Title of the project

"Smart Home Automation by Home Assistant"

Course: **Project / Internship (Phase II)**

Course Code: **CIS499**

Department of Computing and Information System (CIS)

Submitted By:

Md Hasibur Rahman Rafat

Id: 193-16-483

Supervised By:

Dr. Mohammed Nadir Bin Ali

Associate Professor
Department of Computing and Information System

Submission Date: 29 September, 2024

APPROVAL

This Project titled **"Smart Home Automation by Home Assistant"**, Submitted by **Md Hasibur Rahman Rafat**, ID No: **193-16-483** to the Department of Computing & Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information System and approved as to its style and contents. The presentation has been held on 29-09-2024.

BOARD OF EXAMINERS

Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Chairman

Md. Nasimul Kader
Assistant Professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Mehedi Hassan
Lecturer
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Miftah Uddin
Head of SBU, Brain Station 23

External Examiner

Declaration

I hereby declare that, this project has been done by me under supervision of **Dr. Mohammed Nadir Bin Ali, Associate Professor, Department of Computing and Information System (CIS)** of Daffodil International University. I am also declaring that this project or any part of it has never been submitted anywhere else for the award of any educational degree like, B.Sc.

Supervised By

02/10/24

Dr. Mohammed Nadir Bin Ali
Associate Professor
Department of CIS
Daffodil International University

Submitted By

Rafat

Md. Hasibur Rahman Rafat
193-16-483
Department of CIS
Daffodil International University

Acknowledgement

Gratitude First and foremost, I would want to express my gratitude to Almighty Allah for providing me with the strength and wisdom to go on this journey of self-improvement. It would not have been feasible to finish this endeavor without His direction and assistance. I owe a debt of gratitude to Dr. Mohammed Nadir Bin Ali my educational institution for creating a setting full of possibilities and tools that stoked my curiosity and helped me advance in this sector. I have gained more information and skills from this incredible experience, particularly in the areas of new technologies and business practices. My sincere gratitude goes out for all of his help and advice with this endeavor. His words of wisdom and constant encouragement have not only helped me overcome obstacles, but motivated me to reach my highest potential. The main reason for my academic growth has been his confidence in my talents and his readiness to impart knowledge. Many thanks to Dr. Mohammed Nadir Bin Ali sir. I am confident that his calm and collected approach to even the most difficult tasks will aid in providing proper guidance for projects of this magnitude. He has been so patient, so forgiving, and so supporting that I feel like I owe him my life. I appreciate your help and for making this difficult journey to demonstrate my tenacity.

Abstract

The growth of smart home automation is quickly modifying our interaction with our homes, allowing for exceptional control and bespoke experiences. The center of this revolution is Home Assistant, an open-source system that amalgamates several smart home devices into one system. Home Assistant has developed an architecture that promotes high customization, allowing support for countless integrations, from lighting and climate systems to security solutions and media players. This paper investigates the functionalities and perks associated with the use of Home Assistant for smart home automation. This report looks at how the platform takes advantage of Internet of Things (IoT) technologies to form a coordinated and interoperable setting, integrating devices between different vendors. Through live data processing and simple interfaces, Home Assistant supports automation of advanced routines, providing users with the ability to oversee home systems from anywhere, improving home management efficiency over all. Besides, the report investigates the effects of smart home automation related to energy consumption and sustainability. Home Assistant's ability to proficiently utilize resources and offer in-depth analytics helps to greatly reduce the ecological impact of households. Home Assistant reflects an essential resource for today's homes, allowing for the creation of smarter and more efficient living conditions. The open-source design encourages a cooperative community among developers and users, continually broadening its functions and assuring it stays at the leading edge of smart home technology. Not just increasing life quality, Home Assistant contributes to a sustainable future by integrating these advanced systems, making it a key element in current smart home solution

Table of Contents

List of Figures:

List of Tables:

Chapter 1 (Introduction)

1.1 Introduction

The idea of a smart home has changed from a visionary hope to a real option, revolutionizing how residents connect with their homes. Thanks to advancements in IoT technologies, this change is mostly because of its ability to provide uninterrupted communication and coordination among assorted smart devices in a residence. With unprecedented convenience, security, and energy management, home automation systems improve the quality of life for the people who live there.

In the lead of this cutting-edge domain is Home Assistant, a celebrated open-source platform for its ability to be versatile and comprehensively integrate. Users can use Home Assistant to combine a diversity of devices from different manufacturers into a synchronized smart home ecosystem. The flexibility found in this platform is derived from its ability to manage over a thousand integrations including lighting, climate control, security systems, and home entertainment.

1.2 Background and Motivation

The reasons driving smart home automation are varied. Initially, it answers the growing call for convenience, it streamlines normal tasks including the management of thermostats, toggling lights on and off, and monitoring security around the home. Besides that, it increases security by merging surveillance systems with smart locksets and sensors capable of sensing fire, gas leaks, or entries that are not authorized. In addition, it raises energy efficiency by controlling the specific usage of home appliances, reducing unneeded power consumption and helping to lower utility bills.

1.3 Problem Statement

In spite of the many advantages, the market for smart homes is frequently troubled by fragmentation, as products from different vendors adhere to different protocols and platforms. The serious challenge created by interoperability's great lack complicates user experience and limits the potential of home automation.

1.4 Project objective

The main focus of this paper is to analyze how Home Assistant, because of its open-source design and ample integration features, confronts these issues. It attempts to show the way in which Home Assistant can unite various smart home devices into a cohesive, mutually interoperable, and user-friendly system. The platform also intends to bring to light its influence on energy efficiency and sustainability, demonstrating practical applications along with their associated advantages.

1.5 Scope

The range of home automation with Home Assistant includes a variety of approaches to merging, operating, and automating devices and smart home systems. The flexibility alongside extensive integration

features enables Home Assistant to become the central command for a detailed smart home ecosystem. Here are the primary areas covered within this scope:

1. Device Integration

Home Assistant supports a wide range of devices from various manufacturers, allowing seamless integration across different categories, including:

Lighting:

- Smart bulbs, switches, and dimmers.
- Color-changing and adaptive lighting systems.

Climate Control:

- Thermostats and smart HVAC systems.
- Ceiling fans and air conditioners.

Security and Surveillance:

- Smart locks and doorbells.
- Surveillance cameras and motion detectors.
- Intrusion alarms and security systems.

Entertainment:

- Smart TVs and sound systems.
- Media streaming devices (e.g., Chromecast, Fire TV).

Appliances:

- Smart refrigerators, ovens, washing machines, and robotic vacuum cleaners.
- Smart plugs and energy monitoring devices.

2. Protocols and Standards

The platform's wide compatibility includes support for various communication protocols, which helps in unifying devices from different ecosystems:

- Wi-Fi: For most home networks and devices.
- Bluetooth and BLE: For proximity-based integrations.
- Zigbee and Z-Wave: For reliable, low-power mesh networking.
- Matter/Thread: For emerging standards aimed at improving interoperability.
- MQTT, REST API, and Webhooks: For custom integrations and advanced users.

3. Automation and Scripting

Home Assistant offers powerful automation and scripting capabilities:

- Automations: Create complex routines based on conditions, events, or time schedules (e.g., turning off lights when no one is home, adjusting thermostat settings based on weather forecasts).
- Scenes: Pre-configured settings that can control multiple devices simultaneously (e.g., setting lighting and temperature for a movie night).
- Blueprints: Reusable automation templates that can be shared within the community for easy setup.

4. User Interface

User interaction and control are streamlined through various interfaces:

- Lovelace Dashboards: Fully customizable user interfaces for real-time monitoring and control.
- Mobile Apps: Native apps for Android and iOS for on-the-go access.
- Voice Control: Integration with voice assistants like Google Assistant, Amazon Alexa, and Apple's Siri.

5. Energy Management

A significant aspect of smart home automation focused on sustainability and cost savings:

- Real-Time Monitoring: Tracking energy consumption using smart meters and plugs.
- Automated Savings: Scheduling and automating devices to reduce energy use during peak times.
- Integration with Renewables: Managing renewable energy sources like solar panels and storage batteries.

6. Security and Privacy

Emphasis on maintaining robust security and protecting user data:

- Encryption: Ensuring secure communication between devices and the Home Assistant server.
- User Management: Multiple user roles and permissions to secure sensitive controls and information.
- Local Processing: Preference for local control over cloud-based solutions to reduce privacy risks.

7. Community and Extensibility

Home Assistant's open-source nature fosters a vibrant community:

- Add-ons: Extending functionality through community-developed add-ons and integrations.
- Blueprints and Custom Components: Shared automations and customizations available from the community.
- Documentation and Support: Comprehensive resources, forums, and community support for troubleshooting and learning.

8. Real-World Applications and Use Cases

Addressing practical implementations and benefits:

- Residential Homes: From small apartments to large family homes, optimizing comfort and convenience.
- Energy-Efficient Settings: Focusing on homes designed to maximize energy savings and environmental sustainability.
- Special Needs Integration: Tailoring smart home features for accessibility and special needs.

1.6 Overview and Benefits

Overview

- Home Assistant has risen as a key player in the field of smart home automation, offering a complete, adjustable, and easy to use solution for merging an extensive range of smart devices and systems. Home Assistant is designed to serve as the core platform for home automation, helping to manage, control, and automate different smart home features so they work together without a hitch.

  Key Features:

- **Extensive Device Integration**: Leveraging more than a thousand integrations, which feature prominent smart home brands and protocols, Home Assistant is one of the most versatile systems currently available management of their smart home ecosystem.
- **Advanced Automation**: This service enables users with sophisticated automation tools to configure routines, scenes, and triggers that automate everyday tasks and increase comfort.
- **Local Control**: Elevates local processing to an important position, which increases privacy and cuts down on the need for cloud services, vital for users concerned about data protection.
- **Community-Driven**: Due to its open-source framework, Home Assistant receives support from an active community of developers and end-users that reaffirm its growth by regularly adding new integrations and sharing automation blueprints.

Benefits

Adding Home Assistant to a smart home environment offers multiple advantages, boosting different parts of how we live day by day. Below are the key benefits:

1. Enhanced Convenience

- Centralized Control: Home Assistant brings together the management of every smart device into a single easy-to-use interface, accessible through web browsers, mobile apps, or voice commands.
- Automated Routines: Automated repetitive tasks permit users to perform tasks like altering the thermostat when they leave home, switching off lights before bedtime, or brewing coffee each day, giving them both time and effort savings.
- Voice Assistance: Integration without seams with voice assistants including Amazon Alexa, Google Assistant, and Apple Siri makes it possible to control smart home devices hands-free.

2. Improved Security

- Comprehensive Security Monitoring: The ability to integrate with security cameras, smart locks, door/window sensors, and alarms allows Home Assistant to supply a complete outlook on home security.
- Real-Time Alerts: Immediate responses to potential dangers depend on instant alerting for security events, including those showing motion detection, door/window operation, smoke emergencies, or excess CO2 levels.
- Remote Access: Being away from home does not prevent users from enjoying comfort due to their ability to control and track their home security remotes.

3. Energy Efficiency and Cost Savings

- Energy Monitoring: Smart meters and plugs enable Home Assistant to keep an eye on existing energy usage, allowing users to find and reduce energy waste.
- Automated Energy Management: An automated approach to heating, cooling, and lighting systems tied to occupancy and time of day is likely to greatly reduce energy costs and lessen the household's ecological footprint.
- Integration with Renewable Energy: Realizing the greatest deployment of renewable resources along with energy savings requires the skillful management of solar panels, battery storage, and electric vehicle chargers.

4. Personalization and Flexibility

- Custom User Interfaces: Users of Lovelace dashboards are able to design personalized interfaces that conform to their specific needs and preferences, giving an exclusive and individualized user experience.
- Extensive Customization: People skilled in YAML and Python scripting are able to design very thorough automation rules and integrations because of Home Assistant's support.
- Adaptability: As it continually progresses with new features and integrations coming from the community, the platform continues to work effectively in support of the latest smart home technologies.

5. Scalability and Future-Proofing

- Expandable Ecosystem: The user needs can grow with Home Assistant, seamlessly integrating new devices and systems when they are incorporated into the smart home.
- Support for Emerging Standards: Following many updates and community contributions, Home Assistant is able to remain in step with developing technologies and standards, including Matter/Thread, to ensure its long-term sustainability.

Chapter 2 (Literature Review)

Home automation has made impressive progress in recent decades thanks to fast developments in IoT technologies, wireless communication protocols, and artificial intelligence. The literature review presented here reviews key findings, technical trends, and architecture in relation to smart home automation, concentrating on the attributes, benefits, and hurdles associated with frameworks such as Home Assistant.

Evolution of Home Automation

1. Early Developments:
   - • At the beginning, home automation systems targeted simplifying principal controls of lighting and HVAC systems. In the 1970s, the first extensively adopted protocol allowed users to control household appliances from afar through the method of power line communication. As the internet and wireless technologies developed, so did automated homes that can operate more complex systems accessible via both web interfaces and mobile apps remotely. The 1970s, permitting remote control of household devices using power line communication.
   - With the advent of the internet and wireless technologies, home automation expanded to include complex systems and remote-control capabilities via web interfaces and mobile applications.

2. Modern Smart Home Systems:
   - Contemporary smart home solutions support many devices and work with advanced wireless communication standards including Zigbee, Z-Wave, Wi-Fi, and, lately, Matter/Thread. The sophistication of these systems comes from the features of robust automation and customization flexibility available to users.
   - Companies such as Google, Amazon, and Apple have stepped into the sector, presenting their ecosystems (Google Home, Amazon Alexa, Apple HomeKit), each advocating for smart hubs that connect a variety of device systems are much more sophisticated, featuring robust automation and user customization capabilities.

Key Technologies and Protocols

1. Internet of Things (IoT):
   - The root of smart home technologies, IoT makes it possible for devices to communicate and receive control signals over the internet. Ashton (2009) points out in their research that the Internet of Things may facilitate the transformation of standard objects into linked systems that improve both lifestyle convenience and efficiency.
   - Palattella et al. (2016) summarize in their review the standards and protocols related to the Internet of Things, with a particular focus on wireless communication technologies Zigbee and Z-Wave in automating smart homes.

2. Wireless Communication Protocols:

- Kocakulak and Butun (2017) conducted studies which reveal that Zigbee, Z-Wave, and other protocols provide reliable, low-power alternatives to home automation, suitable for the needs of various settings.
- The release of Matter (originally known as Project CHIP) intends to bring together these efforts by creating a single IP protocol that encourages interoperability between different devices and platforms (CSA, 2021).

Home Assistant and Open-Source Platforms

1. Overview of Home Assistant:
   - Paulus Schoutsen launched Home Assistant in 2013 as an open-source platform that has won over users with its wide range of integration possibilities and its flexibility.

The platform supports a vast range of devices and services, allowing users to customize their automation scripts and interfaces extensively.

2. Flexibility and Customization:
   - In their 2020 report, Moutinho et al. emphasize the requirement for customizing smart home systems, indicating that tools, such as Home Assistant, deliver configurable options that improve the enjoyment of users and the productivity of the systems.
3. Community and Development:
   - Thanks to the involved Home Assistant community, the platform experiences increased traction by means of new integrations, blueprints, and custom components, which Bhattacharjee and Das (2021) have skillfully investigated.

Benefits of Smart Home Automation

1. Convenience and Comfort:
   - Research from Davidoff et al. (2006) investigates ways in which automated routines and smart controls can greatly improve convenience and comfort levels in daily living.
   - The work done by Lee et al. (2020) indicates that smart homes create a higher quality of life by automating common tasks and achieving a coherent living experience.
2. Energy Efficiency:
   - A variety of studies, including those of Deng et al. (2018), underline the promise of smart home systems in optimizing energy consumption, which may lead to important cost reductions and a lessened environmental effect.
   - • These results highlight the capabilities of Home Assistant for energy management, comprising real-time monitoring and the automation of energy-consuming device control, in support of sustainable living.

3. Security:
   - As indicated by Kim et al. (2019), smart home systems that integrate security enhance intrusion protection and improve how response to emergencies occurs.

- The combination of different security devices into one comprehensive system by Home Assistant guarantees effective home security management.

Challenges and Considerations

1. Interoperability Issues:
   - • According to Babar et al. (2010), interoperability is a major challenge. With many vendors and protocols, the achievement of seamless integration continues to be a key issue.
   - Home Assistant resolves this challenge by supporting several protocols and delivering bridging alternatives for devices that ordinarily wouldn't work together.
2. Privacy and Security:
   - The field of research has widely analyzed the fears related to data privacy and security within smart home. Zeng et al. (2017) stress in their research the critical need for solid data protection methods.
   - Home Assistant's focus on improving local control helps to lower the need for cloud data processing, thus improving privacy and limiting a number of security vulnerabilities.
3. User Adoption and Usability:

   - Brush et al. (2011) have found that usability acts as a barrier to the general acceptance of smart home technologies. Setting up and overseeing complex systems might be a difficulty for users.
   - Home Assistant is striving to make its user interface easier to use and to produce comprehensive documentation and community support in order to resolve these usability challenges.

Chapter 3 (Theory)

3.1 Internet of Things (IoT)

The Internet of Things (IoT) represents a core technological breakthrough that ties devices, allowing them to communicate and work together across the internet. The existence of this connectivity allows for the automation of many systems in addition to remote control, thereby increasing both efficiency and convenience.

3.1.1 Features of IoT

- Connectivity: IoT devices are connected through various network protocols, enabling communication and data exchange.
- Interactivity: Devices can interact with each other and with users, providing real-time monitoring and control.
- Automation: IoT systems enable the automation of tasks and processes, reducing the need for human intervention.
- Data Analytics: IoT devices collect and analyze data, providing insights that can be used for optimizing operations.
- Scalability: IoT systems are scalable, supporting a large number of connected devices and expanding as needed.
- Integration: IoT can be integrated with other technologies, such as AI and machine learning, to enhance capabilities.

3.1.2 Advantages of IoT

- Improved Efficiency: Automation and real-time data analysis streamline operations and reduce manual effort.
- Cost Savings: Optimized operations and reduced resource consumption lead to significant cost savings.
- Enhanced Convenience: Remote monitoring and control provide users with convenience and flexibility.
- Data-Driven Decision Making: IoT systems generate valuable data insights that inform better decision-making.
- Increased Safety and Security: IoT devices enhance safety through real-time monitoring and alerting systems.
- Personalization: IoT enables personalized user experiences by tailoring system responses to individual preferences.

3.1.3 Disadvantages of IoT

- Security Concerns: IoT devices are susceptible to cyberattacks, posing risks to data privacy and security.
- Complexity: The integration of diverse devices and protocols can be complex and challenging to manage.
- Interoperability Issues: Ensuring compatibility among various IoT devices from different manufacturers can be difficult.
- Data Overload: The vast amount of data generated by IoT devices can be overwhelming and require robust data management solutions.
- Cost of Implementation: Initial costs for setting up IoT infrastructure can be high, including the purchase of devices and networking equipment.

3.1.4 Application Grounds of IoT

- Smart Homes: Home automation systems for lighting, climate control, security, and entertainment.
- Healthcare: Remote patient monitoring, wearable health devices, and smart medical equipment.
- Industry 4.0: Automation of manufacturing processes, predictive maintenance, and supply chain management.
- Agriculture: Precision farming technologies, soil sensors, and automated irrigation systems.
- Transportation: Fleet management, smart traffic systems, and connected vehicles.
- Retail: Inventory management, smart shelves, and personalized shopping experiences.

3.1.5 IoT Technologies and Protocols

- Wi-Fi: Commonly used for connecting IoT devices to local networks and the internet.
- Bluetooth: Ideal for short-range communication between devices.
- Zigbee: A low-power, low-data-rate wireless protocol for home automation and industrial applications.
- Z-Wave: Similar to Zigbee, used primarily in home automation.
- LoRaWAN: Long-range, low-power wireless communication for wide-area networks.
- MQTT: Lightweight messaging protocol for small sensors and mobile devices.
- HTTP/HTTPS: Standard protocol for web communication, used for IoT devices that connect to web servers.

3.2 NodeMCU

NodeMCU is an open-source development board and firmware based on the ESP8266 Wi-Fi module. It is widely used in IoT projects due to its low cost, ease of use, and robust functionality.

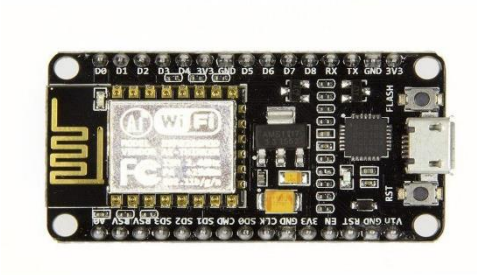


Figure 1. Picture of NodeMCU.

3.2.1 Pin Configuration of NodeMCU Development Board

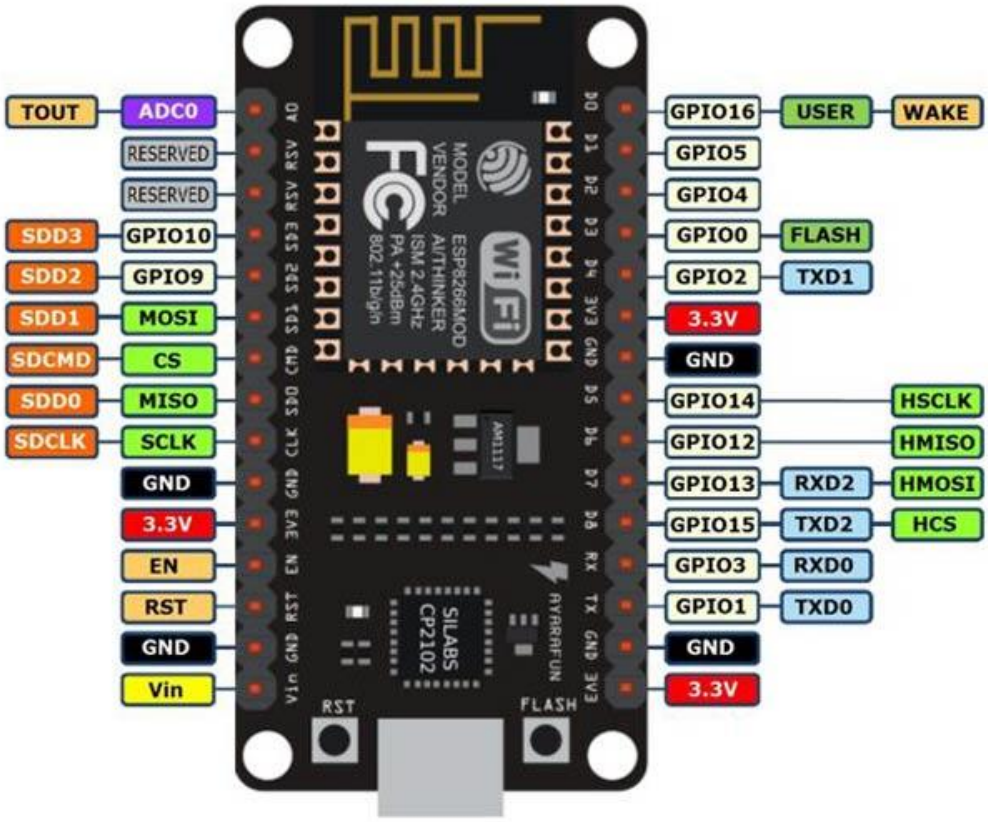


Figure 2. NodeMCU Pinout.

- GPIO Pins: General Purpose Input/Output (GPIO) pins for connecting sensors and actuators.
- ADC Pin: Analog-to-Digital Converter (ADC) pin for analog input.
- Power Pins: 3.3V, Vin, and Ground pins for powering the board and connected devices.

- UART Pins: Universal Asynchronous Receiver/Transmitter (UART) pins for serial communication.
- SPI Pins: Serial Peripheral Interface (SPI) pins for communication with peripherals like sensors and displays.
- I2C Pins: Inter-Integrated Circuit (I2C) pins for communication with multiple devices using two wires.

3.3 ESP32

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth capabilities. It offers enhanced performance and versatility, making it suitable for complex IoT applications.

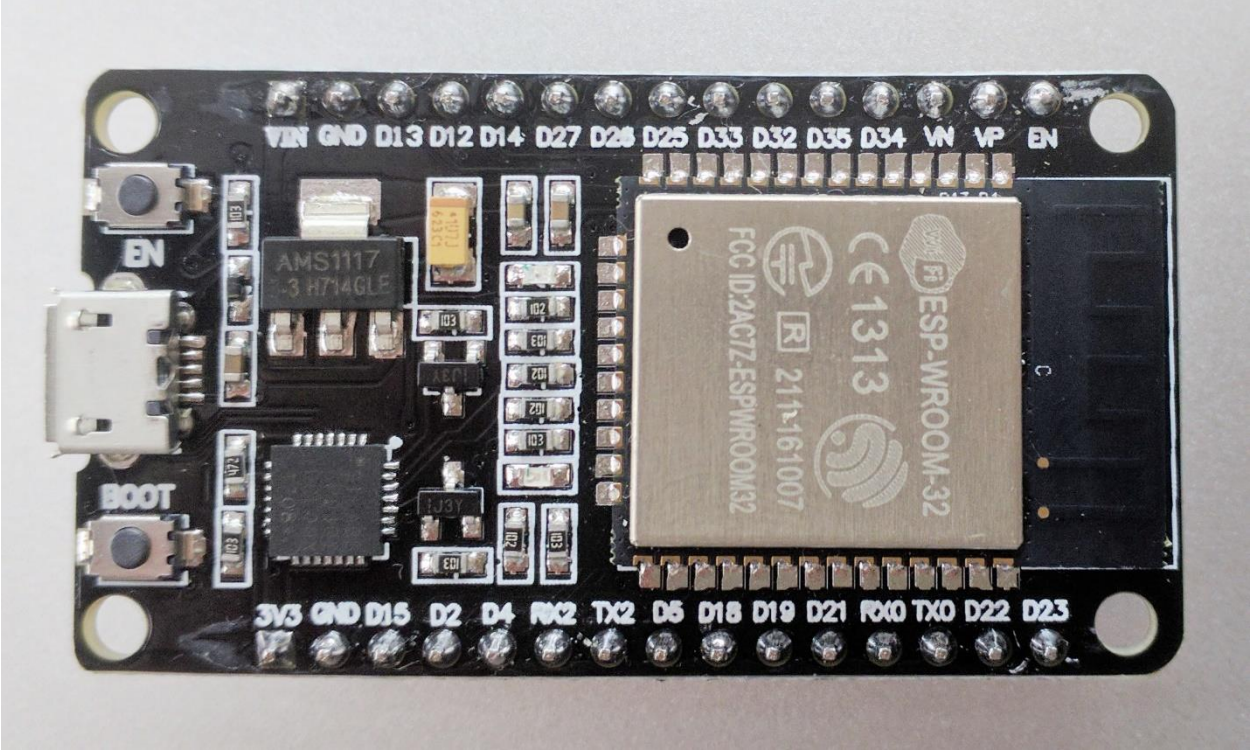


Figure 3. Picture of ESP32.

3.3.1 Pin Configuration of ESP32

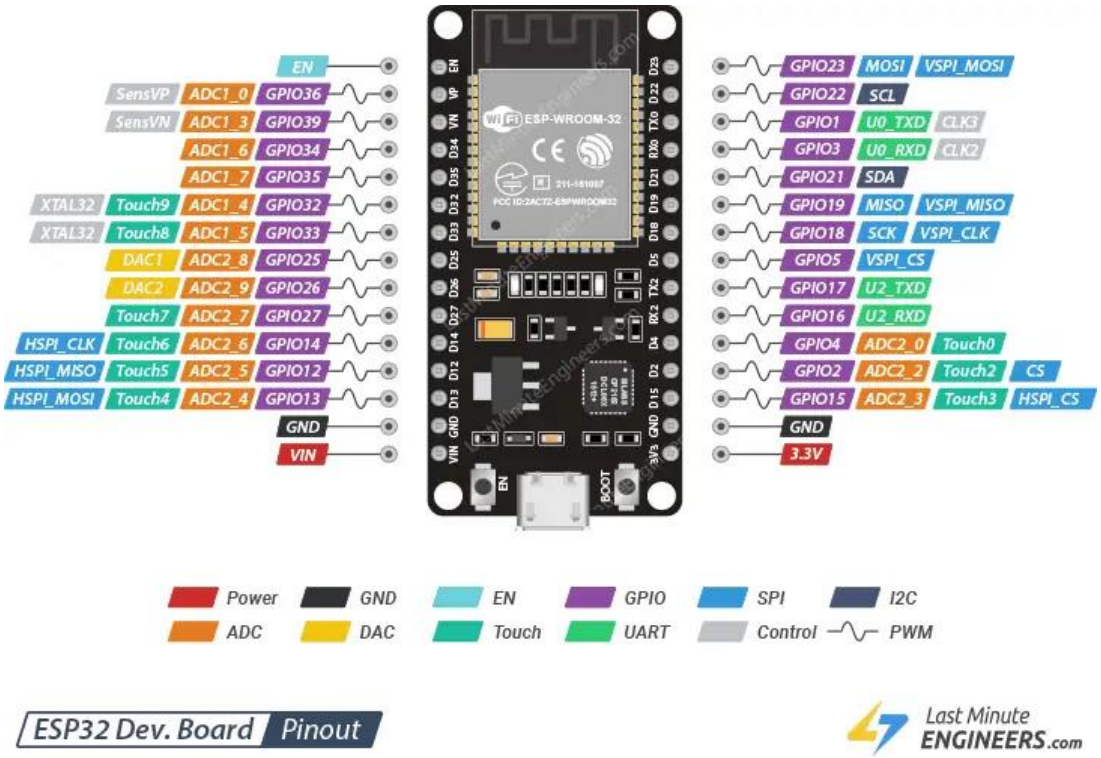


Figure 4. ESP32 Pinout.

- GPIO Pins: Multiple GPIO pins for connecting various sensors and actuators.
- Analog Pins: Multiple ADC channels for analog input.
- DAC Pins: Digital-to-Analog Converter (DAC) pins for analog output.
- Power Pins: 3.3V, 5V, and Ground pins for powering the board and connected devices.
- UART Pins: Two sets of UART pins for serial communication.
- SPI Pins: SPI interface pins for communication with peripherals.
- I2C Pins: I2C interface pins for communication with multiple devices.
- PWM Pins: Pulse-Width Modulation (PWM) pins for controlling devices like motors and LEDs.
- Capacitive Touch Pins: Pins for touch input.

3.4 Overview of the Project

The project aims to develop a comprehensive smart home automation system using Home Assistant, integrating multiple devices through ESP32 and ESP8266 microcontrollers.

Components and Integrations:

- ESP32 with 4-Channel Relay: Control high-power appliances such as lighting and fans.
- ESP8266 with DHT11 Sensor, Buzzer, and MQ-02 Gas Sensor: Monitor temperature, humidity, and air quality while providing alerts for unsafe conditions.
- ESP8266 with RC522 RFID Reader: Implement secure access control through NFC tags.

Goals:

- Automation: Create automated routines for managing home devices based on user preferences and environmental conditions.
- Security: Enhance home security through real-time monitoring, alerts, and access control systems.
- Energy Management: Optimize energy consumption by monitoring and controlling energy-intensive devices.
- User Convenience: Provide a user-friendly interface through Home Assistant for centralized control of the smart home environment.

Implementation Steps:

- Device Setup: Configure ESP32 and ESP8266 microcontrollers with necessary sensors and relays.
- Integration with Home Assistant: Connect and integrate devices with Home Assistant for centralized management.
- Automation Scripts: Develop and implement automation scripts and routines to enhance functionality.
- User Interface: Design and customize Home Assistant dashboards for ease of use.
- Testing and Optimization: Test the system to ensure reliability and optimize performance.

This project showcases practical applications of IoT in creating a smart home environment, highlighting the versatility and benefits of using Home Assistant with ESP32 and ESP8266 microcontrollers.

Chapter 4 (Hardware modelling and setup)

4.1 Main Features of the Smart Home Automation Prototype

This smart home automation prototype takes advantage of the solid features of Home Assistant along with the adjustable ESP32 and ESP8266 microcontrollers to form a well-organized and functional home automation system. As follows are the principal features of this prototype:

1. Centralized Control

- Home Assistant Integration: All devices are integrated into the Home Assistant platform, providing a centralized interface to monitor and control various aspects of the home automation system.
- Customizable Dashboards: Lovelace Dashboards in Home Assistant allow for a personalized user interface, making it easy to view and manage connected devices.

2. Multi-Device Integration

- ESP32 with 4-Channel Relay:
  - Control multiple high-power devices such as lights, fans, and other appliances.
  - Enable automated routines and manual control via Home Assistant.

- ESP8266 with DHT11 Sensor, Buzzer, and MQ-02 Gas Sensor:
  - Environmental Monitoring: Real-time monitoring of temperature, humidity, and air quality.
  - Safety Alerts: Gas sensor detects harmful gases, triggering the buzzer and sending alerts through Home Assistant.

- ESP8266 with RC522 RFID Reader:
  - Access Control: Secure access to the home using NFC tags.
  - Logging and Monitoring: Track entry events and monitor access through Home Assistant.

3. Automation and Scheduling

- Automated Routines: Create sophisticated automation routines based on time, events, or conditions (e.g., turning on lights when motion is detected, adjusting thermostat based on outdoor temperature).
- Scene Configuration: Set up scenes to simultaneously control multiple devices (e.g., "Good Night" scene to turn off lights, lock doors, and lower thermostat).

4. Energy Efficiency and Management

- Energy Monitoring: Use smart plugs and other devices to track and analyze energy consumption in real-time.

- Energy Saving Automations: Implement routines to reduce energy usage during peak times (e.g., turning off non-essential devices when not in use).

5. Security and Alerts

- Real-Time Monitoring: Utilize sensors and cameras for real-time monitoring of security events.
- Instant Notifications: Receive immediate alerts for security breaches (e.g., door/window sensors, gas leaks).
- Buzzer Alerts: Audible warnings provided by the buzzer for critical events.

6. Remote Access and Control

- Mobile Access: Control and monitor the home automation system remotely via Home Assistant's mobile app, available on Android and iOS.
- Secure Remote Control: Ensure that all remote communications are encrypted and secure.

7. Voice Assistant Integration

- Voice Control: Integrate with popular voice assistants like Amazon Alexa, Google Assistant, and Apple Siri for hands-free operation of smart home devices.
- Voice Commands: Execute predefined automation scripts and control devices using simple voice commands.

8. User-Friendly Interface

- Intuitive Control Panels: Customizable control panels to suit user preferences and enhance ease of use.
- Interactive Elements: Real-time updates on dashboards, interactive buttons, and responsive design for a smooth user experience.

Specific Features Highlighted by Implementation

ESP32 with 4-Channel Relay

- Versatile Control: Manage multiple high-power devices connected to the relay board, from lighting systems to electric appliances.
- Flexible Automation: Combine relay operations with other smart home devices for complex, coordinated actions (e.g., turning off all lights and appliances upon exiting the home).

ESP8266 with DHT11 Sensor, Buzzer, and MQ-02 Gas Sensor

- Comprehensive Environmental Data: Accurate readings of temperature and humidity levels help automate climate control systems for optimal comfort.
- Safety Measures: The gas sensor's integration enhances home safety by detecting potential gas leaks and triggering immediate responses (e.g., turning off the gas supply or activating an exhaust fan).

ESP8266 with RC522 RFID Reader

- Secure Entry System: Provides a modern solution for access control, replacing traditional keys with NFC tags.
- Access Logs: Detailed logging of NFC tag usage to monitor and analyze access patterns, ensuring security is maintained.

4.2 Project Layout

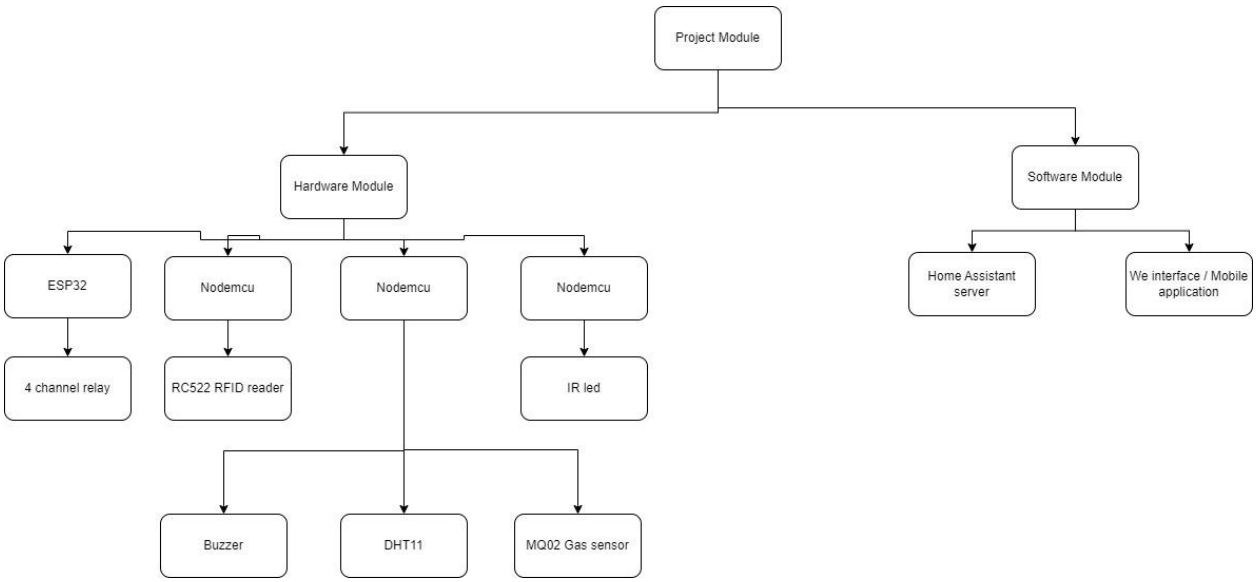


Figure 5. Full layout of the project.

4.3 Component required

Table 1. Required components for the project.

SL No	Component	Quantity
1	NodeMCU	3
2	Esp32	1
3	4 Channel relay	1
4	Mq-02 Gas sensor	1
5	5V Buzzer	1
6	MFRC52 RFID reader Module	1
7	IR led	1

4.4 Setting up the system

4.4.1 Downloading and run the Home Assistant OS on Virtual box:

Download preferred OS for your system that will run the OS from this link

https://github.com/home-assistant/operating-system/releases/tag/13.1

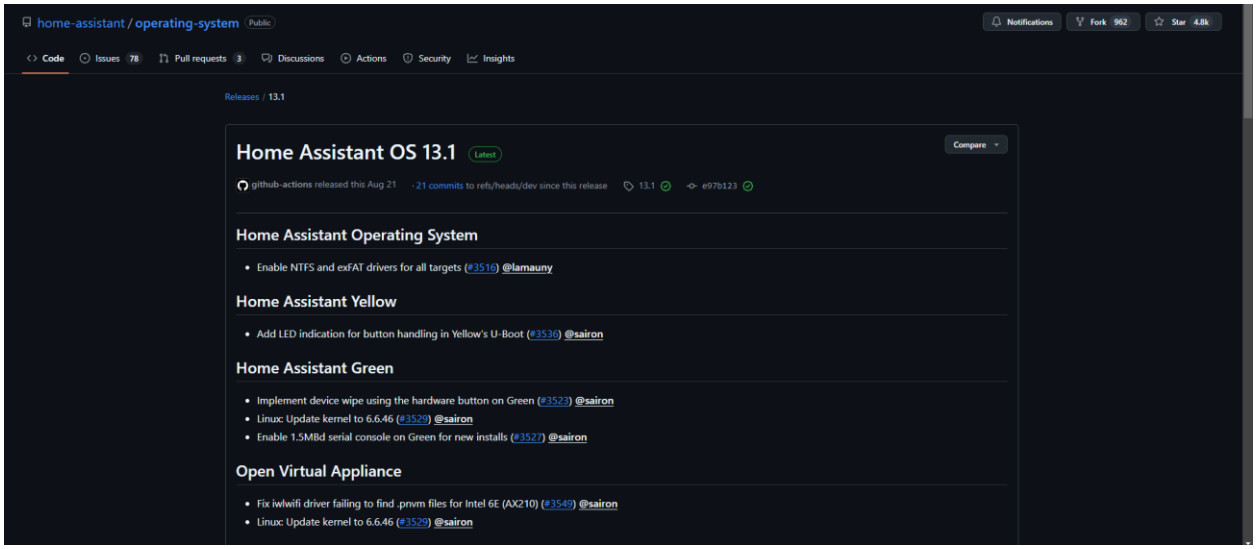


Figure 6. Home assistant OS download page.

After downloading create new virtual machine:

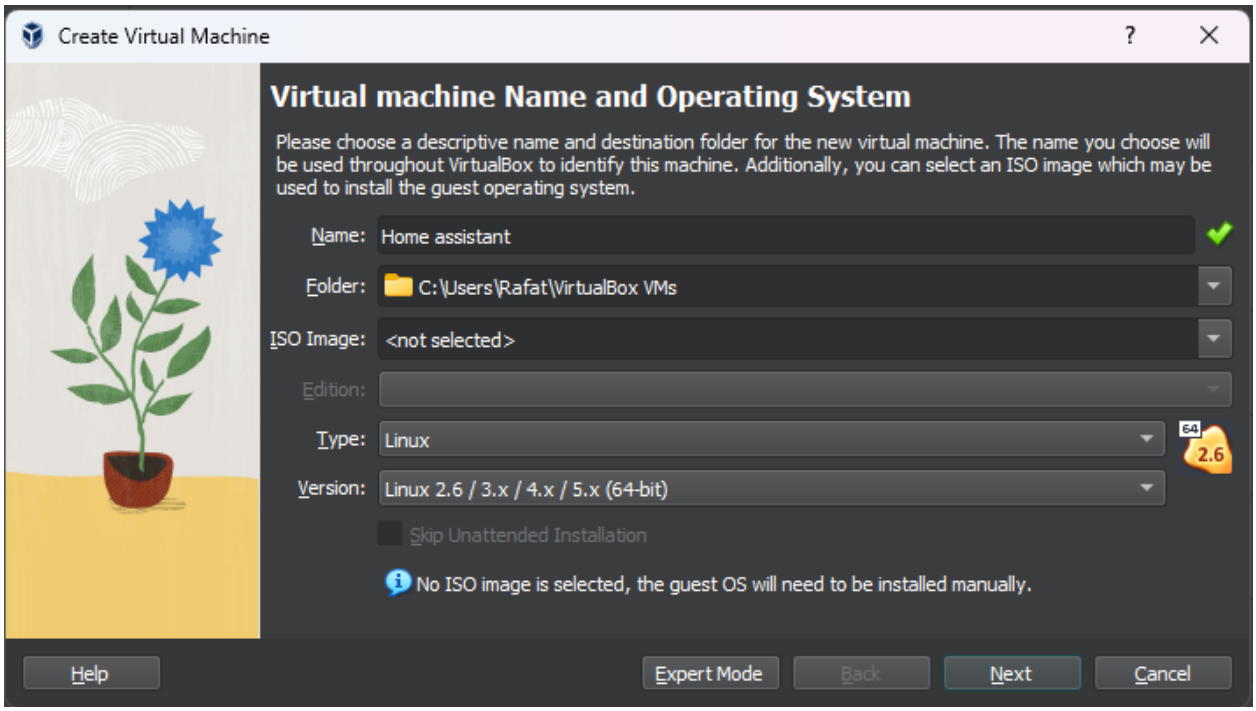


Figure 7. Creating virtual machine for the Home assistant server.

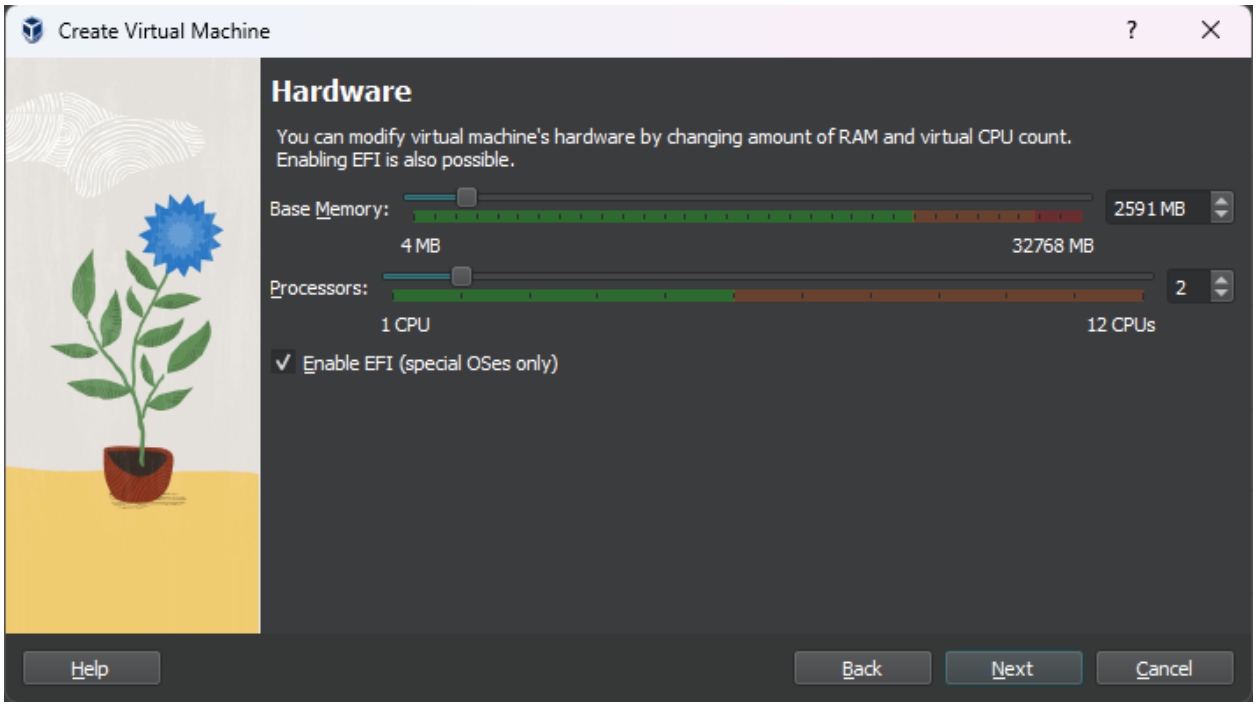


Figure 8. Creating virtual machine for the Home assistant server.

Select the downloaded Home assistant vdi disk

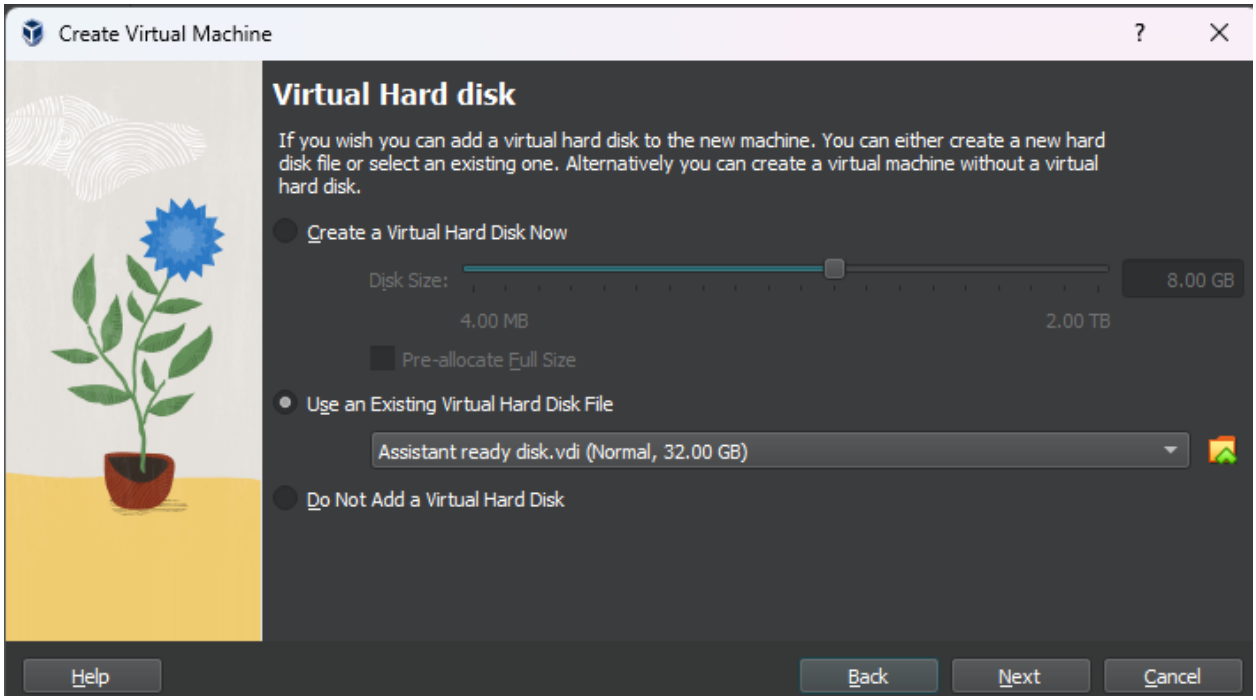


Figure 9. Creating virtual machine for the Home assistant server.

Now create the machine

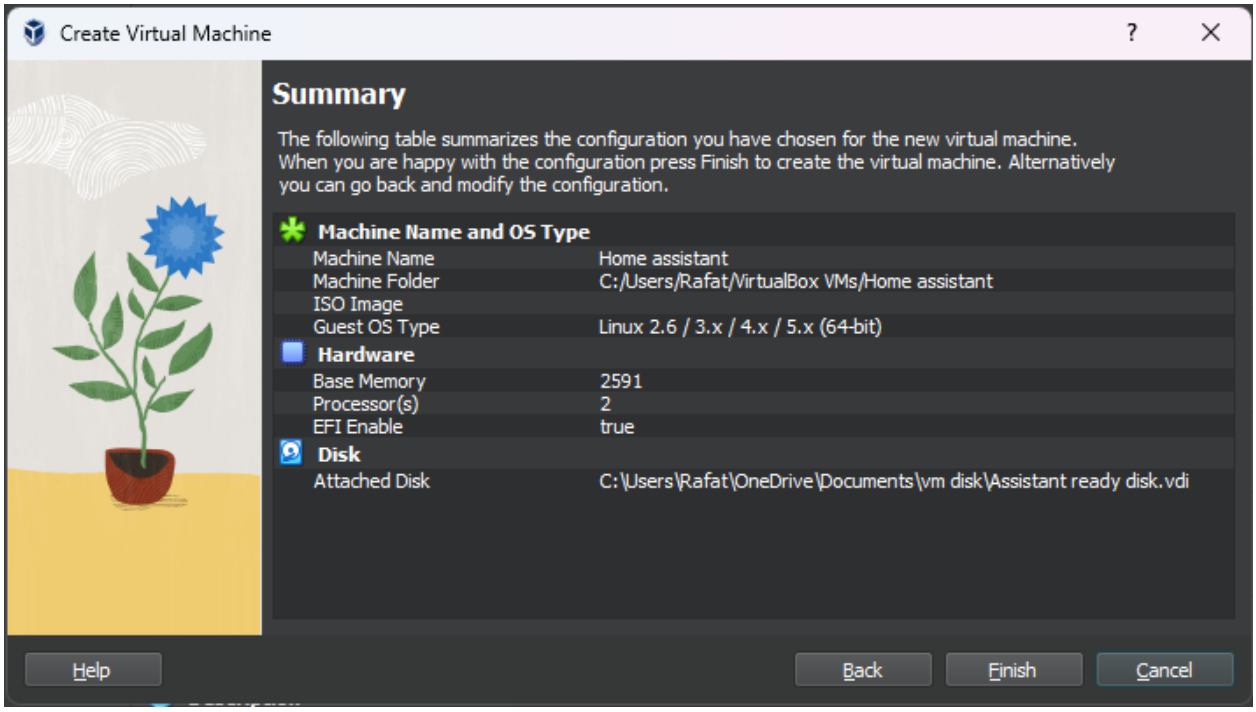


Figure 10. Creating virtual machine for the Home assistant server.

Now start the virtual machine

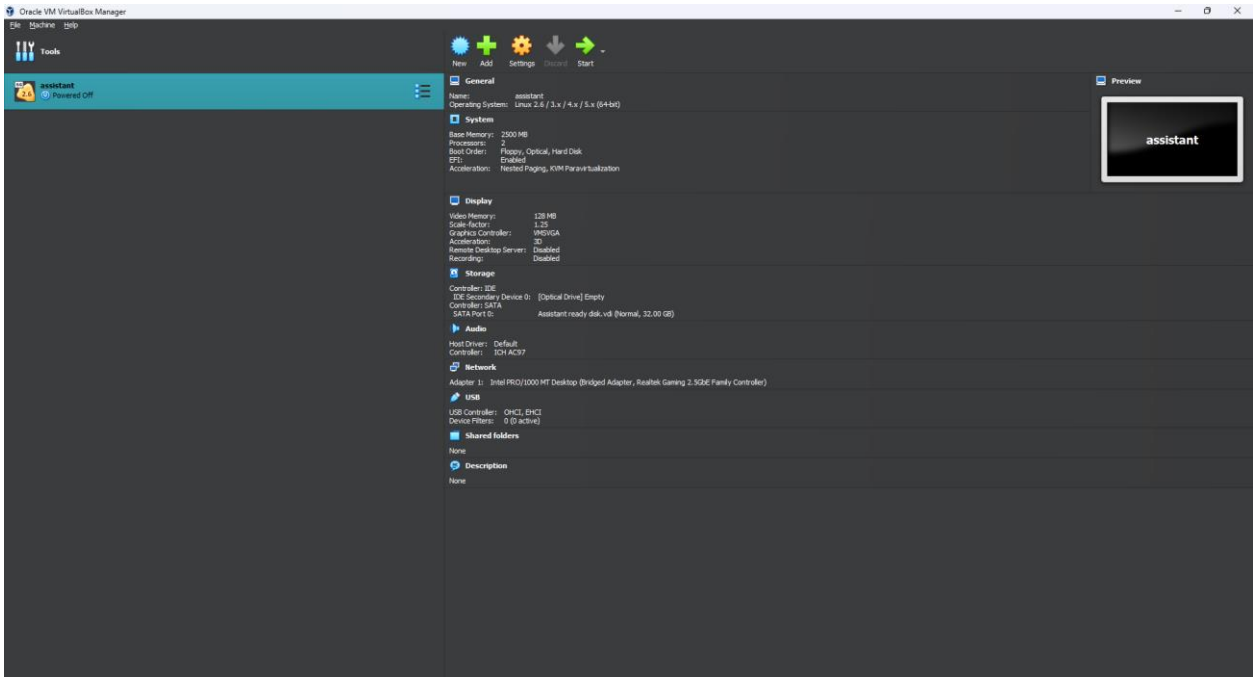


Figure 11. Start the Home assistant server.

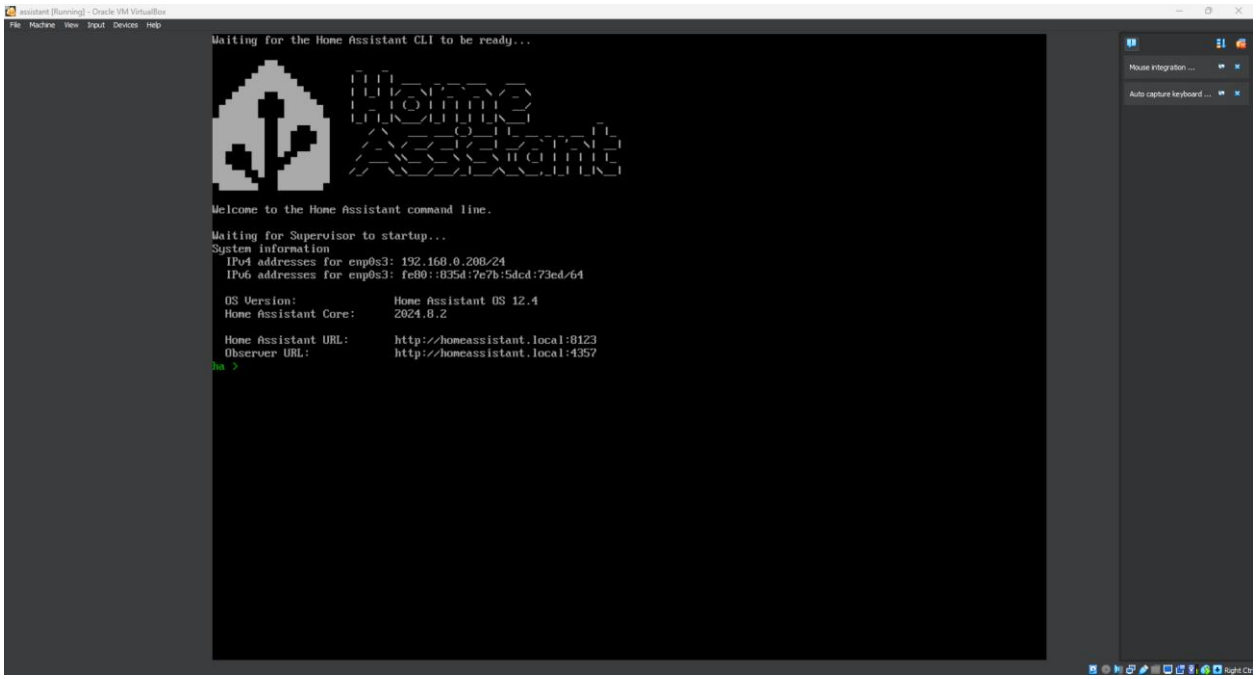


Figure 12. Home assistant server started.

Now go to browser and search the address given by the home assistant server. Mine is:

http://192.168.0.208:8123

There is also application for Android and IOS called Home assistant to control your smart home from your mobile devices.

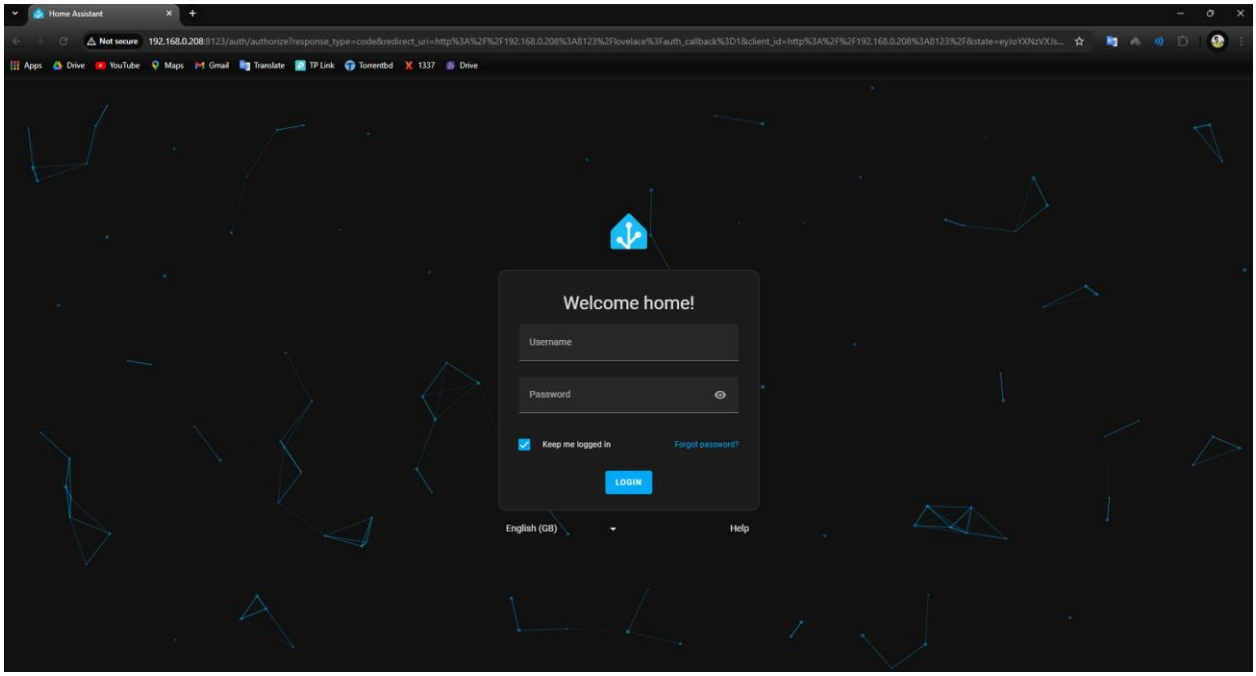


Figure 13. Home assistant login page.

I already created the username and password that's why it's asking me for login.

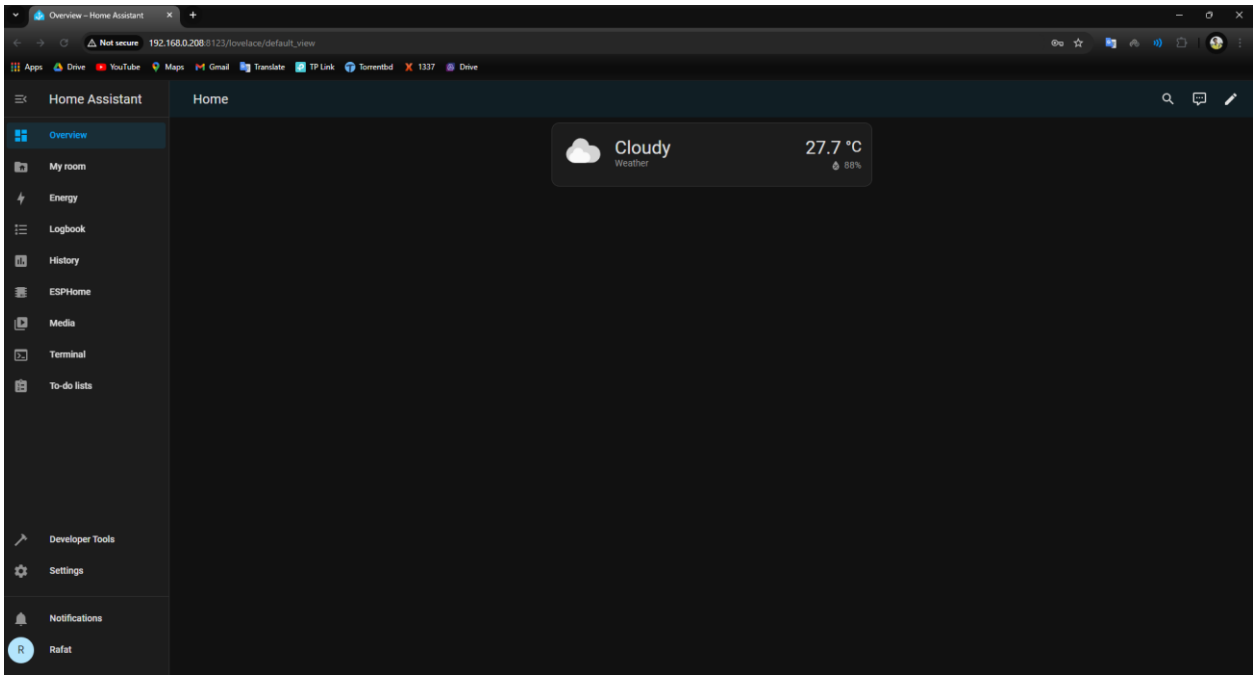


Figure 14. Home assistant dashboard.

Now I need to install ESPHome to communicate my esp devices to the server.

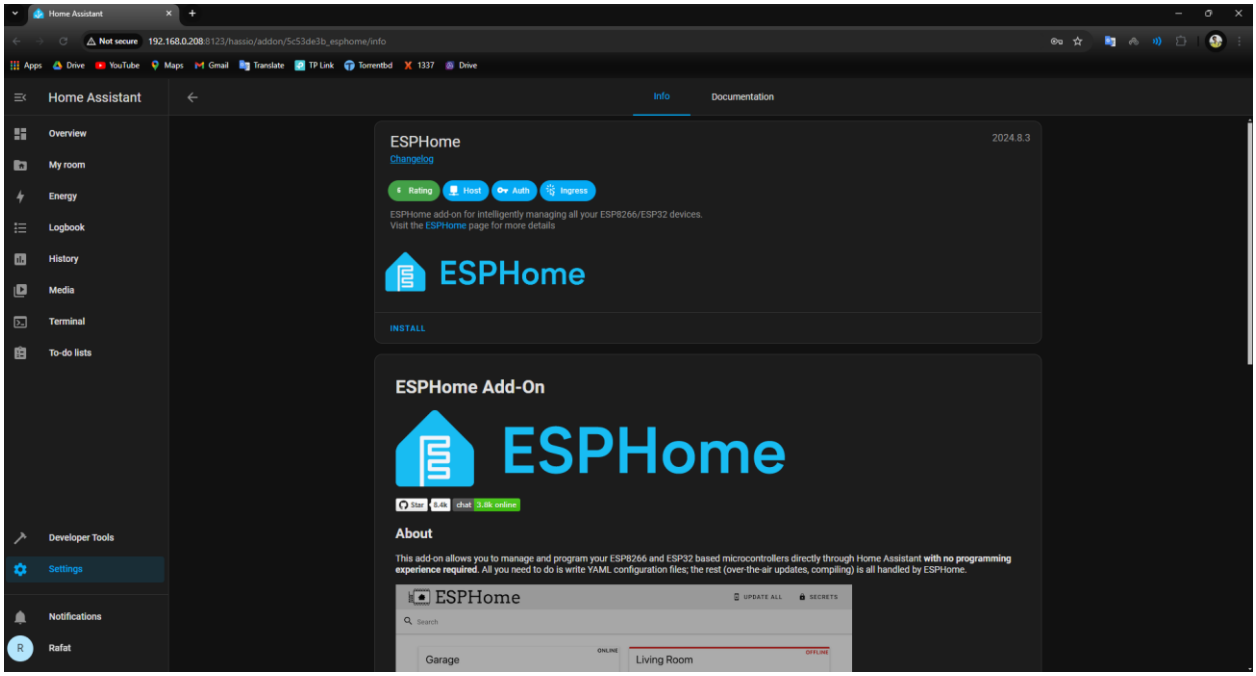


Figure 15. Installing ESPHome on Home Assistant.

Go to setting> Add-ons> Add-on store> ESPHome. Then hit install. It will install ESPHome to the HAOS (Home assistant Operating system).

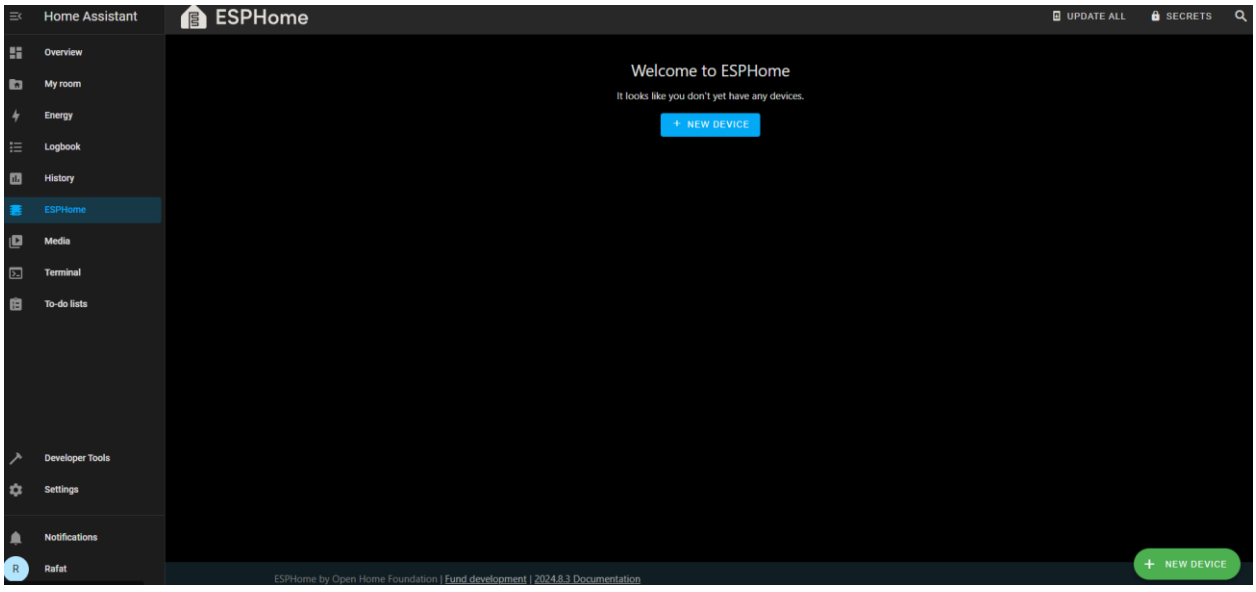


Figure 16. ESPHome dashboard.

I already created and added all my custom devices in the ESPHome.

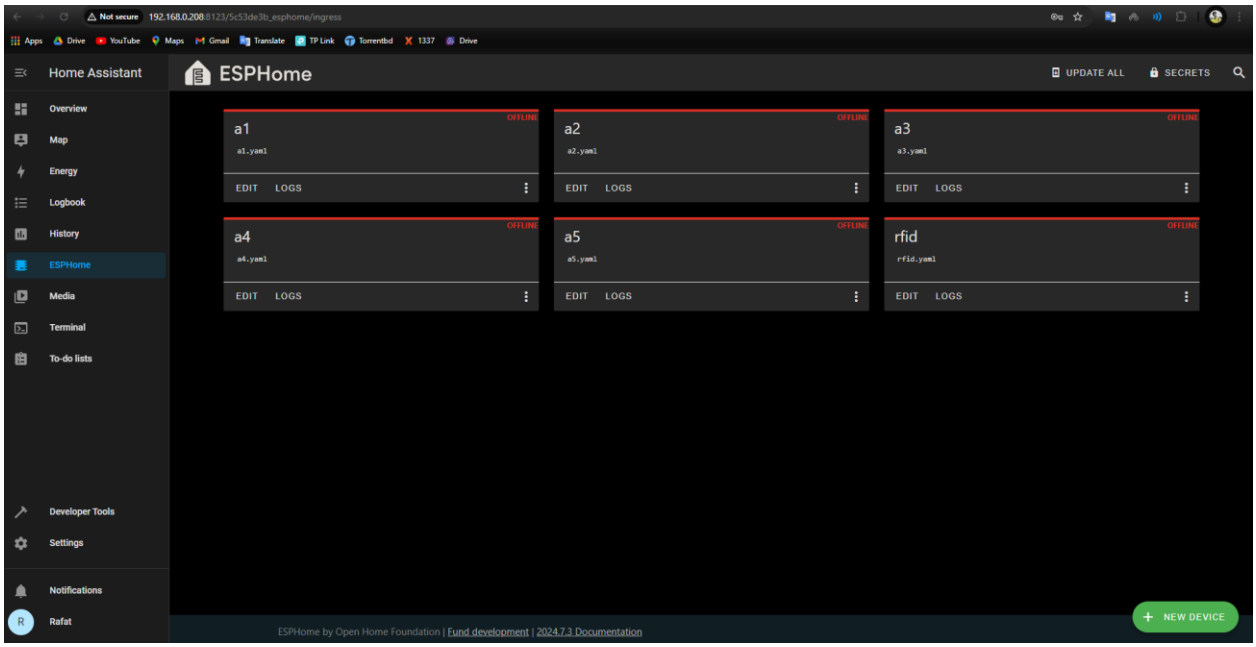


Figure 17. ESPHome dashboard after configured all board.

4.4.1 Driver installation for hardware interfacing

Download and install CP210x driver for nodemcu:

Figure 18. Installing Drivers For microcontroller.

After download and unzip right click on silabser.inf and hit install

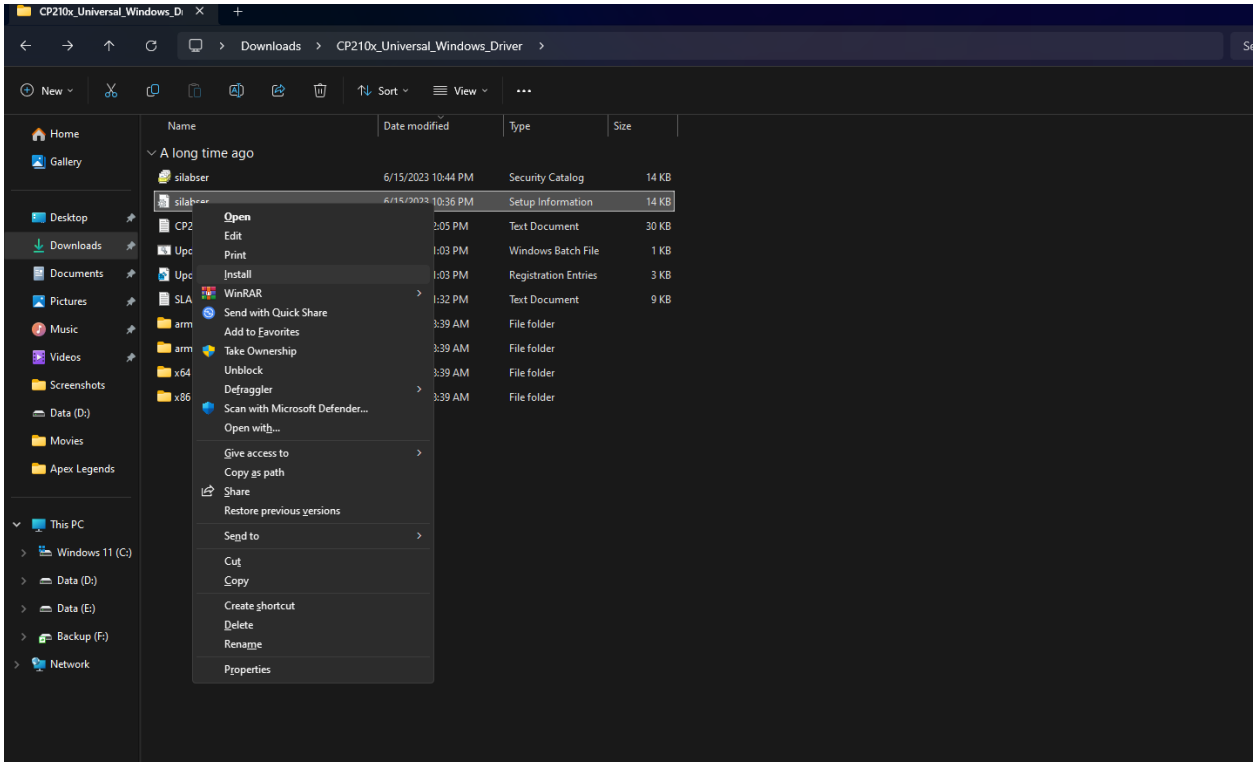


Figure 19. Installing Drivers For microcontroller.

After that download and install CH9102x driver for esp32 device:

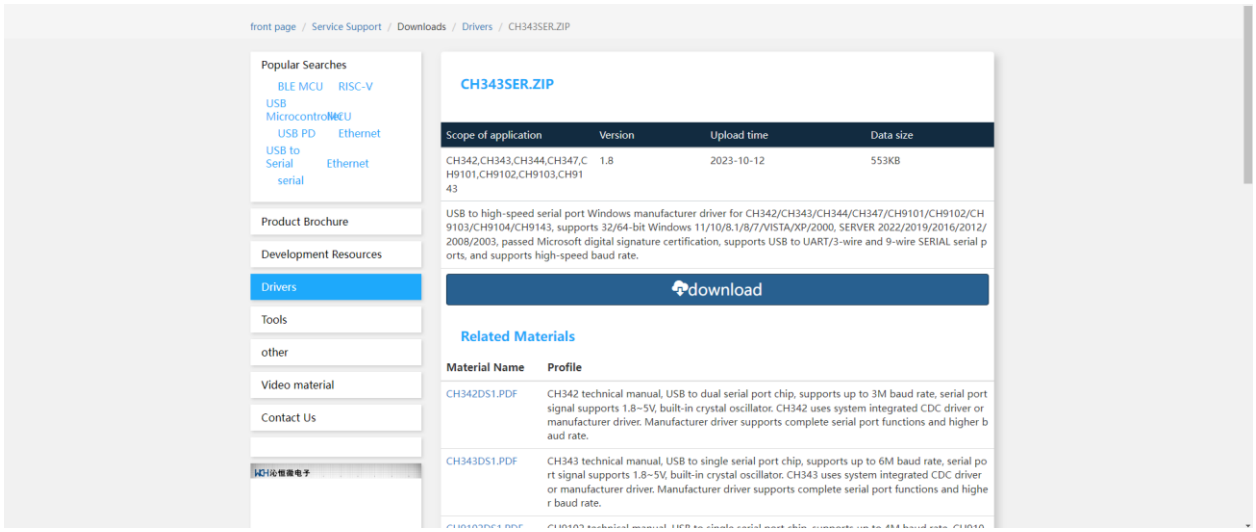


Figure 20. Installing Drivers For microcontroller.

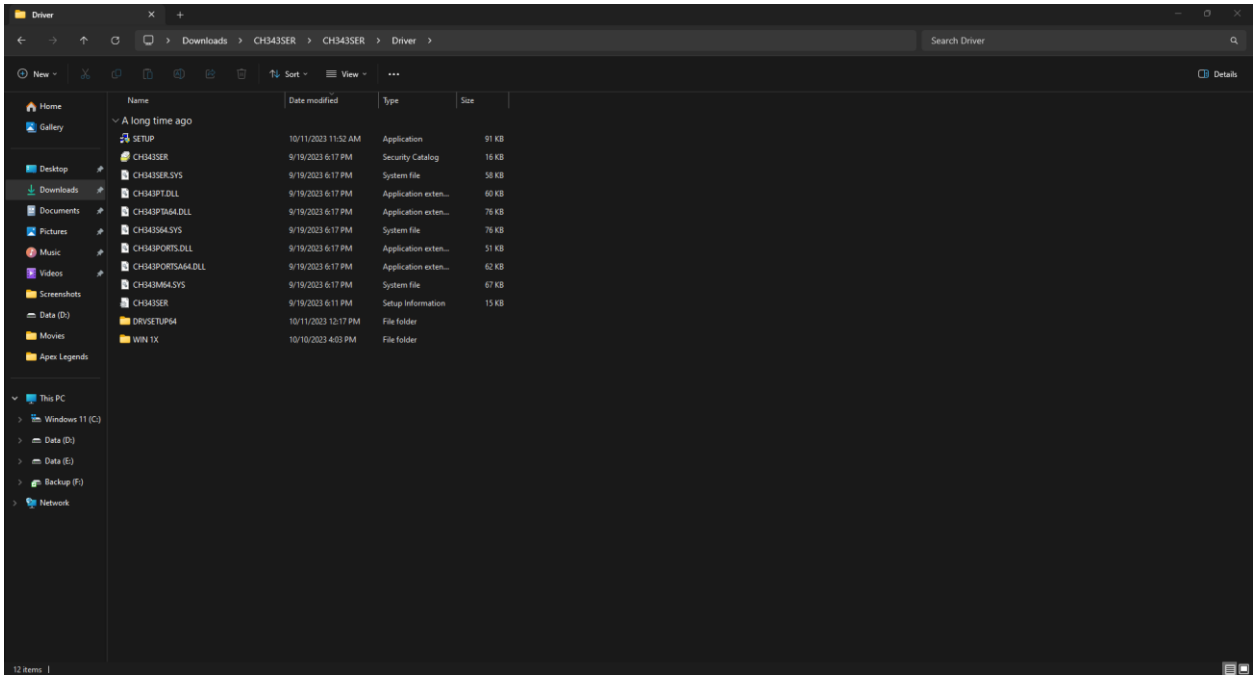


Figure 21. Installing Drivers For microcontroller.

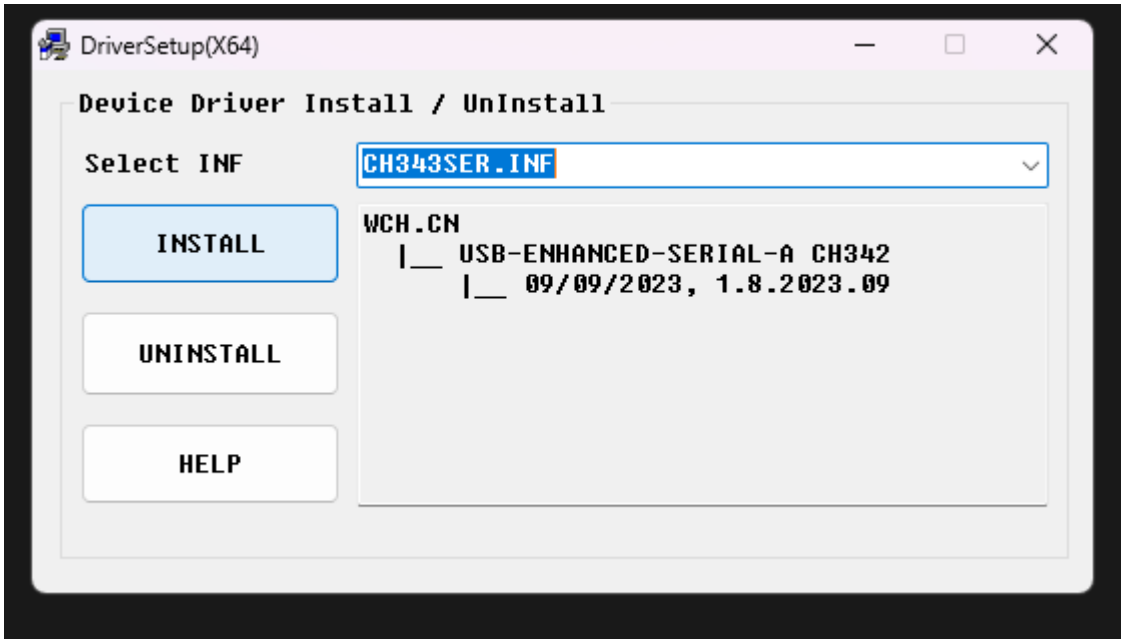


Figure 22. Installing Drivers For microcontroller.

Click install to install all drivers.

Done.

Connect 1st NODEMCU Board to the computer.

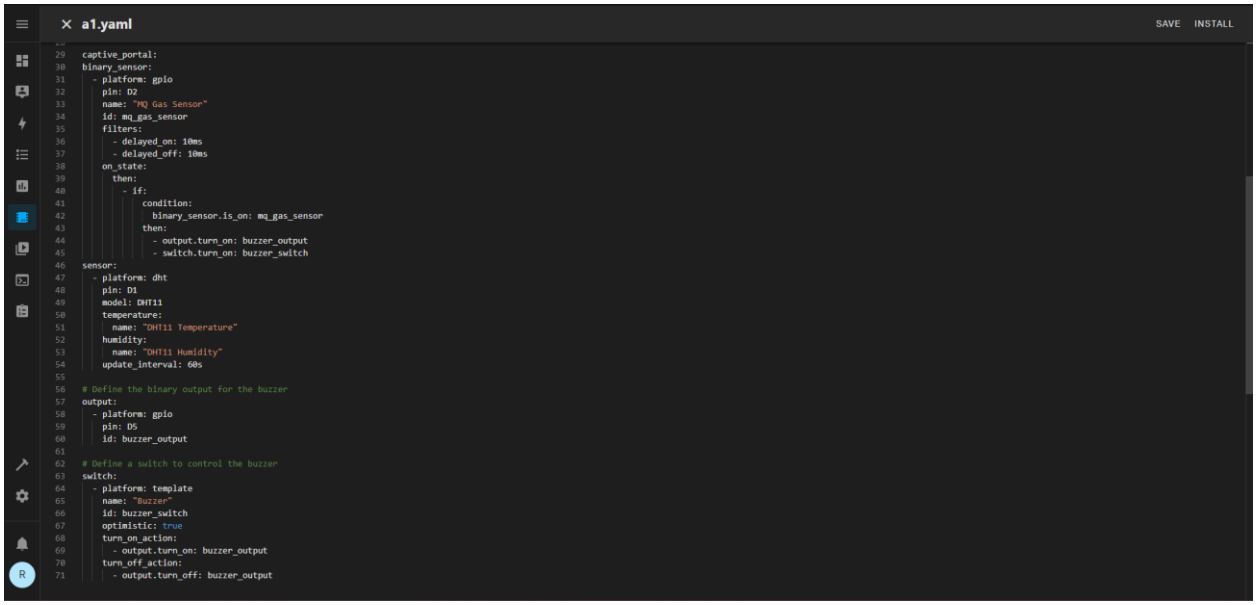


Figure 23. NodeMCU firmware code.

After writing the code click install. Then click download project to download the name.bin file to upload the code.

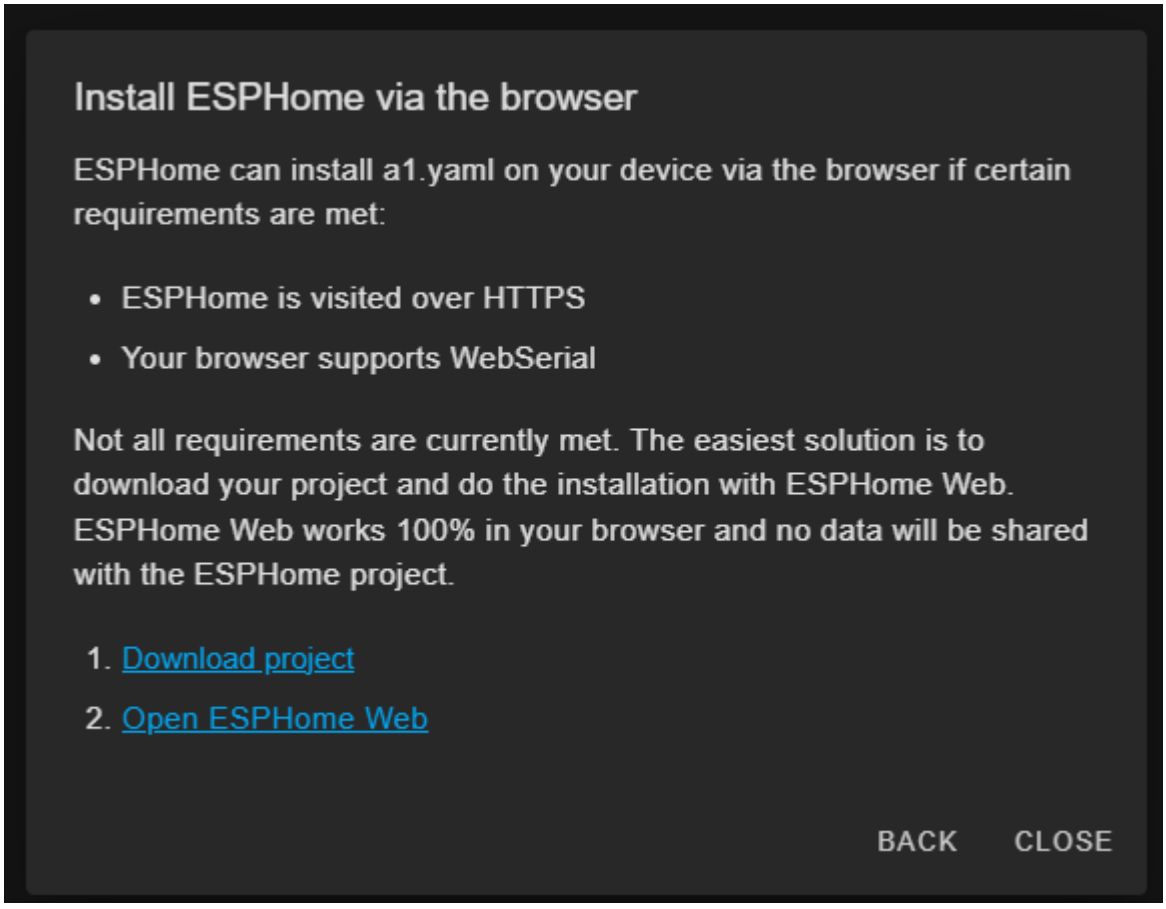


Figure 24. NodeMCU firmware uploading.

Then click open ESPHome web to upload the code.

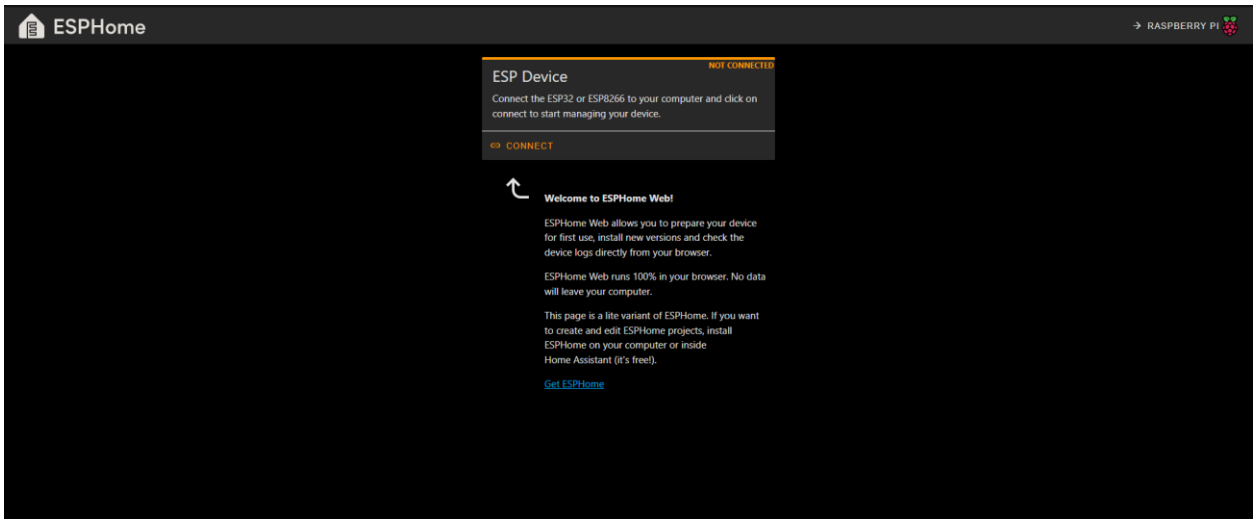


Figure 25. NodeMCU firmware uploading.

Click connect and select your board serial port and click connect.

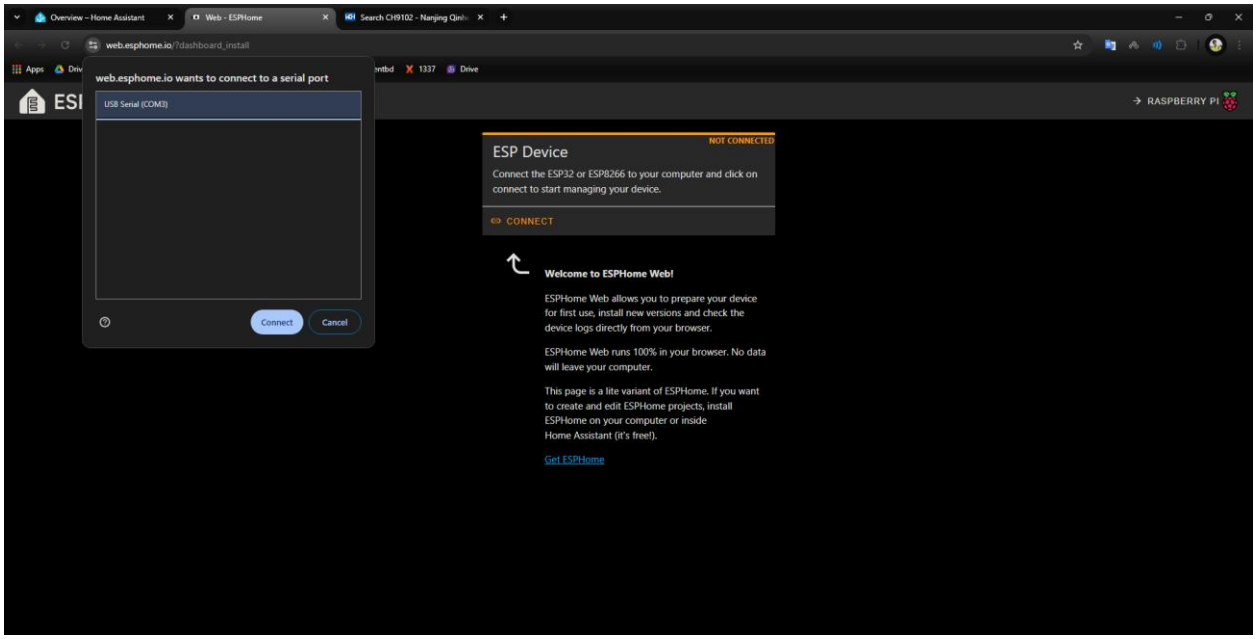


Figure 26. NodeMCU firmware uploading.

Then click install and choose file. Then choose the downloaded name.bin file and hit install. This will upload the code to the nodemcu board.

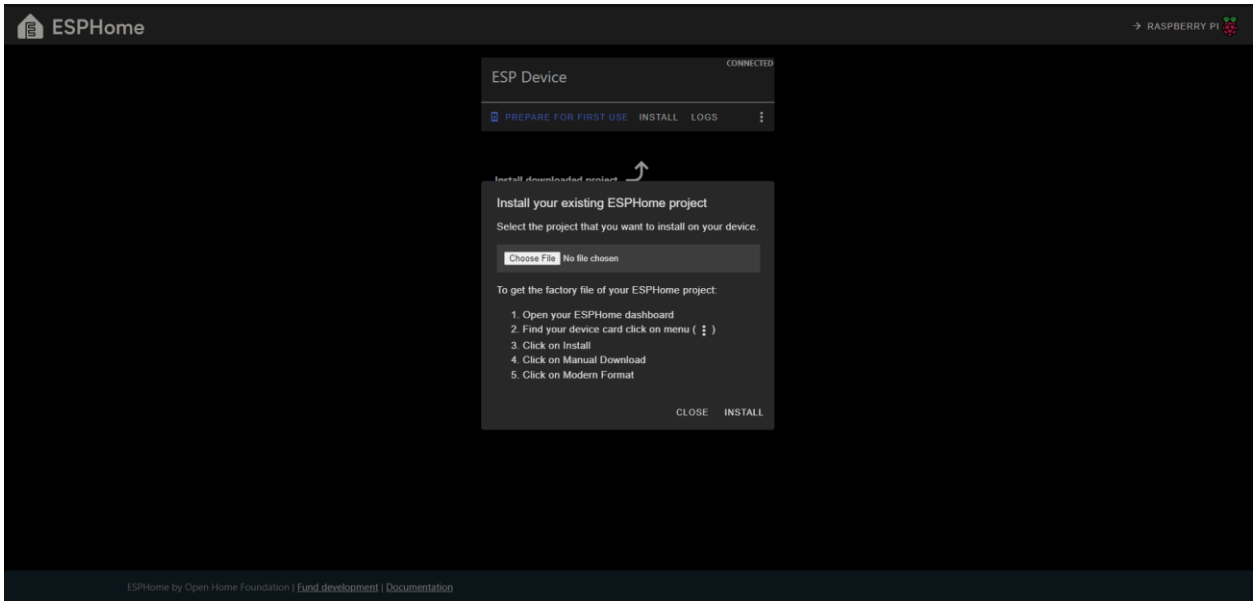


Figure 27. NodeMCU firmware uploading.

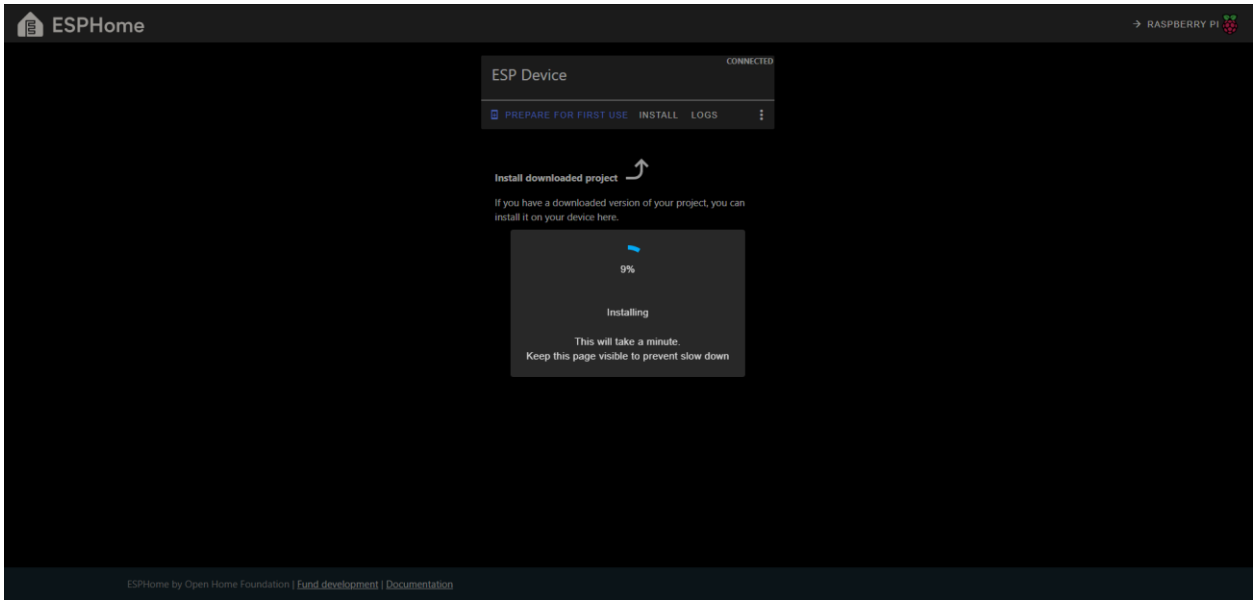


Figure 28. NodeMCU firmware uploading.

That's how I uploaded my code to all my NODEMCU and ESP32 boards.

4.5 Hardware assembly

1: NODEMCU with a Buzzer, DHT11 sensor and a MQ02 Gas sensor.

Connection Layout

Table 2. NodeMCU connection layout with buzzer.

Nodemcu	Buzzer
D5	VCC
GND	GND

Table 3. NodeMCU connection layout with DHT11 temperature and humidity sensor

Nodemcu	DHT11
VCC	VCC
D1	DATA
GND	GND

Table 4. NodeMCU connection layout with MQ02 gas sensor.

Nodemcu	MQ02 Gas sensor
VCC	VCC
D2	D0
GND	GND

Then I soldered all components together with wire.

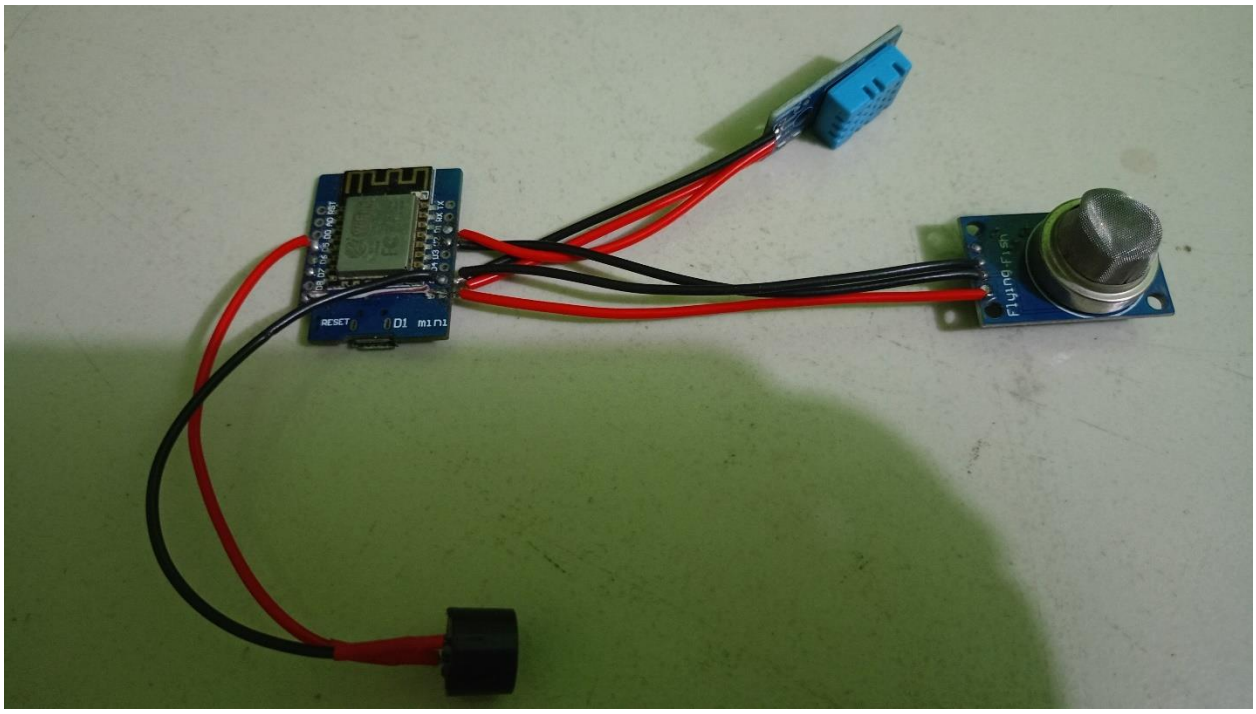


Figure 29. NodeMCU with DHT11, MQ-02 and Buzzer

2: ESP32 with 4 channel relay

Connection layout:

Table 5. ESP32 connection layout with 4 channel relay.

ESP32	4 channel relay
VCC	VCC
GND	GND
D25	IN1
D26	IN2
D27	IN3
D14	IN4

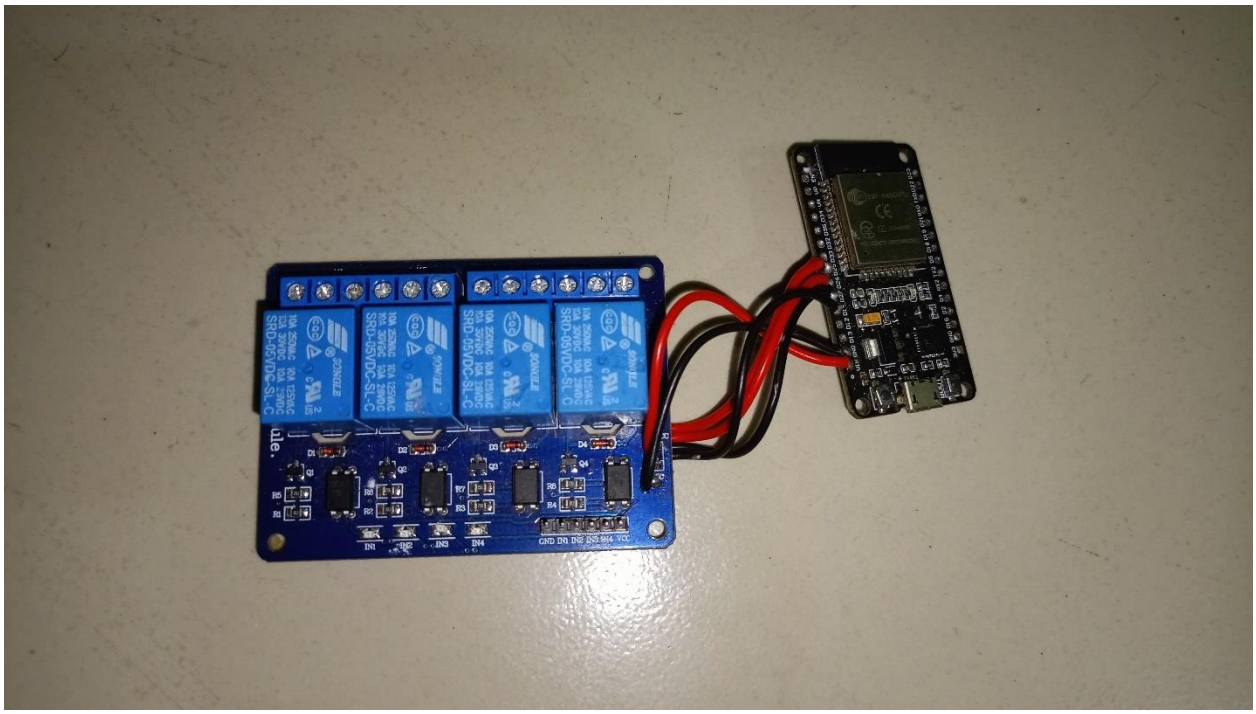


Figure 30. ESP32 with 4 channel relay.

3: Nodemcu with IR led:

Connection layout:

Table 6. NodeMCU connection layout with IR led

Nodemcu	IR led
GND	Negative terminal
D1	Positive terminal

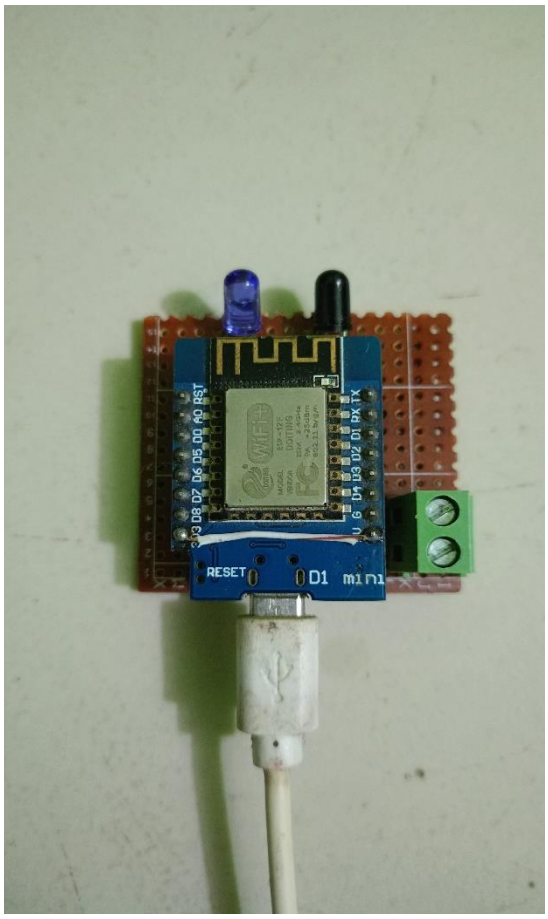


Figure 31. NodeMCU with IR led.

4: Nodemcu with RC522 rfid reader:

Connection layout:

Table 7. NodeMCU connection layout with RC522 RFID reader.

Nodemcu	RC522
3v3	3v3
GND	GND
D3	RST
D4	SDA
D5	SCK
D6	MISO
D7	MOSI

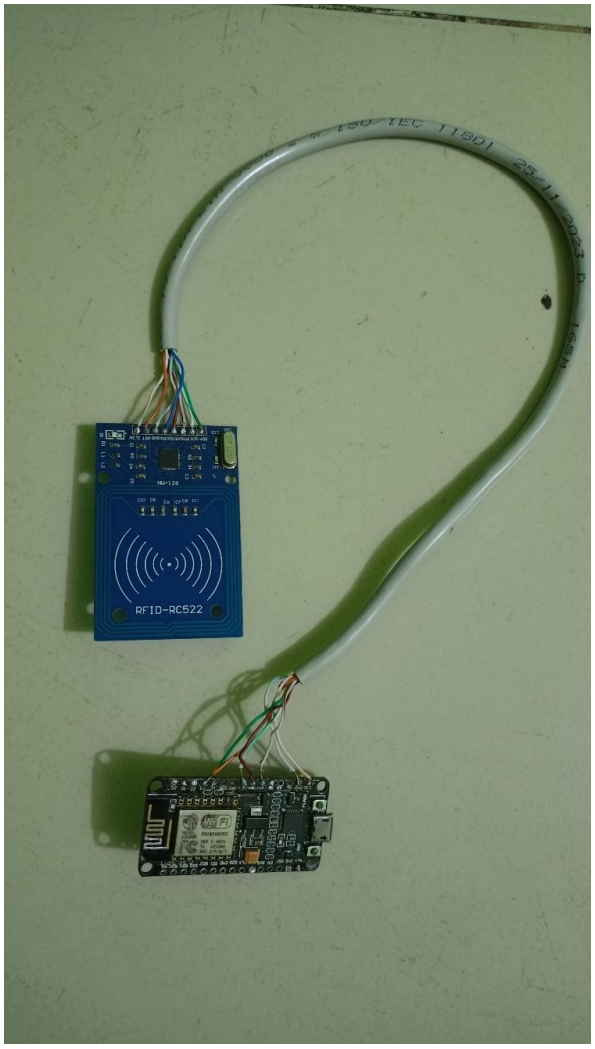


Figure 32. NodeMCU with RC522 module.

Chapter 5 (Logic and Operation)

5.1 Flow chart

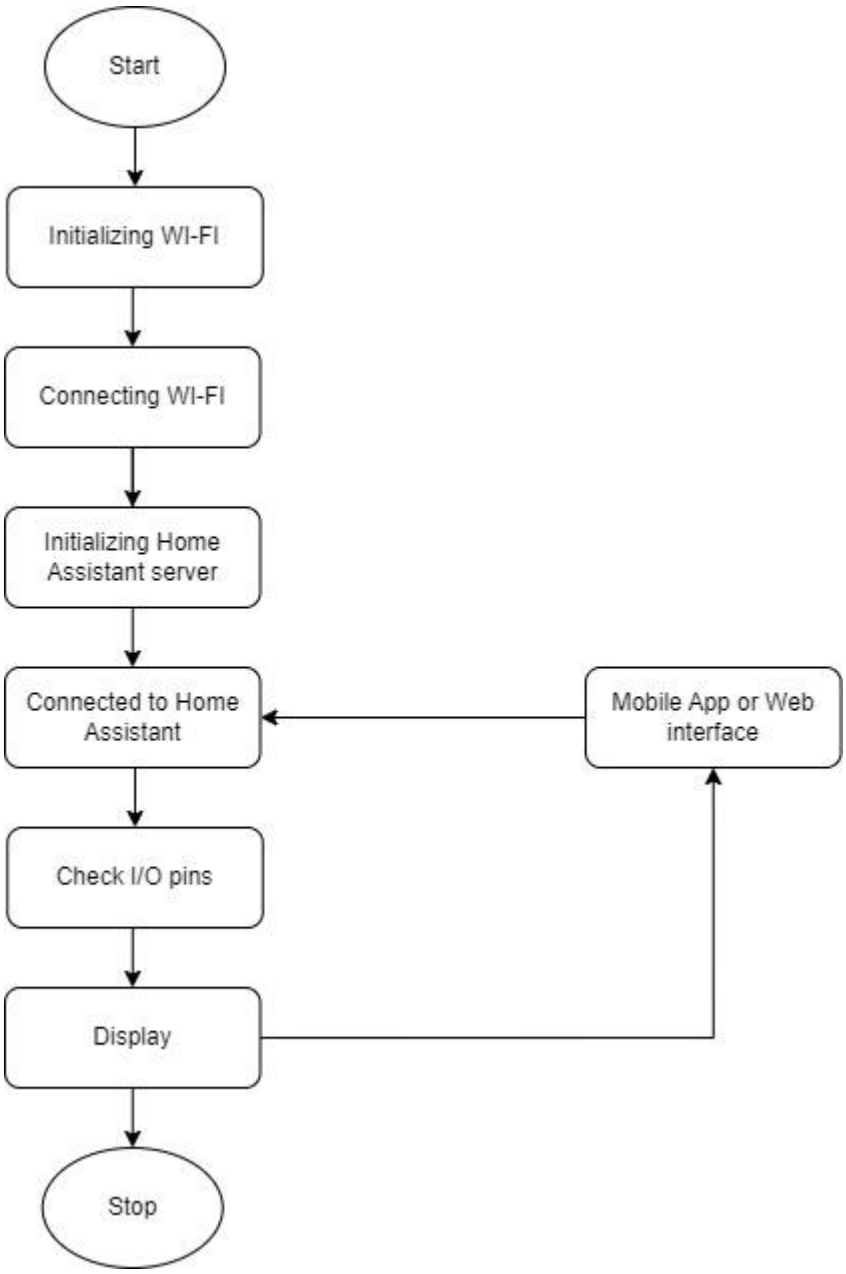


Figure 33. Flow chart of the whole system.

5.2 Principle and operation

Home Assistant Server

Principle

Home Assistant is a centrally located open-source platform for automating smart home technology. It functions based on the idea of integrating a variety of IoT devices into one coherent system, leveraging local control for the purpose of enhancing privacy and reliability. The main concept is to enable users to control and automate their smart home devices through a single interface, ensuring fluid interoperability among different brand and protocol offerings.

Operation

1. Installation:

- Hardware: Home Assistant can be installed on many hardware options such as Raspberry Pi, a dedicated server, or a virtual machine.
- Software: The platform is typically installed via a Linux-based operating system or Docker container.

2. Integration:

- Device Discovery: Home Assistant can automatically discover devices on the home network or allow users to manually add integrations.
- Configuration: Devices are configured using YAML files or the web-based user interface, specifying entity names, states, and attributes.

3. Automation:

- Scripts and Automations: Users can create automation scripts using YAML or through the built-in automation editor to perform specific tasks based on triggers (e.g., time, sensor readings).
- Scenes and Groups: Define scenes (e.g., "Movie Night") and groups (e.g., "All Lights") for unified control of multiple devices.

4. User Interface:

- Lovelace Dashboard: Customizable dashboards allow users to create tailored interfaces to monitor and control their smart home devices.
- Mobile and Web Access: Accessible via web browsers and mobile apps, enabling control from anywhere.

5. Local Control and Privacy:

- Local Processing: Home Assistant emphasizes local control, ensuring that most automation and control processes occur within the local network, reducing dependency on cloud services and enhancing privacy.

NodeMCU (ESP8266)

Principle

The development board NodeMCU makes use of the ESP8266 microcontroller, meant for creating IoT applications. Operation principally relies on offering Wi-Fi connectivity to a diverse range of sensors and actuators for purposes of remote monitoring and management.

Operation

1. Hardware and Pin Configuration:

- GPIO Pins: General-purpose input/output pins for connecting sensors and actuators.
- Power Supply: Operates on 3.3V, with a built-in voltage regulator when powered via USB.
- Communication: Supports UART, SPI, and I2C protocols for interfacing with various devices.

2. Programming:

- MicroPython/Arduino IDE: The NodeMCU can be programmed using MicroPython or the Arduino IDE, making it accessible for both beginners and advanced users.
- Libraries: Various libraries are available for interfacing with sensors, actuators, and Home Assistant.

3. Wi-Fi Connectivity:

- Network Setup: Connects to the home Wi-Fi network, allowing it to communicate with Home Assistant and other devices.
- MQTT Protocol: Often used with MQTT (Message Queuing Telemetry Transport) to send and receive data from Home Assistant.

4. Sensor and Actuator Integration:

- DHT11 Sensor: For temperature and humidity measurements.
- MQ-02 Gas Sensor: For detecting gas concentrations.
- Buzzer: For audible alerts in case of detected anomalies.

5. Operation Example:

- Data Collection: Reads sensor data (temperature, humidity, gas levels) at regular intervals.
- Communication: Sends data to Home Assistant via MQTT, triggering automations such as turning off a gas stove or sending notifications when gas levels exceed safe thresholds.

ESP32

Principle

Due to dual-core processors and an uplift in performance and features, ESP32 is a better microcontroller than ESP8266 and now also supports Bluetooth. Its principle is to achieve powerful connectivity and processing for devices within the Internet of Things (IoT).

Operation

1. Hardware and Pin Configuration:

- GPIO Pins: Multiple GPIO pins for versatile use in various applications.
- ADC and DAC: Includes analog-to-digital and digital-to-analog converters for analog signal interfacing.
- Wireless Communication: Supports Wi-Fi and Bluetooth connectivity.

2. Programming:

- Development Environments: Can be programmed using the Arduino IDE, PlatformIO, or Espressif's ESP-IDF.
- Libraries and Frameworks: Supports numerous libraries for rapid development and integration with Home Assistant.

3. Wi-Fi and Bluetooth:

- Dual Connectivity: Capable of simultaneous Wi-Fi and Bluetooth operations, useful for complex IoT setups.
- MQTT and HTTP: Communicates with Home Assistant using MQTT, HTTP APIs, or WebSockets.

4. Relay Control:

- 4-Channel Relay: Manages four high-power devices through relays, enabling automation of lights, fans, and other appliances.
- Pin Control: Utilizes GPIO pins to switch relays based on automation commands received from Home Assistant.

5. Operation Example:

- Device Control: Receives commands from Home Assistant to toggle devices connected to the relay board.
- Automations: Implements automation scripts to manage device operations based on time, sensor data, or user actions (e.g., turning off lights when no motion is detected).

Combined Operation in the Prototype

By combining these components, the prototype achieves a sophisticated home automation system. The Home Assistant server acts as the central hub, integrating various devices controlled by NodeMCU and ESP32 boards.

- Sensor Monitoring: NodeMCU collects environmental data and sends it to Home Assistant.
- Device Control: ESP32 controls high-power devices through relays, responding to commands from Home Assistant.
- Automation: Home Assistant processes sensor data and triggers automations to control devices.
- User Interface: Users monitor and control their smart home through customizable Home Assistant dashboards and mobile apps.
- Security: ESP8266 with RC522 RFID reader provides secure access control, logging use and managing entry points.

This integration creates a reliable, scalable, and versatile smart home automation system that enhances convenience, security, and energy efficiency.

5.3 Wireless communication network

Wireless communication systems are the key framework for current smart home automation, which allows devices to communicate and connect without any need for physical wiring. A number of wireless technology protocols are operating in IoT environments, giving rise to benefits for certain use cases.

Wireless Communication Technologies for IoT

1. Wi-Fi:

- Range: Moderate to high (indoor ranges typically up to 50 meters, outdoor ranges up to 100 meters with clear line of sight).
- Data Rate: High, capable of supporting large data transfers and multimedia streaming.
- Power Consumption: Generally higher than other IoT-specific technologies.

2. Bluetooth:

- Range: Short (typically up to 10 meters for Bluetooth Classic, up to 100 meters for Bluetooth Low Energy).
- Data Rate: Moderate, suitable for low to moderate data transfers.
- Power Consumption: Low, especially in Bluetooth Low Energy (BLE) mode.

3. Zigbee:

- Range: Low to moderate (typically up to 10-20 meters, extendable via mesh networking).
- Data Rate: Low, designed for small data packets.
- Power Consumption: Very low, ideal for battery-operated devices.

4. Z-Wave:

- Range: Low to moderate (up to 30 meters, extendable via mesh networking).
- Data Rate: Low, suitable for control commands and status updates.
- Power Consumption: Very low, designed for long battery life.

5. LoRaWAN:

- Range: Very high (up to 15 kilometers in rural settings, several kilometers in urban areas).
- Data Rate: Very low, ideal for infrequent, small data packets over long distances.
- Power Consumption: Very low, optimized for long-term deployments.

6. Matter/Thread:

- Range: Moderate, similar to Zigbee and Wi-Fi.
- Data Rate: Moderate to high, designed for interoperability and secure, low-latency communication.
- Power Consumption: Moderate, balancing between performance and efficiency.

5.3.1 Advantages of Wi-Fi Over Other Wireless Technologies

There are various strong reasons for the wide use of Wi-Fi in smart home automation. The benefits, especially with regard to range, data rate, and simplicity, resonate well with many smart home users as well as developers.

1. High Data Rate:

- Wi-Fi facilitates much greater data rates than many protocols dedicated to the Internet of Things. The possession of this capability is essential for applications that involve vast data transfers, including live streaming video from security cameras or comprehensive data from several sensors.

2. Established Infrastructure:

- •The ease of access to Wi-Fi everywhere in housing facilitates setting up new equipment, meaning there's no demand for individual hubs or gateways. It’s easier and quicker to set up devices since they can connect straight to the network.

3. Wide Range:

- differential from Bluetooth and Zigbee because it offers reliable connectivity for comprehensive domestic settings or large areas. This functionality reduces the demand for supplementary repeaters or mesh network organization.

4. Compatibility and Standardization:

- Operating on existing frequencies prominently used internationally (2.4 GHz and 5 GHz), Wi-Fi allows various devices from various manufacturers to interact. Standardization here supports interoperability along with the reliability that will last over the long term.

5. High Throughput and Scalability:

- Advanced standards—Wi-Fi 5 (802.11ac) and Wi-Fi 6 (802.11ax)—turn Wi-Fi networks into systems that can efficiently provide high throughput and accommodate multiple devices without encountering major performance issues.

6. Secure Communication:

- Included in Wi-Fi technology are powerful security measures, like WPA3 encryption, to encourage secured communication and to resist unauthorized access.

7. Versatility:

- The flexibility of Wi-Fi to deliver multiple data transmission types, such as basic sensor data and detailed multimedia flows, makes it suitable and ideal for smart home frameworks that are fully integrated.

5.3.2 Applications Benefiting from Wi-Fi

1. Smart Home Hubs:

- • Units of centralized control that collect data and execute communications between assorted devices. The required bandwidth and reliability for these hubs to operate efficiently come from Wi-Fi.

2. Security Cameras:

- There is considerable demand for bandwidth in real-time video surveillance of high definition and for data storage. Users are able to keep an eye on their homes away from home, thanks to Wi-Fi, supported by mobile apps.

3. Smart Appliances:

- Users can handle and look after the settings for intelligent refrigerators, stoves, and laundry machines due to both smartphones being available and Wi-Fi connections.

4. Home Entertainment Systems:

- Links in creating a larger smart home ecosystem, Wi-Fi enables the nonstop streaming of movies, music, and other content to smart TVs and speakers.

5.Environmental Control:

- Smart thermostats, along with air purifiers and lighting systems, use Wi-Fi to deliver real-time data and control options, thus guaranteeing superior home comfort and energy savings.

Conclusion

A large number of smart home applications choose Wi-Fi due to its outstanding data rate, broad coverage, integrated infrastructure, and reliable security functionality. In contrast to wireless technologies such as Bluetooth, Zigbee, and Z-Wave, Wi-Fi's adaptability and ubiquity provide a strong and unbroken basis for high-level smart home automation. Using Wi-Fi enables users to create a home setting that is integrated, responsive, and impactfull, increasing convenience, security, and the full quality of life.

5.4 Cost estimation

Table 8. Total cost of the system.

SL No	Component	Quantity	Price
1	Nodemcu	3	900
2	Esp32	1	450
3	4 Channel relay	1	290
4	Mq-02 Gas sensor	1	150
5	5V Buzzer	1	20
6	MFRC52 RFID reader Module	1	220
7	IR led	1	10
		Total	2040 TK

Chapter 6 (Conclusion and Future Scope)

6.1 Result

The implementation of Home Assistant alongside ESP32 and ESP8266 microcontrollers has produced a holistic and effective automated home situation. This section reviews the particular results produced through system implementation, demonstrating the effectiveness and advantages of the integration.

1. System Integration and Unified Control

- Successful Integration: The adoption of several smart devices through Home Assistant was successfully achieved. The ESP32 board was able to regulate the 4-channel relay to manage appliances with high powers, and the ESP8266 boards were responsible for environmental sensing (through the DHT11 sensor and MQ-02 gas sensor) along with secure access control through the RC522 RFID reader.
- Centralized Management: The main dashboard in Home Assistant easily managed all devices, presenting an accessible, controlled interface through web and mobile applications. This merge enabled simple monitoring and regulating of the whole smart home system.

2. Automation and Efficiency

- Automated Routines: Many automation routines were produced and were successfully run. To illustrate, systems for automated lighting utilizing time and motion sensors, modifications for climate control according to temperature and humidity data from the DHT11 sensor and security notifications triggered by data from both the gas sensor and RFID reader.
- Energy Savings: The automated administration of lights, equipment, and environment controls resulted in obvious energy savings. The utilization of devices was only when necessary, markedly decreasing inessential power consumption.

3. Enhanced Security

- Real-Time Alerts: The system delivered monitoring capabilities that were real-time alongside alerts for both security compromises and environmental hazards. The buzzer rang and immediate alerts went out through Home Assistant as a result of the MQ-02 sensor detecting gas leaks right away.
- Secure Access Control: The system has added secure and trackable entry thanks to the inclusion of the RC522 RFID reader, which has increased home security. Access events were overseen and recorded, making possible a thorough security oversight.

4. User Experience and Convenience

- Customizable Dashboards: Users had the opportunity to produce and personalize Lovelace dashboards that addressed their particular needs, which improved usability and accessibility. The

dashboards delivered real-time data visualization, coupled with buttons for control and updates on the status of all incorporated devices.

- Voice Control: Amazon Alexa and Google Assistant are voice assistants that now allow users to simply manage their smart home systems. Users were able to do jobs such as switching on lights or changing the thermostat with just simple voice instructions.

5. Environmental Monitoring

- Real-Time Data Collection: Home Assistant received data about the environment (temperature, humidity, gas levels) in a reliable manner. The data was instrumental in automating climate control systems that kept comfort levels at their best within the house.
- Sustainable Living: The utility of the system to monitor and govern energy use was key to the development of more sustainable lifestyles. Management of resources was facilitated through real-time energy usage data, which mirrors green living objectives.

6. Scalability and Flexibility

- Scalable Setup: The design of the system as modules made expansion quite easy. Further sensors and devices might be incorporated into the current framework without significant modifications.
- Future Integration: The project's successful implementation proved the possible integration of novel IoT technologies and devices for assuring future viability and flexibility.

6.2 Limitations

Despite its many successes, the smart home automation project also faced several limitations:

Network Dependency:

The system's reliable operation relies mainly on a consistent Wi-Fi connection. A change or disruption in the network can produce either delays or failures in the communication between devices and their automation workflows.

Power Consumption of Wi-Fi Devices:

Wi-Fi quite often supplies outstanding connectivity, but it usually has a larger appetite for power compared to other low-power Internet of Things protocols such as Zigbee or Z-Wave, possibly rendering them less efficient for devices that depend on batteries.

Interoperability Challenges:

The issues derive from the many protocols and standards needed in the process of uniting devices manufactured by different firms. Accomplishing seamless interoperability needed considerable configuration effort.

Security Concerns:

Preserving the system from cyber-attacks is incredibly important. Local control is the focus of Home Assistant. However, the need for extensive security measures arises from any exposure to external networks, increasing the complexity of setup.

User Technical Expertise:

The setup and customization of Home Assistant at the beginning can be intimidating for individuals with restricted technical proficiency. Acquiring knowledge about YAML configurations and integrations might include tackling a serious learning challenge.

Scalability Issues:

As the number of connected devices rises, the role of managing and sustaining the system may become more complicated. Network congestion and problems with devices are two likely scalability issues that could develop.

6.3 Conclusion

Thanks to its smart home automation project, Home Assistant and ESP32/ESP8266 microcontrollers successfully demonstrated their integration and management of a range of IoT devices. It made clear important developments concerning residential convenience, safety, and energy efficiency thanks to controlled routines and centralized control. This project brought attention to the strong abilities of Home Assistant as a platform capable of managing diverse devices, delivering users a customizable interface that is smooth to use. Among the obstacles encountered by the initiative were network dependency, energy consumption, inadequate interoperability, and the important requirement for robust security methods. In any case, the project demonstrated that smart home automation is feasible and provides rewards, illustrating its potential to modify ordinary dwellings into intelligent and responsive living spaces. Resolving these challenges and limitations will improve future deployments' user experience as well as the reliability of smart home systems. This work stresses the revolutionary improvement IoT provides to make homes smarter, more effective, and safer.

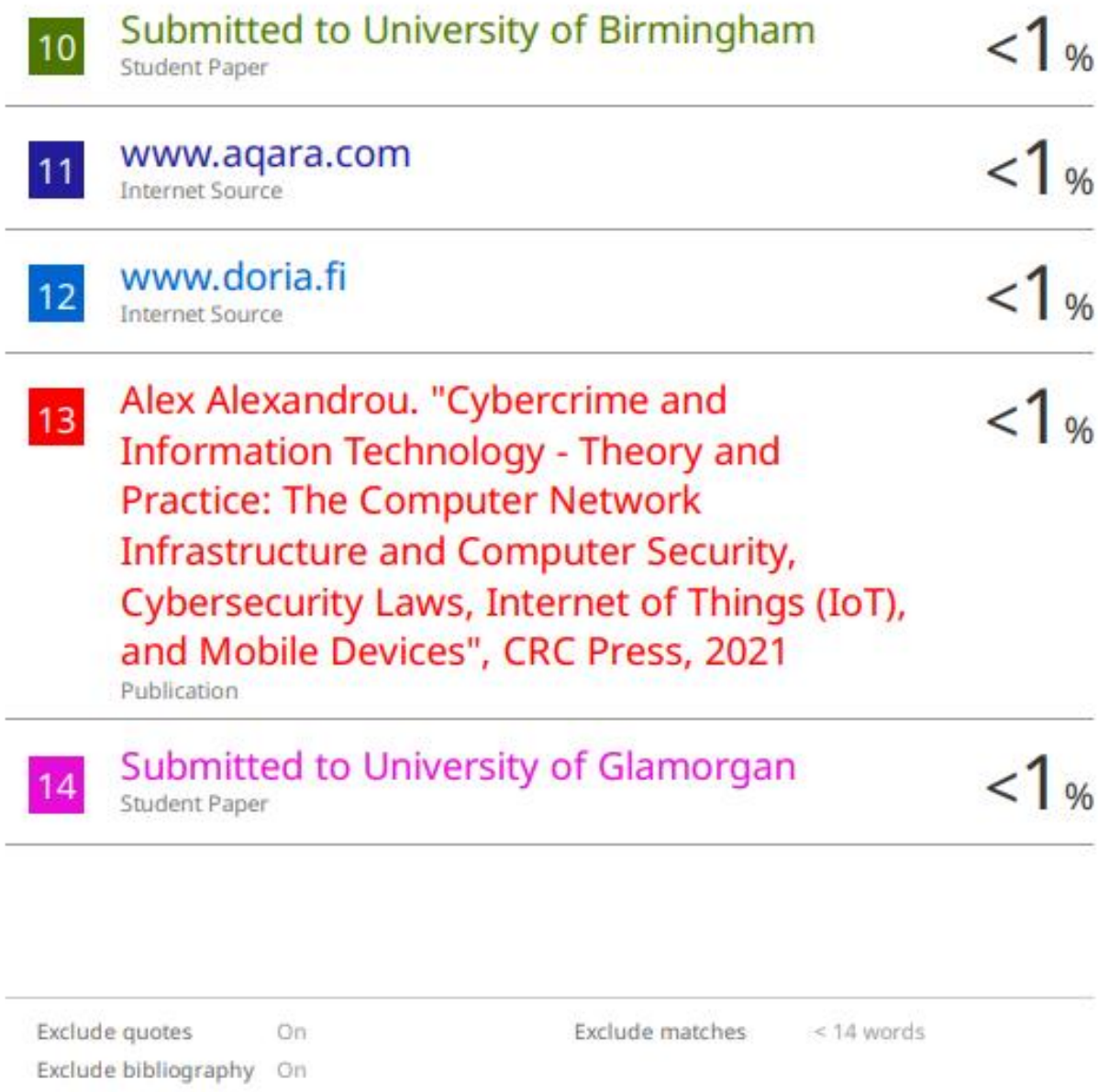
Rafat

ORIGINALITY REPORT

6%	6%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
2	rcciit.org Internet Source	2%
3	www.coursehero.com Internet Source	<1%
4	mediamingle.online Internet Source	<1%
5	Submitted to Nizwa College of Technology Student Paper	<1%
6	fastercapital.com Internet Source	<1%
7	lore.kernel.org Internet Source	<1%
8	utilitiesone.com Internet Source	<1%
9	Submitted to College of Estate Management Student Paper	<1%



10	Submitted to University of Birmingham Student Paper	<1 %
11	www.aqara.com Internet Source	<1 %
12	www.doria.fi Internet Source	<1 %
13	Alex Alexandrou. "Cybercrime and Information Technology - Theory and Practice: The Computer Network Infrastructure and Computer Security, Cybersecurity Laws, Internet of Things (IoT), and Mobile Devices", CRC Press, 2021 Publication	<1 %
14	Submitted to University of Glamorgan Student Paper	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		