

SMART HOME



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Submitted by

Devang Dubey (CS21MTECH14013)

Nilesh Kale (CS21MTECH11022)

Satvik Padhiyar (CS21MTECH14011)

May 7, 2022



Table of Contents

ABSTRACT	3
INTRODUCTION	4
CORE COMPONENTS	5
HARDWARE	5
Node MCU	5
5V 4-channel Relay Module	6
SOFTWARE	7
AWS IoT	7
MQTT	8
Arduino IDE	8
PROJECT MANAGEMENT	9
PROJECT DESIGN	10
BLOCK DIAGRAM	12
CIRCUIT DIAGRAM	13
FLOW CHART	14
IMPLEMENTATION	15
CONFIGURATION AND SETUP	16
VIRTUALIZATION	19
HARDWARE IMPLEMENTATION	23
TECHNICAL SPECIFICATION	27
CONCLUSION AND FUTURE SCOPE	29
REFERENCES	31



ABSTRACT

The overall design of a Smart Home with a low-cost and wireless technology is presented in this project. It focuses on the creation of an IoT-based home automation system that can control multiple components over the internet or be automatically programmed to work based on environmental circumstances. In this project, we will create a firmware for smart control that can be successfully automated while minimizing human contact in order to maintain the integrity of entire electrical gadgets in the home. To automate the process, we used Node MCU, a popular open source IOT platform. Different transmission modes will be used by different components of the system to send the user's control of the devices to the actual appliance via Node MCU. Wireless technology is used in the main control system to provide remote access from a smartphone. We're leveraging cloud server-based connectivity, which will make the idea more realistic by allowing users free access to the appliances. To improve automation, we provided a data transmission network. The system was created with a low-cost design, a user-friendly interface, and ease of installation in mind to control electrical appliances and devices in the home. The status of the appliance, as well as control over it via an android platform, would be available. This system is meant to assist and support elderly and disabled people at home. Also, the smart home concept in the system improves the standard of living at home.



CHAPTER 1

INTRODUCTION



The concept of “Home Automation” has been in existence for several years. “Smart Home”, “Intelligent Home” are terms that followed and have been used to introduce the concept of networking appliances within the house. Smart Home includes centralized control and distance status monitoring of lighting, security system, and other appliances and systems within a house. It enables energy efficiency, improves the security systems, and certainly the comfort and ease of users. In the present emerging market, Smart Home is gaining popularity and has attracted the interests of many users. It comes with its own challenges. Mainly being, in the present day, end users especially elderly and disabled, even though hugely benefited, aren't seen to accept the system due to the complexity and cost factors.

CORE COMPONENTS

HARDWARE

Node MCU

NodeMCU (Node Microcontroller Unit) is a low-cost open source [IOT](#) platform. It initially included [firmware](#) which runs on the [ESP8266 Wi-Fi SoC](#) from Espressif Systems, and hardware which was based on the ESP-12 module.

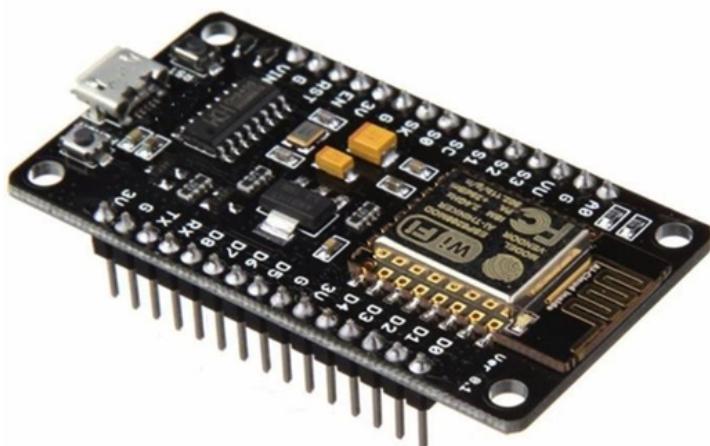




Figure: Node MCU Development Board

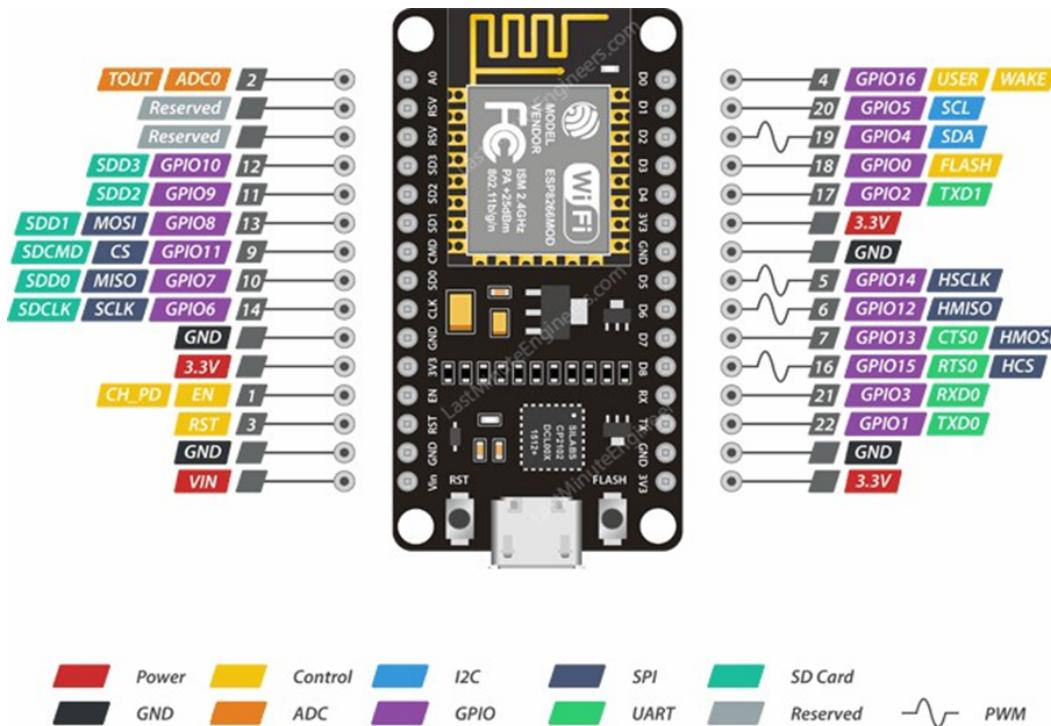


Figure: Node MCU Pin Diagram

5V 4-channel Relay Module



Figure: 5V 4-channel relay interface board



In a 5V 4-channel relay interface board, each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by a microcontroller.

SOFTWARE

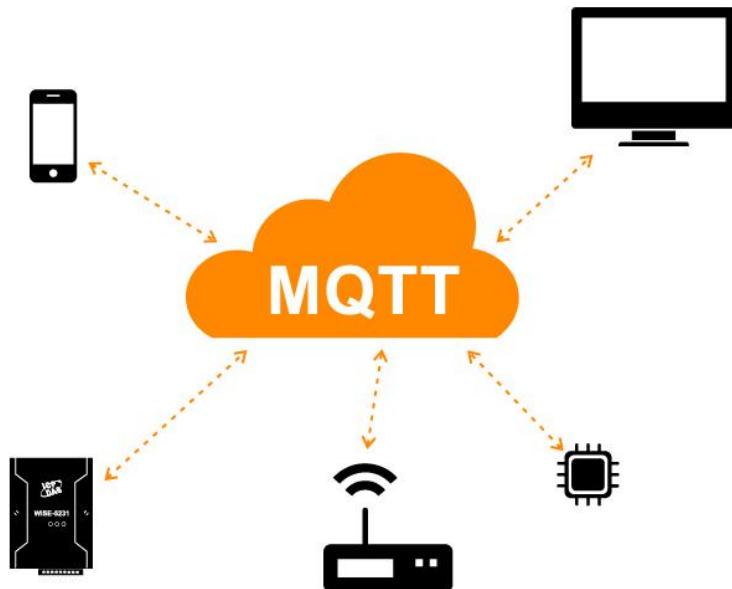
AWS IoT



AWS IoT provides cloud services and device support that you can use to implement IoT solutions. AWS provides many cloud services to support IoT-based applications. So to help you understand where to start, this section provides a diagram and definition of essential concepts to introduce you to the IoT universe.



MQTT



MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

Arduino IDE

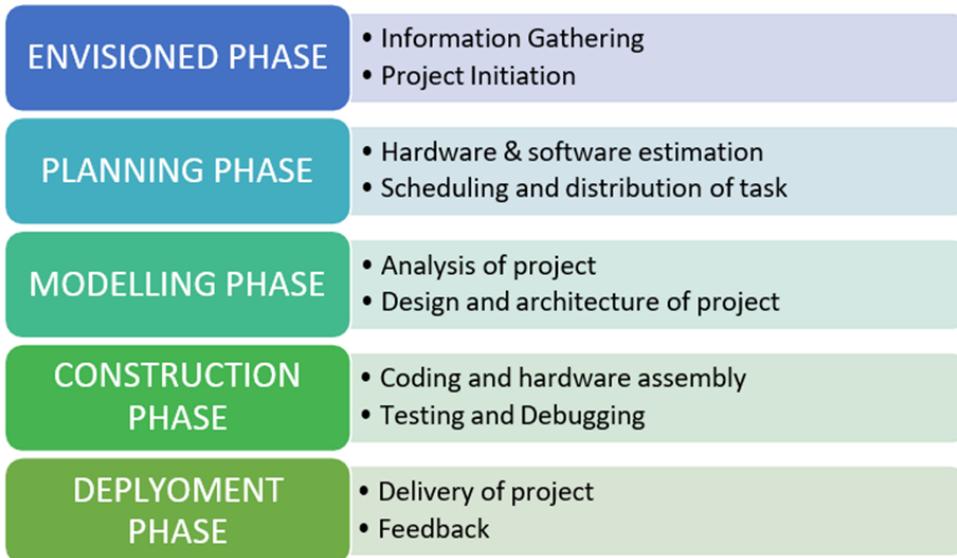


The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++.



PROJECT MANAGEMENT

Management of any project can be briefly disintegrated into several phases. Our project has been decomposed into the following phases:



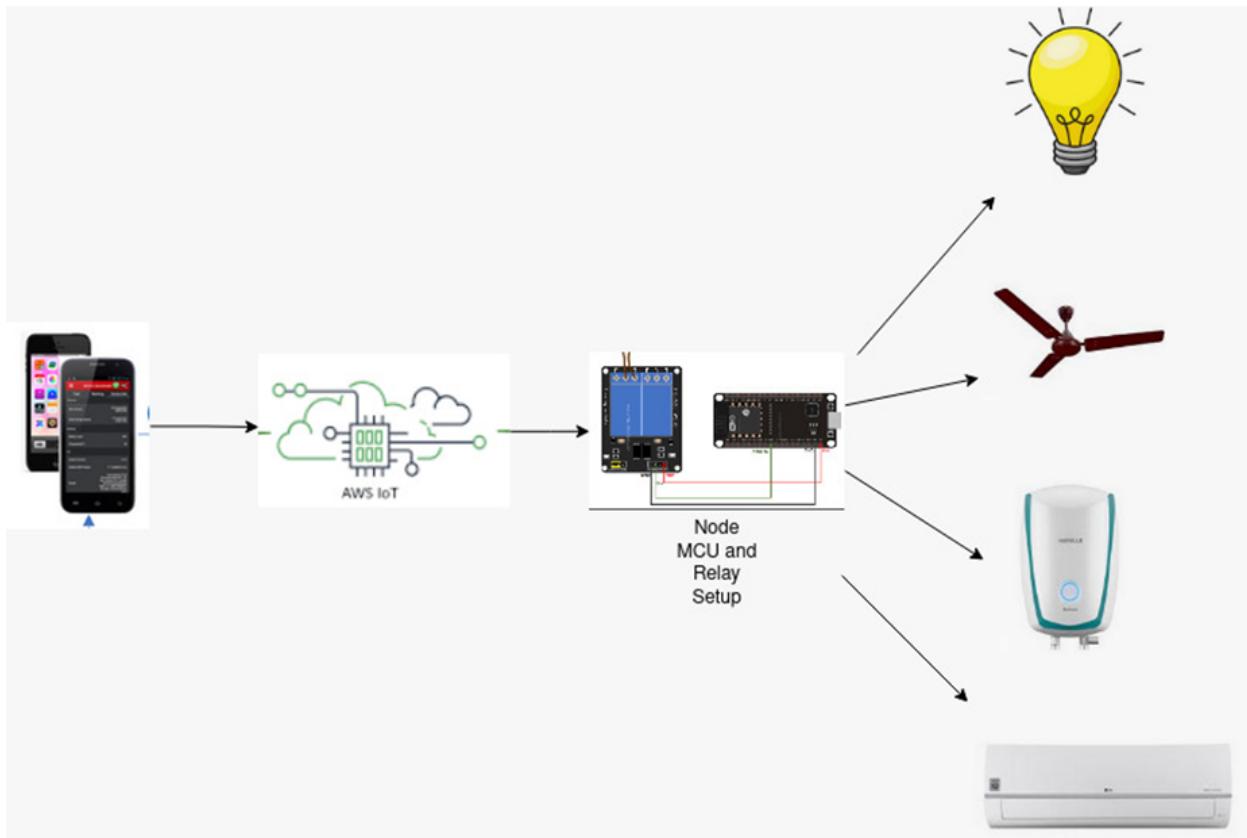


CHAPTER 2

PROJECT DESIGN



DESIGN



The Node MCU unit is the microcontroller or the main controlling unit of the system. The user uses the mobile application in setting commands for functioning of the appliances. The mobile application interprets the command form in user in voice or switch mode and sends a signal to the Node MCU unit, over a wireless network established by Wi-Fi communication. Hence the Wi-Fi module (actually inbuilt into Node MCU), helps the microcontroller establish Wi-Fi communication with a device and take commands from an application over wireless network. The Node MCU on further receiving the signal then turns on/off the appliance with the help of relay. The Node MCU, relay and the final appliances are physically connected. There is a power supply unit that powers the microcontroller, the relay as well as the final appliances. There is also a display unit that displays the status of the application.



BLOCK DIAGRAM

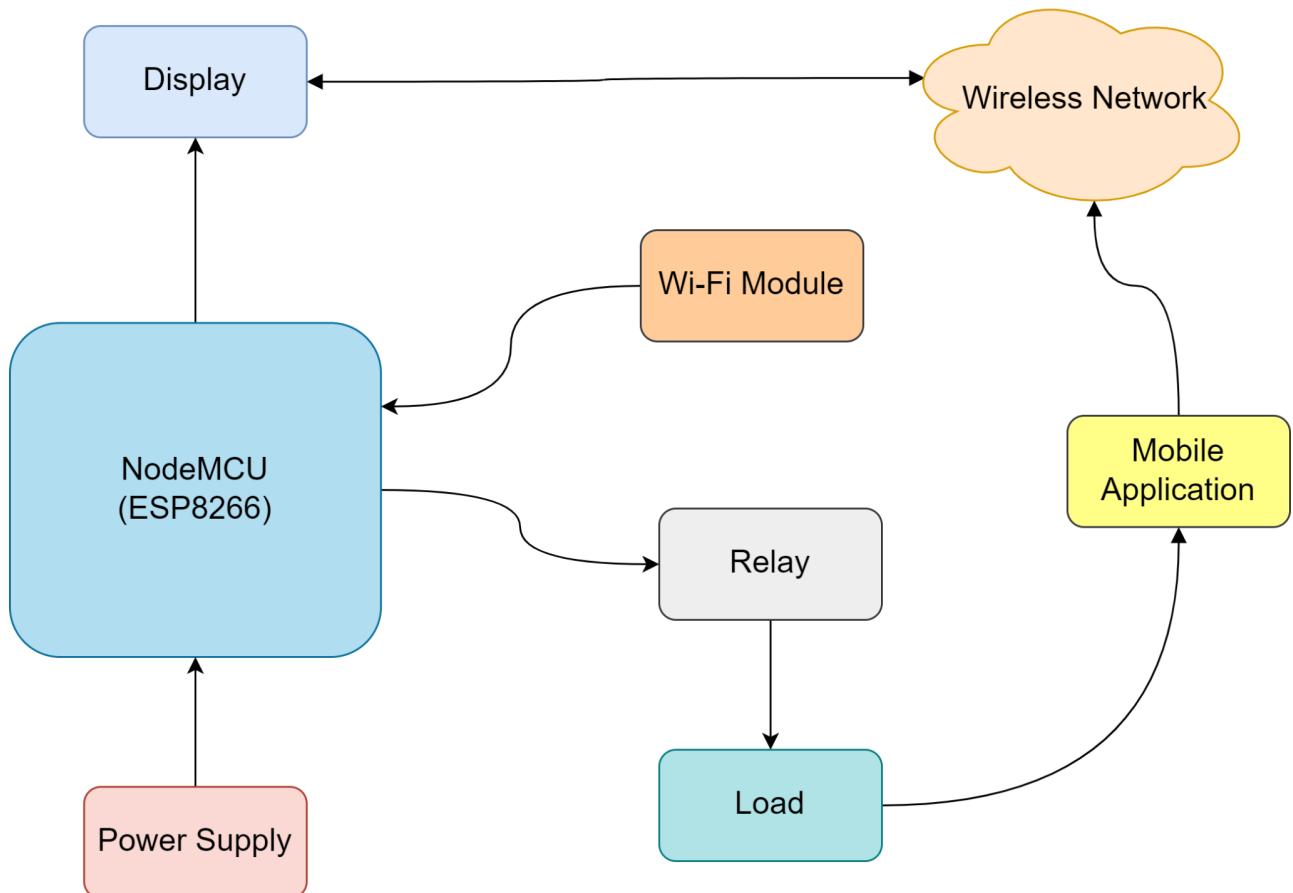


Figure. Block diagram of proposed system.



CIRCUIT DIAGRAM

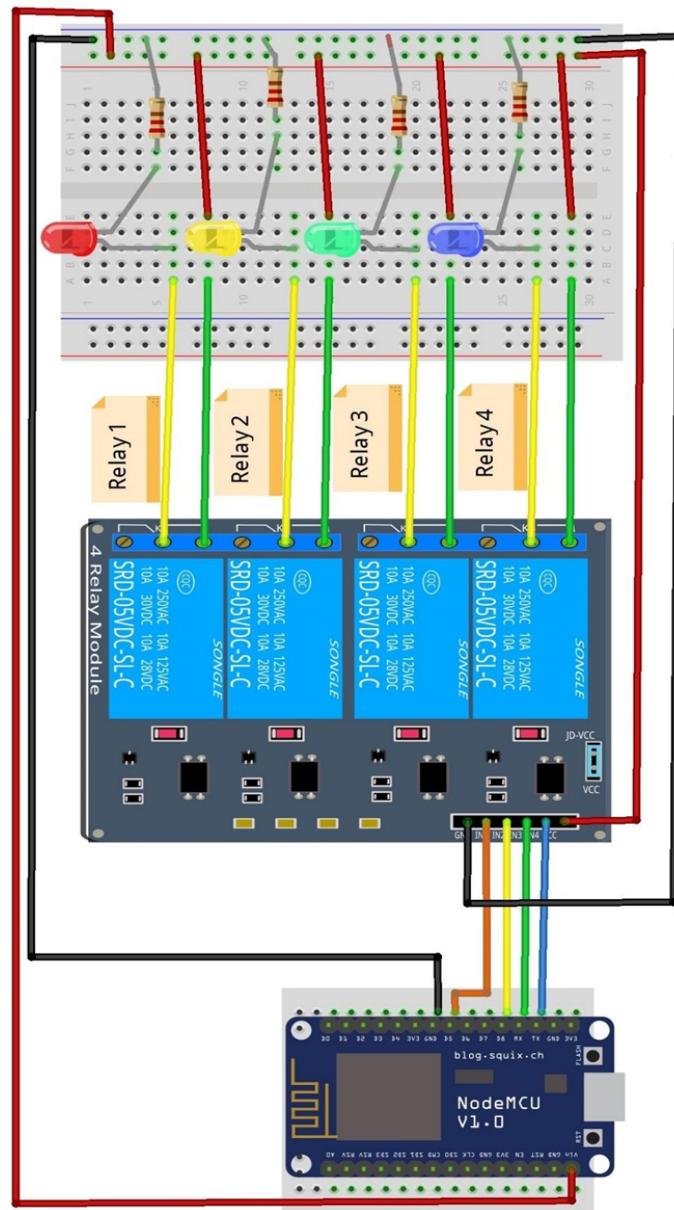


Figure. Connection diagram of Node MCU controlling 4 channel relay module.

FLOW CHART

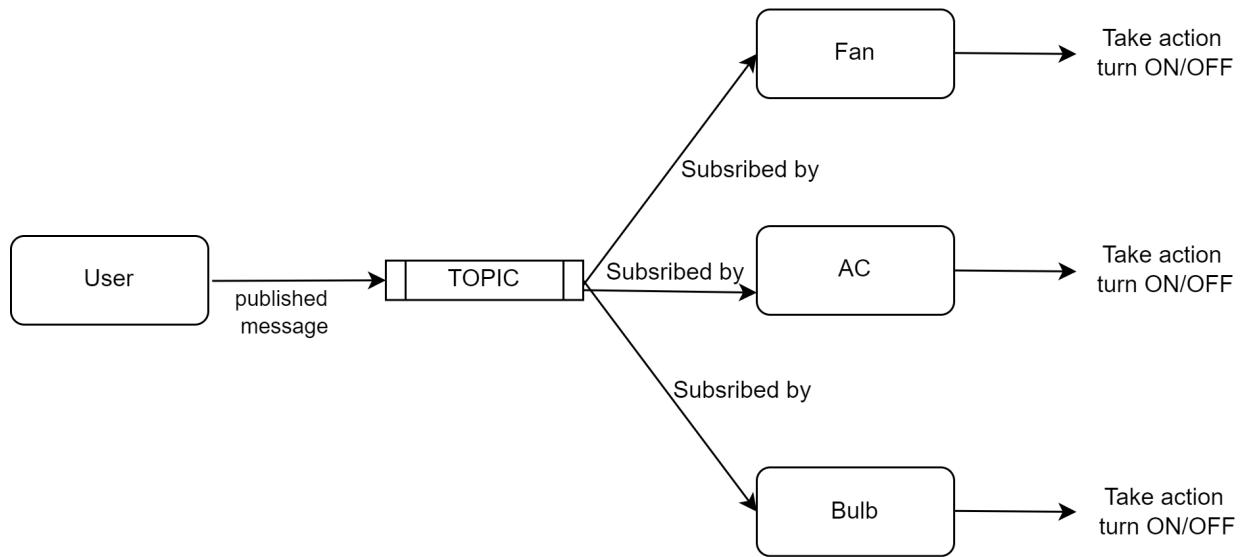


Figure: Data flow diagram of Smart Home Virtualization

This flowchart shows the working of the project. User publishes a message on a topic (Appliances). All the devices that are subscribed to that topic will receive the message. Here all devices are subscribed to the topic. On receiving the message, the devices perform action like turning on and off the device.



CHAPTER 3

IMPLEMENTATION



CONFIGURATION AND SETUP

Before all of the devices can communicate with each other over AWS MQTT on the AWS IoT core, the AWS IoT core must be established and a unique digital certificate generated for each device.

[Alt+S]

Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
bf19e3baec6...te.pem.crt

[Deactivate certificate](#) [Download](#)

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file
bf19e3baec665b731edf78f...aa2784d-public.pem.key

[Download](#)

Private key file
bf19e3baec665b731edf78f...a2784d-private.pem.key

[Download](#)

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

[Download](#)

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

[Download](#)

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

[Done](#)

Figure : Digital Certificate



Each device is first added to the "Thing" area of the AWS IoT core service's "Manage" section.

Each thing has a certificate allotted automatically by the AWS which is required to be downloaded locally and used while connecting to the MQTT broker. And each thing is assigned a policy.

The screenshot shows the AWS IoT Policy Management interface. At the top, there is a title bar with the policy name 'iot_project_policy' and 'Info' link, along with 'Edit active version' and 'Delete' buttons. Below the title bar is a 'Details' section containing policy metadata: Policy ARN (arn:aws:iot:ap-south-1:453402102790:policy/iot_project_policy), Active version (1), Created (April 25, 2022, 15:32:14 (UTC+0530)), and Last updated (April 25, 2022, 15:32:14 (UTC+0530)). Below the details are tabs for 'Versions', 'Targets', 'Noncompliance', and 'Tags', with 'Versions' being the active tab. Under the 'Active version' section, there are three columns: 'Policy effect' (Allow), 'Policy action' (*), and 'Policy resource' (*). There are also 'Builder' and 'JSON' buttons for this section.

Figure: Policy assigned for each thing

MQTT endpoint that all devices connect and communicate through is as follows

The screenshot shows the AWS IoT Device Data Endpoint configuration page. It features a title 'Device data endpoint' with an 'Info' link and a 'Copy' icon. A note below the title states: 'Your devices can use your account's device data endpoint to connect to AWS.' Below this is a descriptive text: 'Each of your things has a REST API available at this endpoint. MQTT clients and AWS IoT Device SDKs also use this endpoint.' Under the 'Endpoint' section, there is a text input field containing the URL 'a30xom1njimvh2-ats.iot.ap-south-1.amazonaws.com'. A 'Copy' icon is located to the right of the URL.

Figure: AWS MQTT Endpoint



Things (5) <small>Info</small>	
An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.	
Create things	
	Advanced search
	Run aggregations
	Edit
	Delete
<input type="text"/> Filter things by: name, type, group, billing, or searchable attribute.	
<input type="checkbox"/> Name	Thing type
<input type="checkbox"/> user	-
<input type="checkbox"/> controller	-
<input type="checkbox"/> air_conditioner	-
<input type="checkbox"/> bulb	-
<input type="checkbox"/> fan	-

Figure: All things configured



VIRTUALIZATION

Device interfaces when launched

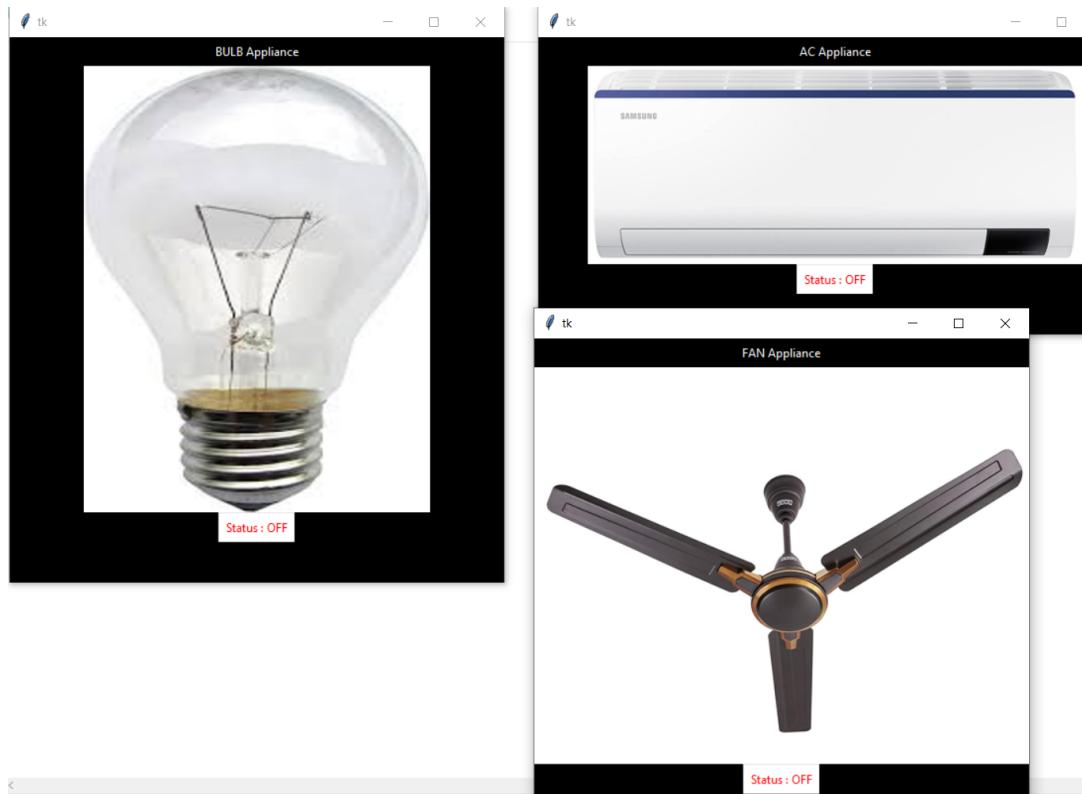


Figure: Interface of Appliances Modules

Running the Interface:

Turning the Fan-On

Windows PowerShell

```
Connecting to a30xom1njimvh2-ats.iot.ap-south-1.amazonaws.com
Connected!
Request with syntax DEVICE-ACTION format only
Request : fan-on
```



Output:

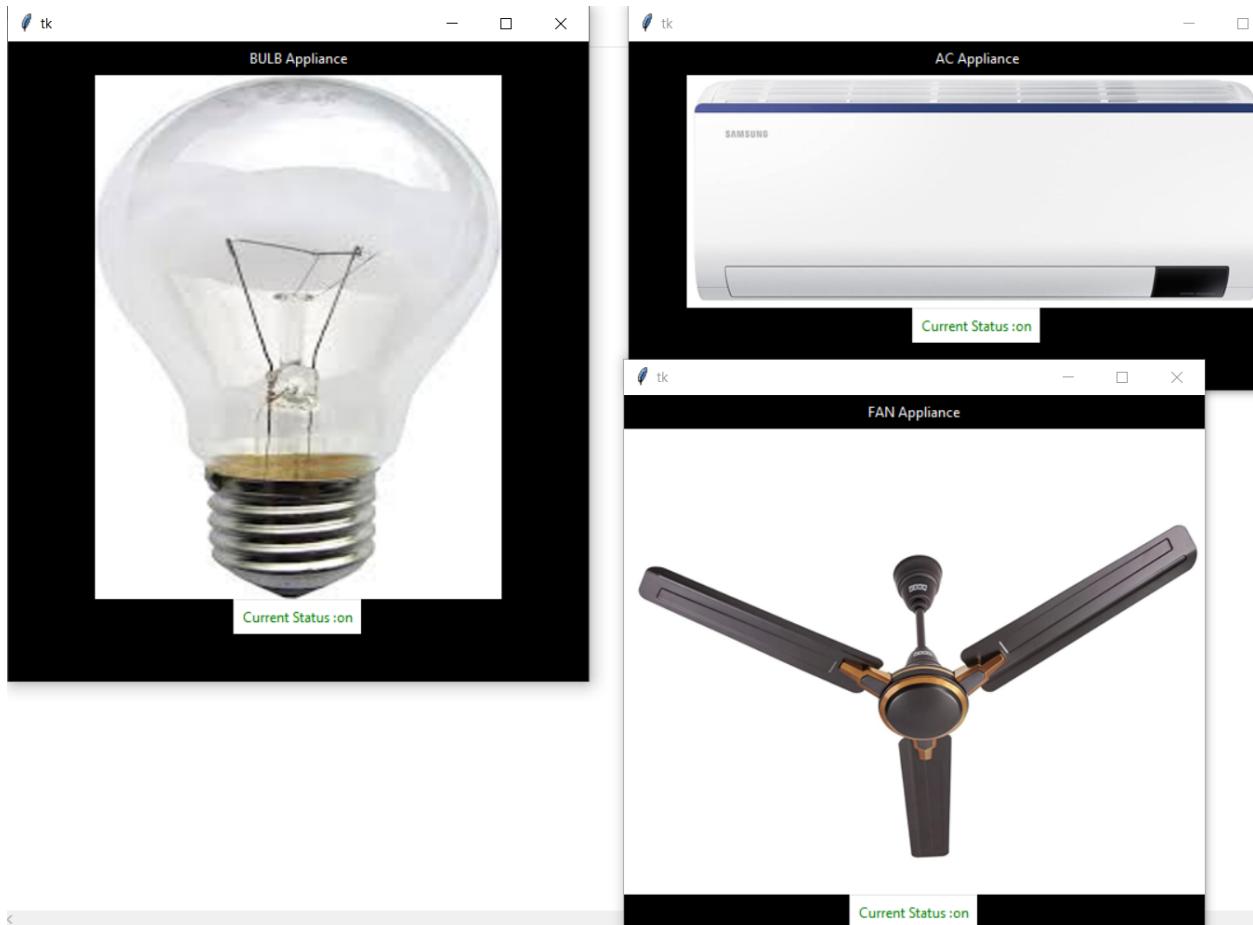


Now, Turning all the devices on:

```
Request with syntax DEVICE-ACTION format only
Request : fan-on
Request : bulb-on
Request : ac-on
Request :
```



Output:



Turning bulb off:

```
Request with syntax DEVICE-ACTION format only
Request : fan-on
Request : bulb-on
Request : ac-on
Request : bulb-off
```



Output:



Message Logs saved on the systems are as follows:

logs.txt - Notepad

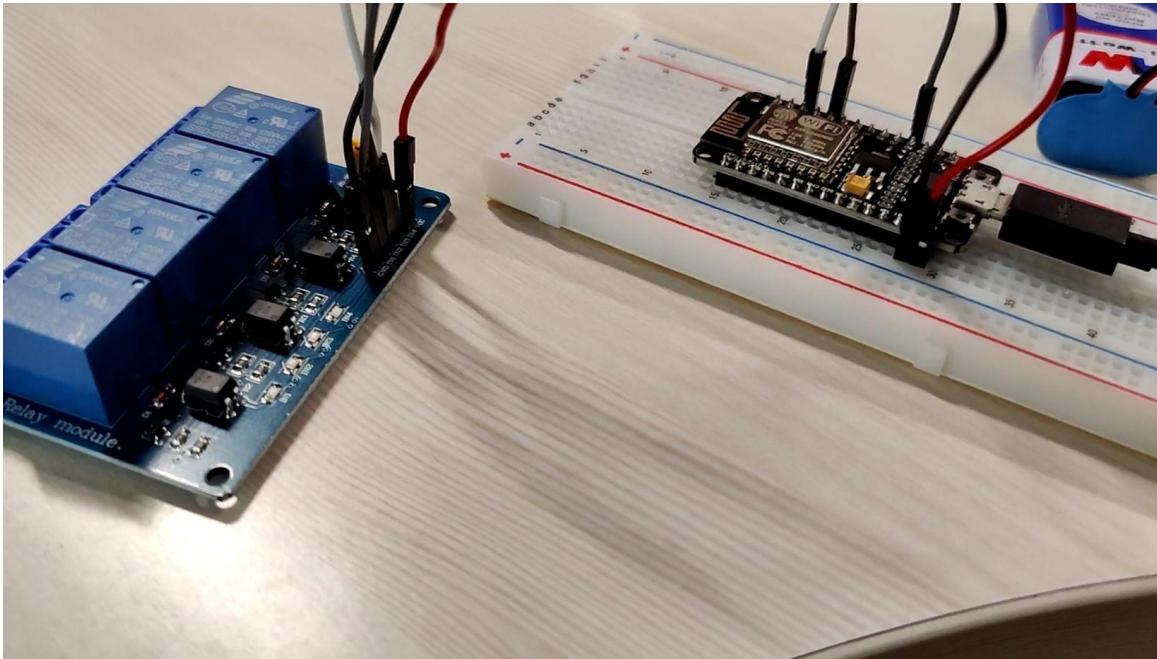
File Edit Format View Help

```
[2022-04-29 13:03:59.323061] Device :fan, Action :on
[2022-04-29 13:04:25.376920] Device :ac, Action :on
[2022-04-29 13:04:32.616579] Device :ac, Action :off
[2022-04-29 13:26:27.630891] Device :fan, Action :on
[2022-04-29 13:26:31.656371] Device :bulb, Action :on
[2022-04-29 13:26:37.087121] Device :ac, Action :on
[2022-04-29 13:26:44.858290] Device :ac, Action :off
[2022-04-29 13:27:22.901014] Device :fan, Action :off
[2022-04-30 12:30:06.088240] Device :fan, Action :on
[2022-05-05 13:33:03.766581] Device :fan, Action :on
[2022-05-05 13:34:17.774617] Device :bulb, Action :on
[2022-05-05 13:34:38.674695] Device :ac, Action :on
[2022-05-05 13:35:36.933000] Device :bulb, Action :off
[2022-05-05 13:41:31.544935] Device :fan, Action :on
[2022-05-05 13:41:35.123541] Device :bulb, Action :on
[2022-05-05 13:41:36.934464] Device :ac, Action :on
[2022-05-05 13:41:41.581784] Device :bulb, Action :off
```

Figure: Message logs saved on system



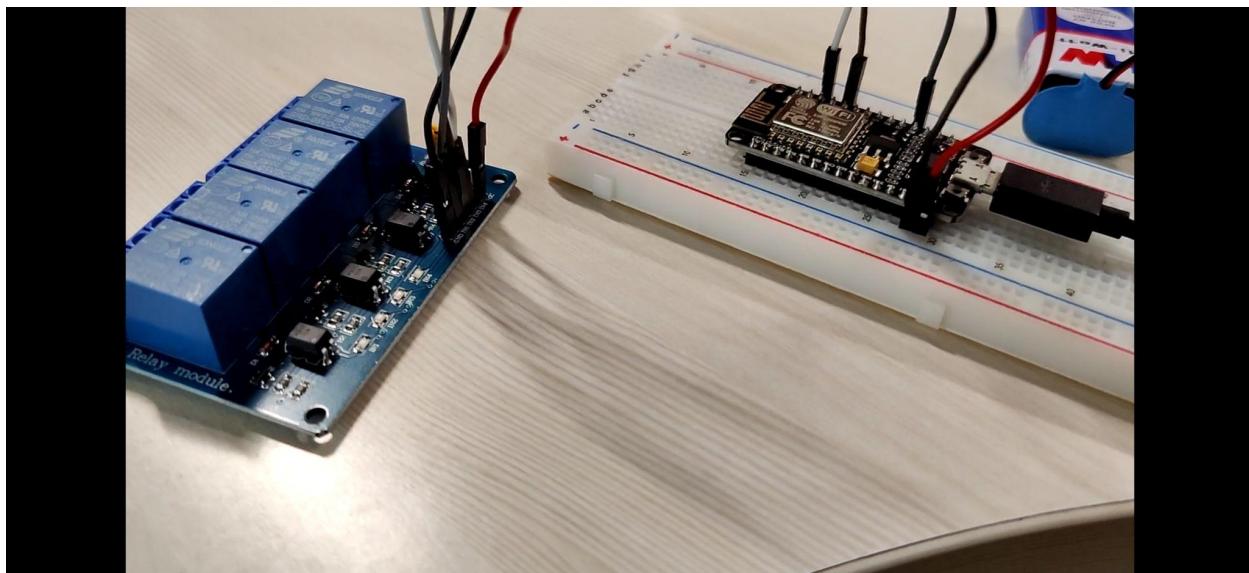
HARDWARE IMPLEMENTATION



Working Demonstration Video is attached with the report.

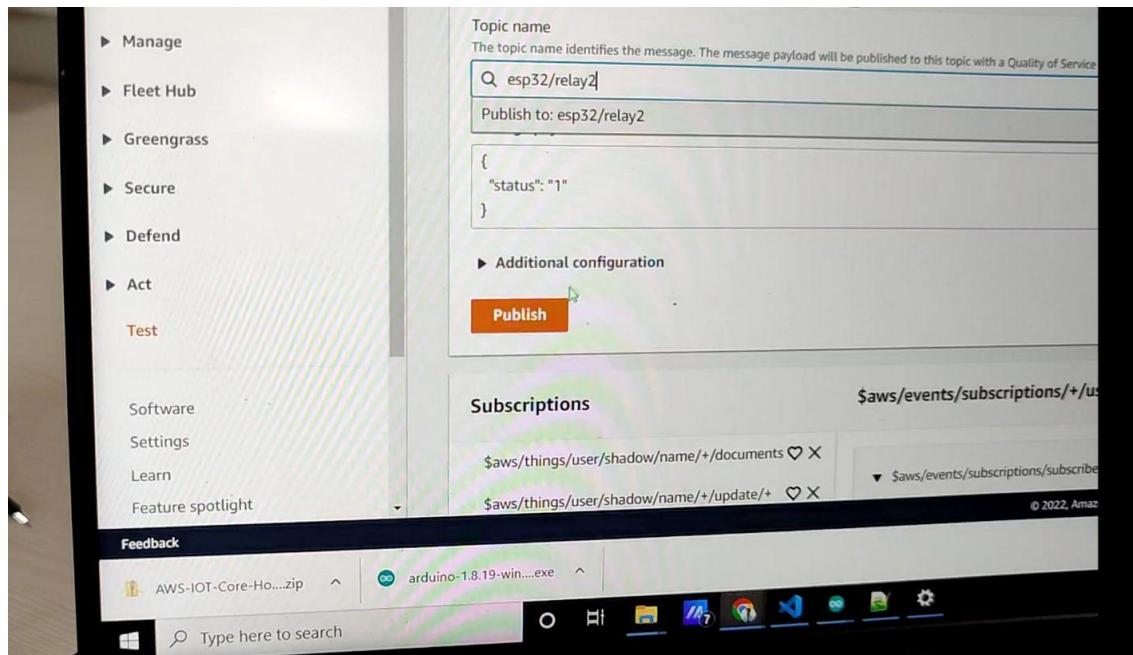
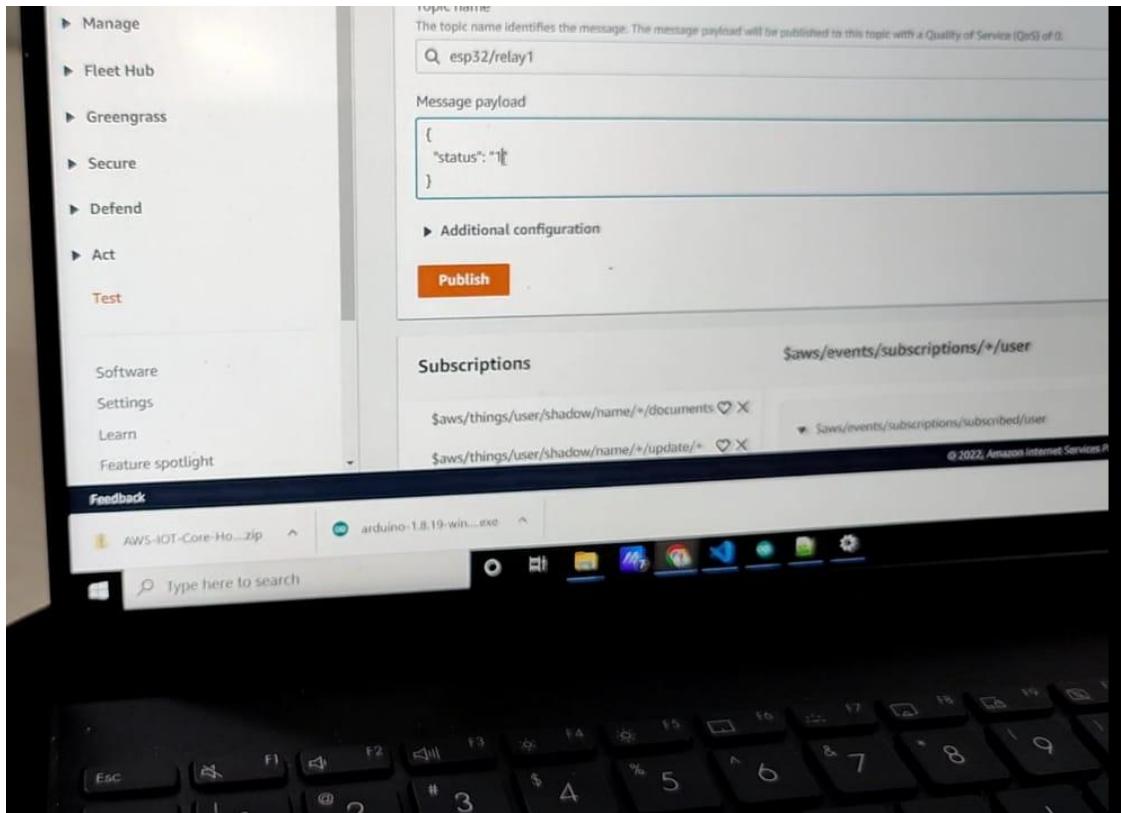
Some Snapshots:

Case 1: Turning relay 1 and 2 on



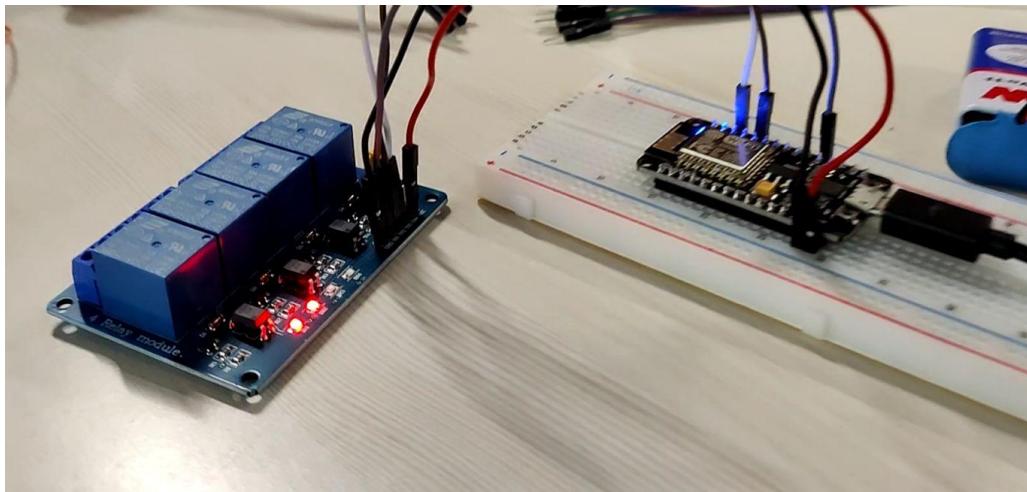


Now , we turn on relay 1 and 2 on AWS IoT core:

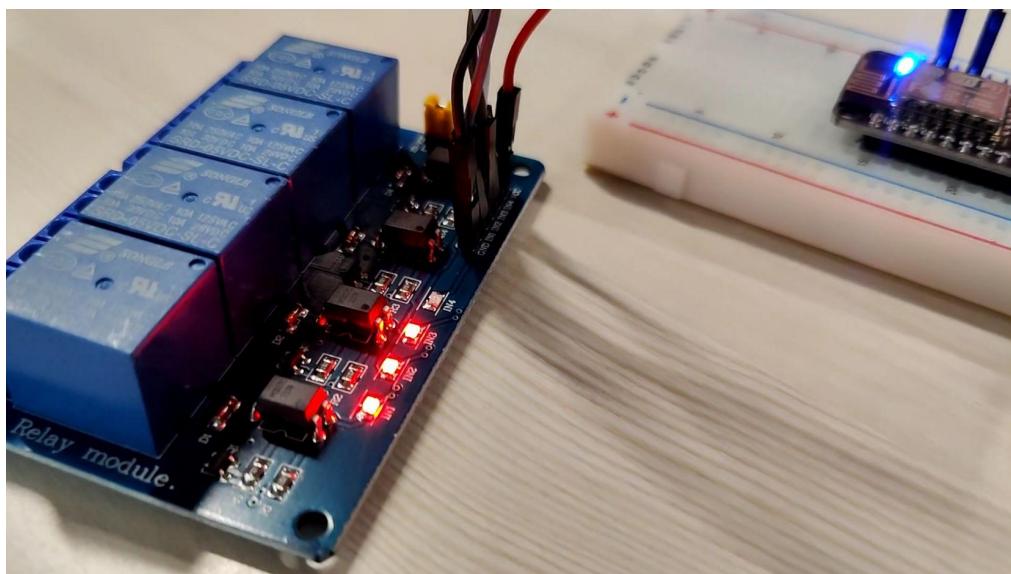




Output:

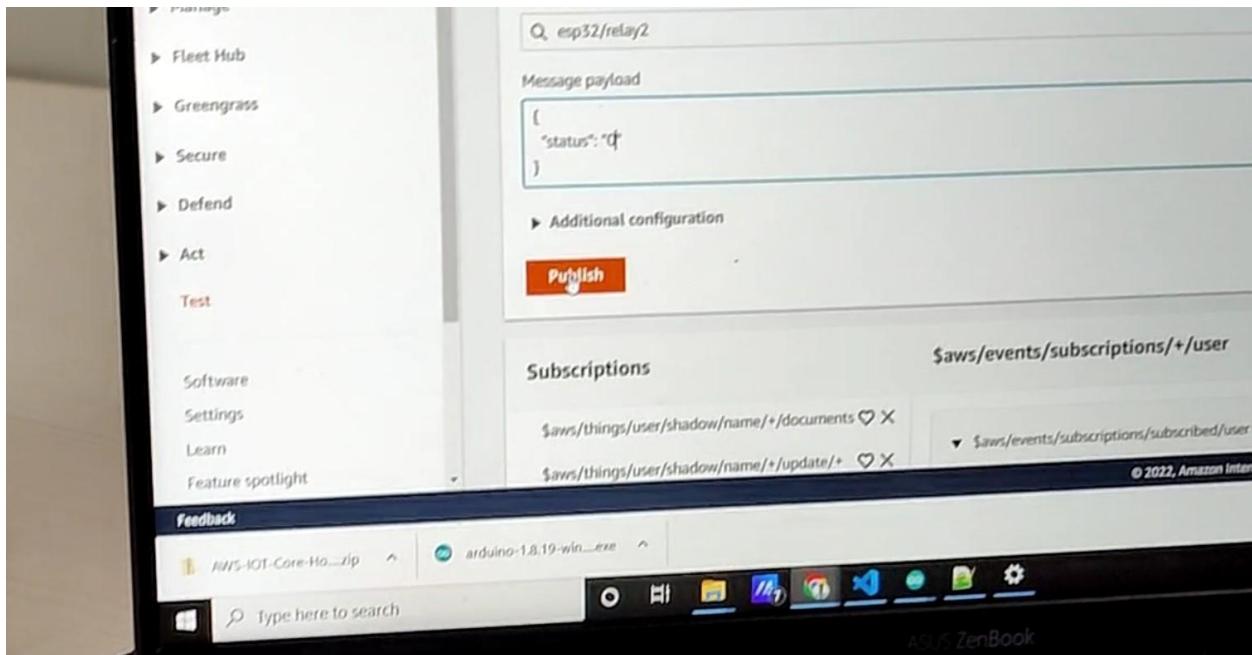


Case 2: Turning relay 2 off when are relays are on

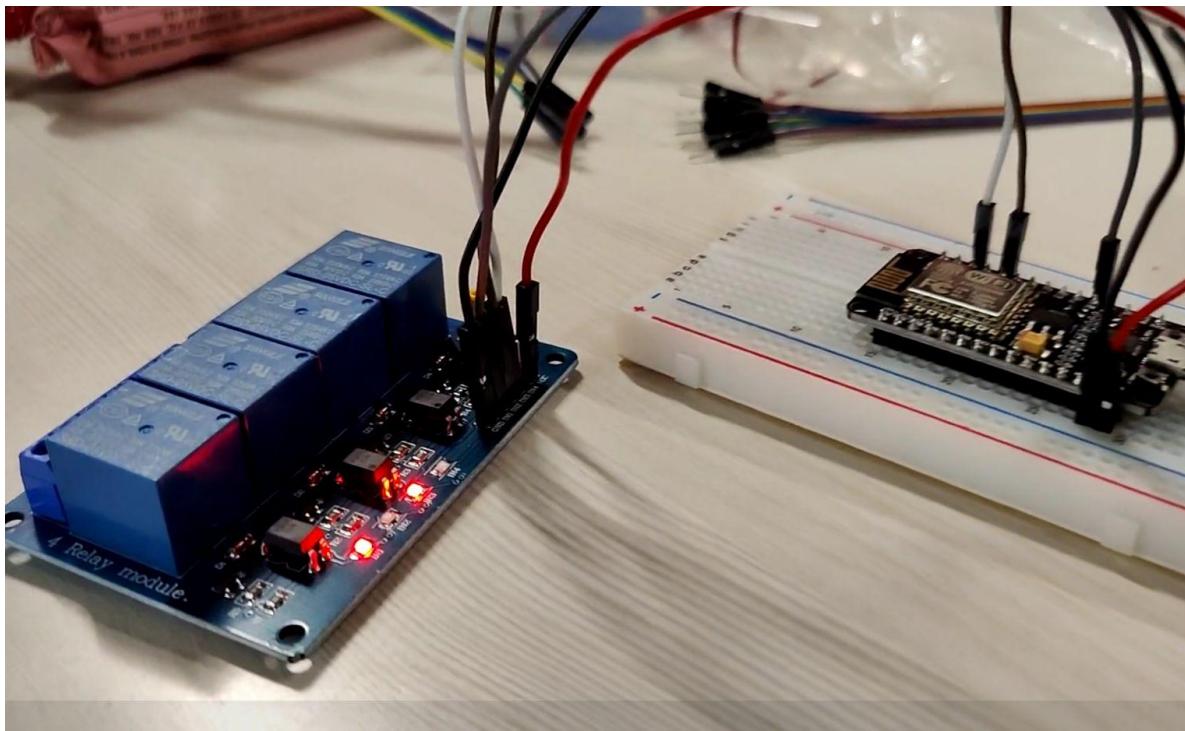




Now , we turn off relay 2 on AWS IoT core:



Output:





CHAPTER 4

TECHNICAL SPECIFICATION



TECHNICAL SPECIFICATION

Item	Specification
Operating System	Windows 10
Programming Languages	Virtualization: Python 3.8 Hardware Implementation(Arduino IDE): C++
Code File	For Virtualization: ➤ user.py (User Module) ➤ bulb.py (Bulb Module) ➤ fan.py (Fan Module) ➤ air_conditioner.py (AC Module) ➤ controller.py (Log Module) For Hardware Implementation: ➤ AWS_Home_Auto.ino
Services Used	AWS IoT Core Lamba Alexa Skills



CHAPTER 5

CONCLUSION AND FUTURE SCOPE



This project has demonstrated that an individual control home automation system can be made inexpensively using low-cost locally available components and can be used to control a wide range of home appliances, including security lamps, televisions, air conditioning systems, and even the entire house lighting system. Even better, the essential components are so small and few that they may be put in a compact, unobtrusive container. This system is scalable and adaptable.

FUTURE SCOPE

In light of the current circumstances, we can develop a dedicated app to implement it. The limitation of being able to control only a few items can be overcome by automating all other home appliances. The prototype could include sensors to automate home appliance control, such as an LDR that detects daylight and switches the lamp accordingly, a PIR that detects motion and sounds an alarm, or a DHT11 sensor that detects ambient temperature and humidity and switches the fan/air conditioner accordingly. This project's scope can be broadened to include not only the home, but also small businesses.



CHAPTER 6

REFERENCES



“IOT Based Smart Security and Home Automation”, by Shardha Soman, Parikshit Solunke, Shaunak Oke, Parth Medhi, Prof. P. P. Laturkar.

“A Low Cost Home Automation System Using Wi-Fi based Wireless Sensor Network Incorporating internet of Things”, by Vikram.N, Harish.K.S, Nihaal.M.S, Raksha Umesh, Shetty Aashik Ashok Kumar; in 2017 IEEE 7th International Advance Computing Conference.

“Voice Controlled Home Automation System using Natural Language Processing and Internet of Things”, by Mrs. Paul Jasmin Rani, Jason Bakthakumar, Praveen Kumaar.B, Praveen Kumaar.U, Santhosh Kumar; in 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)

https://en.wikipedia.org/wiki/Home_automation

<https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>

<https://www.javatpoint.com/arduino-ide>

<https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html>