingSpeak as our storing database and with it's inbuilt analysis tools, we are using it for scalability.

## Data Flow Diagram



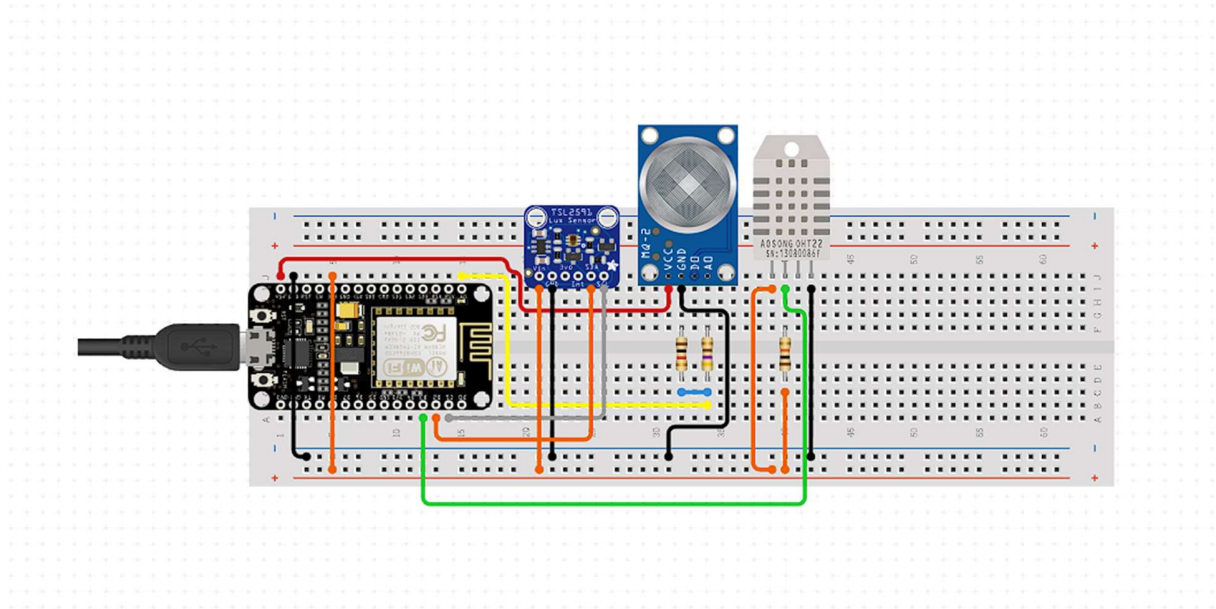
## Decision Diagram



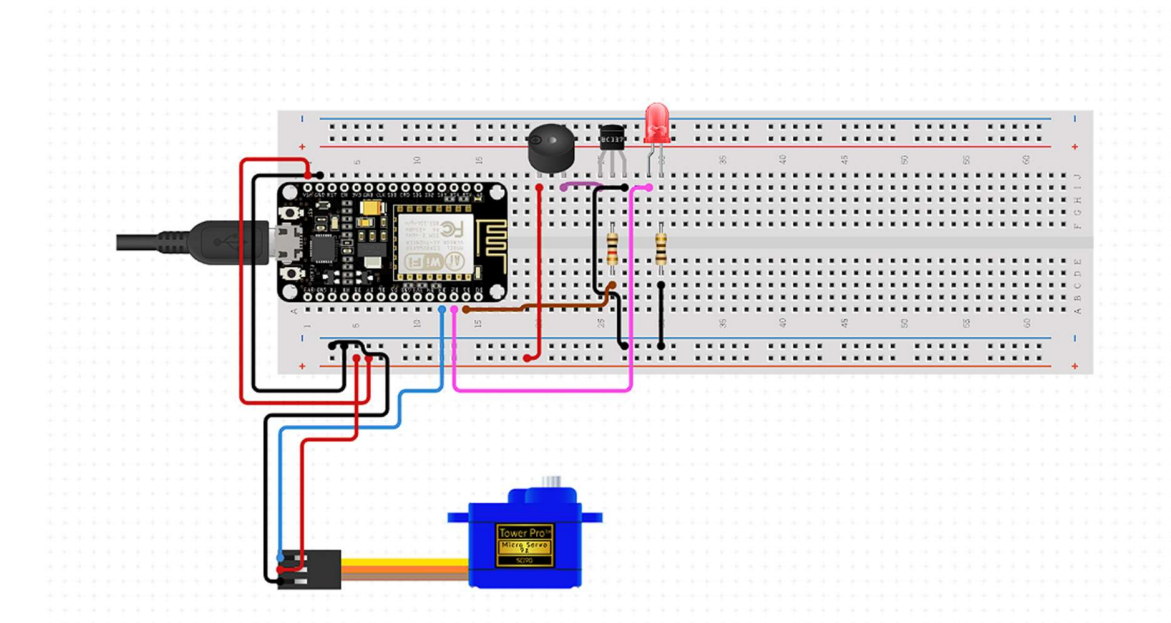


## Circuit Diagram

NodeMCU 8266 (Sensor Node)



NodeMCU 32 (Actuation Node)



## **Conclusion**

In this project, we have developed a window-controlling system using NodeMCU ESP8266, NodeMCU ESP32, LDR sensor, DHT22 sensor, LED, and the Blynk app. The system monitors the temperature, humidity, and light level and automatically opens or closes the window based on the readings. The system also communicates with the user via the Blynk app, allowing them to monitor the environment remotely and control the window manually.

The window-controlling system is just one example of how IoT-based home automation systems can make our lives easier and more convenient. With the help of IoT, we can create smart homes that can automate routine tasks, improve energy efficiency, and enhance home security.

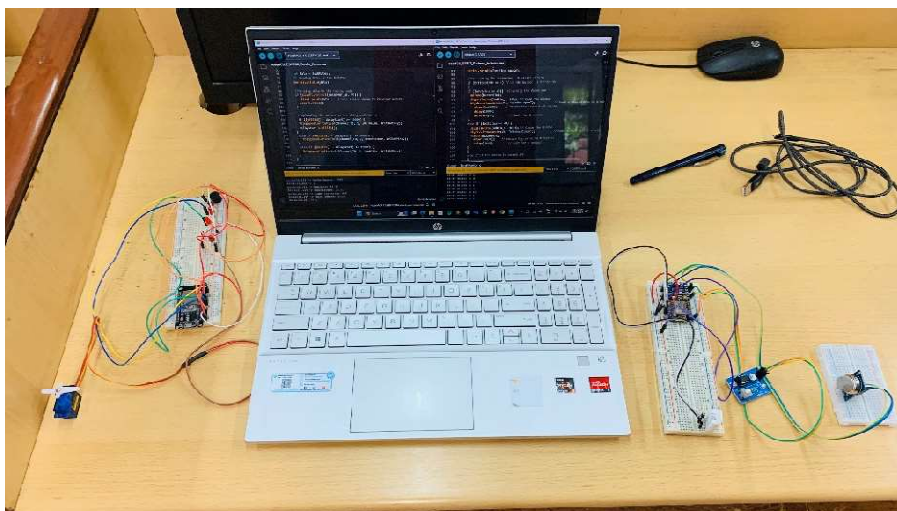
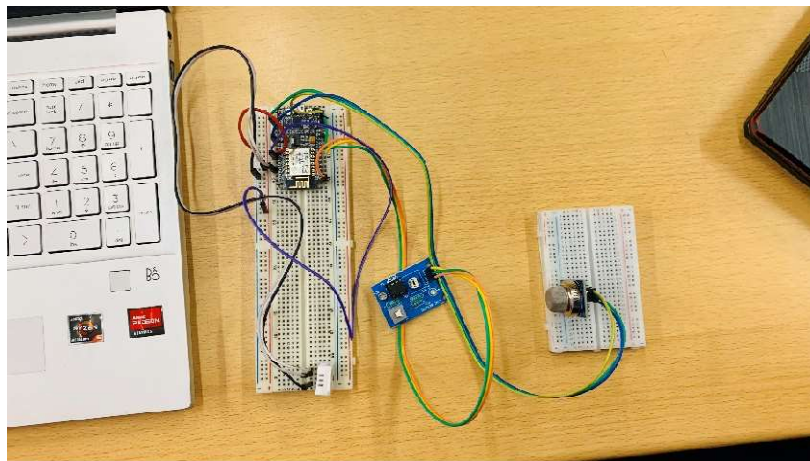
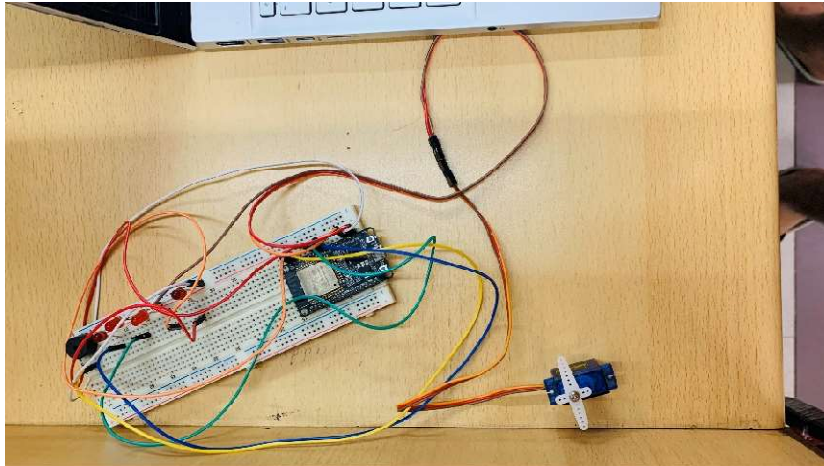
In conclusion, this project has demonstrated how a simple, yet effective home automation system can be developed using NodeMCU ESP8266, LDR sensors, DHT, and the Blynk app. The project's success opens many possibilities for future developments and applications of IoT in home automation. With further enhancements and integration with other sensors and devices, this system can be expanded to provide a complete home automation and security solution.

## Future Scope

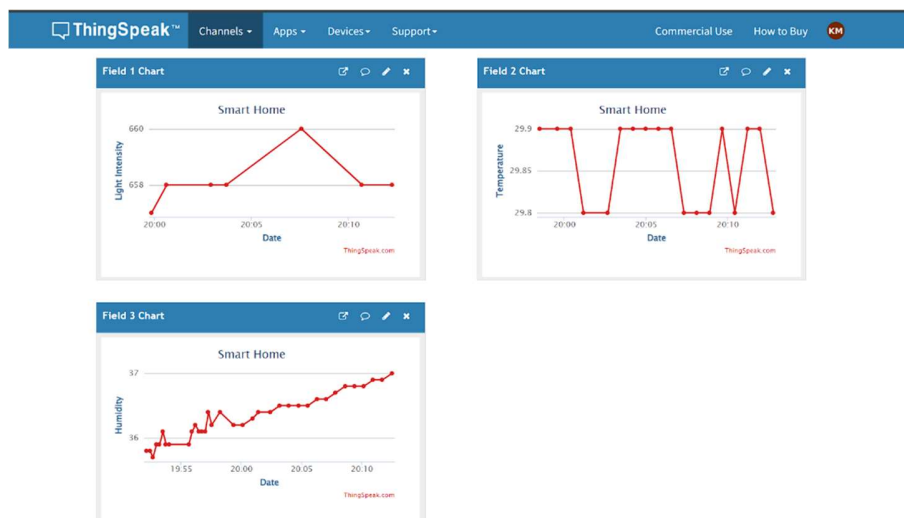
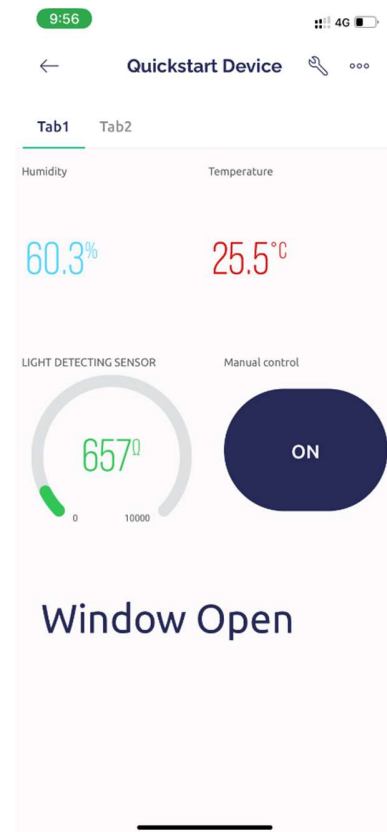
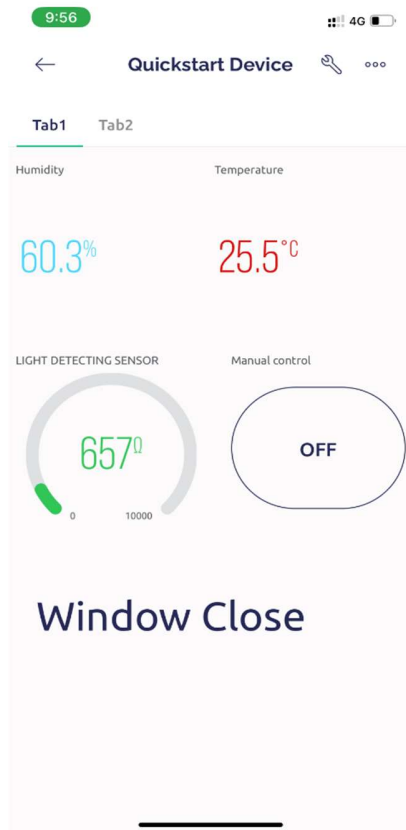
This prototype of a window-controlling system is just the beginning of an advanced home automation system. Here are some future scopes for this project:

1. Integration with other sensors: The system can be extended to include other sensors, such as gas sensors, smoke detectors, and motion sensors, to make the home safer and more secure.
2. Integration with voice assistants: Integration with voice assistants like Amazon Alexa and Google Assistant can make the system more user-friendly. Users can control the window and other devices using voice commands.
3. Integration with machine learning algorithms: Machine learning algorithms can be used to analyze the data collected by the sensors and provide useful insights. For example, if the system detects that the temperature is too high and the humidity is too low, it can automatically turn on the air conditioner and the humidifier.
4. Integration with security cameras: Integration with security cameras can provide a complete home security solution. Users can monitor their homes remotely using the Blynk app and receive alerts in case of any suspicious activity.
5. Integration with renewable energy sources: The system can be integrated with renewable energy sources like solar panels to make the home more energy-efficient and reduce electricity bills.

## Circuit Setup



## Blink App Setup and ThingSpeak



### Bibliography

1. "NodeMCU ESP8266 Documentation." NodeMCU, [nodemcu.readthedocs.io/en/release/index.html](http://nodemcu.readthedocs.io/en/release/index.html).
2. "ESP32 Datasheet." Espressif Systems, [www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
3. "LDR Sensor Overview and Applications." Circuit Digest, 27 Nov. 2019, [circuitdigest.com/article/Ldr-sensor-overview-and-applications](http://circuitdigest.com/article/Ldr-sensor-overview-and-applications).
4. "DHT22 Temperature and Humidity Sensor." Adafruit Industries, [learn.adafruit.com/dht](http://learn.adafruit.com/dht).
5. "Blynk Documentation." Blynk, [blynk.io/](http://blynk.io/).
6. "IoT-Based Home Automation System Using NodeMCU and Blynk Application." International Journal of Advanced Science and Technology, vol. 29, no. 4, 2020, pp. 11571-11582.
7. "ESP8266 and ESP32 Based Home Automation System Using Blynk." IEEE International Conference on Communication and Signal Processing (ICCSP), 2021, pp. 513-517.