Project Report On

Echo Board

“8 channel Board controllable using Alexa”

Abstract

The recent trend in technology has given rise a whole new world of digital voice assistants(DVA) viz. Siri, Google assistant and Alexa. These devices can be used to perform and automate tasks like booking tickets, playing songs and setting alarms. With the simultaneous rapid rise of Internet of things and the fact that these devices are cloud based makes it a match made in heaven for these devices to be used for IoT application. One such application where DVAs can be employed is voice and remote control of switch & plug boards. With technology as a brush and our human mind as a canvas, this project aims at painting the picture for a future where switch boards can be controlled with voice when in vicinity and with mobile application over the internet when very far apart. The projects aim at creating a 8 channel switch board which can be controlled with voice and app, along with features of group controlling of switches and time scheduling for reducing power consumption and safety

Keywords: Internet of things, Digital voice assistants, automation, voice control, embedded system, android application, Alexa echo dot.

Keywords : *Home Automation, IOT, Alexa ,Amazon Echo dot, NodeMCU*

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations:

IEEE	Institute of Electrical & Electronics Engineers
DSCE	Dayananda Sagar College of Engineering
ECE	Electronics & Communication Engineering
IoT	Internet of Things
ASK	Amazon Skills Kit
M2M	Machine to Machine
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
SoC	System on Chip
HTML	Hypertext Markup Language
API	Application Programming Interface
REST	Representational State Transfer
AWS	Amazon Web Services
GPIO	General Purpose Input Output

CHAPTER 1

INTRODUCTION

In this 21st century, the advancement in technology and programming has allowed many tasks to be automated, allowing ease and comfort to the humans. Internet of things is one such technology immensely helped humanity by bridging the gap between humans and physical things. With IoT, environmental data can be easily monitored and many devices can be controlled remotely. Apart from IoT, digital voice assistants are the new fascinating trend which has made it to the house of common people, allowing them to play music, or set alarms or even book tickets, all by voice commands. These two technologies are a byproduct of human innovation and this project aims at innovate by combining these two technologies. Echo board focuses on controlling the plugs or switches on a board, through voice or app. The voice recognition feature of Alexa echo board, which is cloud based, fused with electronics of microcontrollers like ESP8266/nodemcu and actuators, can be used to easily build a 8 channel switch/plug board where the switches can be turned on/off over a voice command. It will features controlling a switch individually, or a group of switches through voices as well as traditional mobile application, allowing greater flexibility to the users. The switches can also be scheduled to turn on/off at a particular time which will help in utilizing electrical power consumption efficiently and thereby allowing easy monitoring and control of switches.

1.1 Overview

The project aims to design an 8-channel switch board which can be controlled through voice commands as well as with mobile app remotely over the internet. The existing systems are expensive and this project will focus on designing a cost-effective and easy to integrate switch board which can be used for multi-purposes. The system will give the impetus to user to name the switches according to their wish as that it is convenient for them to control it. This system will be a boon since it offers app and voice control of switches and hence monitor the switches, which can drastically reduce the power consumption since an option to turn on a switch for a specific time is also present.

1.2 Literature survey

Digital voice assistants are have made it to the homes in the past decade. There are many voice assistants like Siri, google assistant, etc. which have been used with IoT already. Automation and internet of things is a very extensive field which has been researched Most of the switch board which are controlled by voice assistant come as a set with its branded devices or appliance and its controlling system which are expensive. Our aim is to provide an economical voice-controlled switch board which includes the can plug in devices such as lights, fan, mobile charger, laptop, etc. which can be affordable to everyone.

According to Vinay and Kusuma [1] the main problem faced in home automation is high cost and poor manageability. Therefore implementation of IOT based home automation can reduce the cost because we are adopting wireless system and hence no cable is required. Author proposes system with greater flexibility by using Wi-Fi technology to interconnect the sensor to the server. Poor manageability is solved because the user can check the status of his house and the devices through a webpage from any part of the world.

V. Këpuska and G. Bohouta [2] wrote about next generation of voice assistants where Siri, Alexa, Cortona,etc. were analyzed based on various parameters such as security, features, reliability etc. It could be concluded that Alexa faired better than the rest due to it features and it support to IoT applications and automation

A. M. D. Celebre, A. Z. D. Dubouzet, et al [3] designed a home automation system which can be controlled through Siri voice assistant. The system was able to voice control but used raspberrypi as a microcontroller, which is costly. Also, it used Siri voice assistant which is compatible with only Apple products. Hence an alternate model which can be cheaper and assessible to all can be used for voice control of home appliances.

] Alshu'Eili, H., Gupta, G. S., & Mukhopadhyay [4] designed a wireless automation system with voice recognition using RF ZigBee and Bluetooth to control the appliance at home. Although it offers security with voice recognition, the system is not robust since it has uses Bluetooth and has very low range

] A. A. Mohammed and E. Erçelebi [5] published a paper on development in embedded system for smart plugs. In this system, the switches/plugs could be controlled remotely over a local area network (LAN) or internet, using a web page. The system achieves the tasks of controlling the switches remotely but uses web page only, and no voice control when near the switches. Hence, another system can be added onto it to control the system using voice command.

1.3 Objectives

The main objective of the ECHO BOARD is to control appliances that is being connected to it. It will be having one skill that will be controlling the commands processing and detecting which device has to be turned ON/OFF.
The project will have following functionalities

1. A 8 channel switch board which can be controlled through voice by using Alexa Echo Dot
2. The board can be controlled by a mobile app over the internet for remote switching on and off
3. The switches on the board should be reconfigurable and can be renamed according to the user’s preference
4. The switches should be controlled individually as well as in a group
5. Time scheduling to keep a switch ON for a specified duration of time

1.4 Methodology

To achieve the goal of controlling switches through voice and app, Alexa echo dot, a digital voice assistant will be used along with relays and a microcontroller to turn on or off the switches. The Alexa skillset will be employed to make custom command for turning the switches on or off.

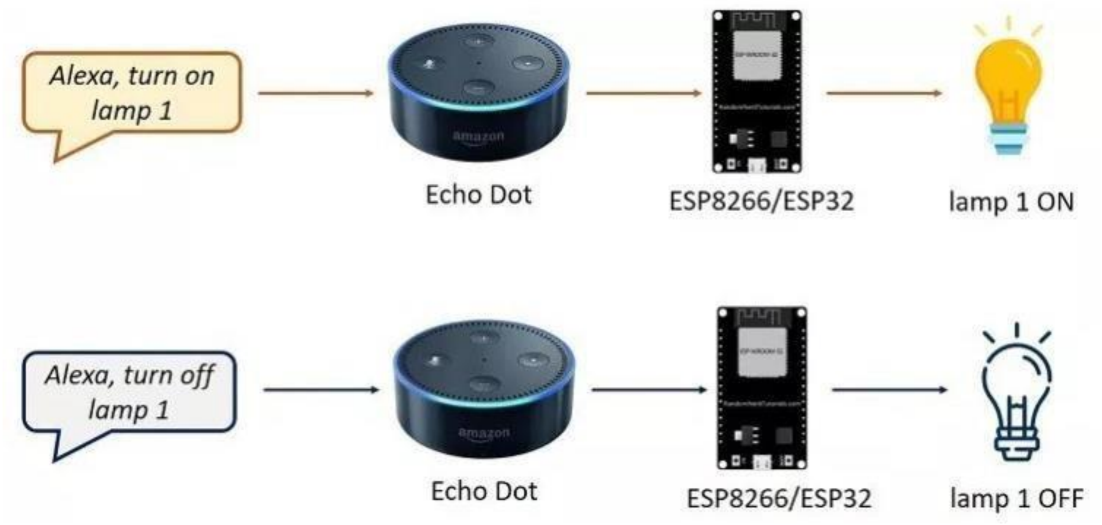


Fig 1.1: Methodology

In fig 1.1, we can see how a lamp connected to the swich can be controlled. A voice command to turn the lamp on is given to alexa, which processes is and send it to the cloud, where it is pulled by the ESP8266 microcontroller which is connected to the internet since it has a WiFi module. The microcontroller runs the controller service and based on the command given, it turns off or on a switch.

The project will replicate this methodology for 8 switches while also grouping two or more switches by custom name and using Alexa scheduling skill set to turn on a switch or group of a specific amount of time.

1.5 Organization of the project report

The project work undertaken by us is organized in the following sequence as follows.

A brief introduction to the work was presented in the introductory chapter in chapter-1.

Block diagram and various design specification for the system were undertaken by us is presented in the chapter – 2.

Hardware and Software tools as well as the technology stack employed in our project work is depicted in the chapter – 3.

Chapter-4 shows the flow of control for a switch in the proposed system. This flowchart depicts how the switches are controlled in the system.

The results of the system design and tests carried out are shown in Chapter-5. Each functionality is discussed in detailed to show what features the system offers .

We then move on to the practical applications where the system can be used. Along with it, the advantages offered by the system as well as its drawbacks are also listed in Chapter-6

Finally, the report concludes with the conclusion and future work in chapter –7.

CHAPTER 2

BLOCK DIAGRAM AND WORKING PRINCIPLE

The purpose and functionality of the system was defined in the previous chapter. The project aims to create a voice and app-controlled switch board which can control each plug individually as well as in groups. In order to achieve this, a domain level specification of the system has to be defined. Fig 2.1 shows the domain level specification of the proposed work in the project.

Domain Level Specification

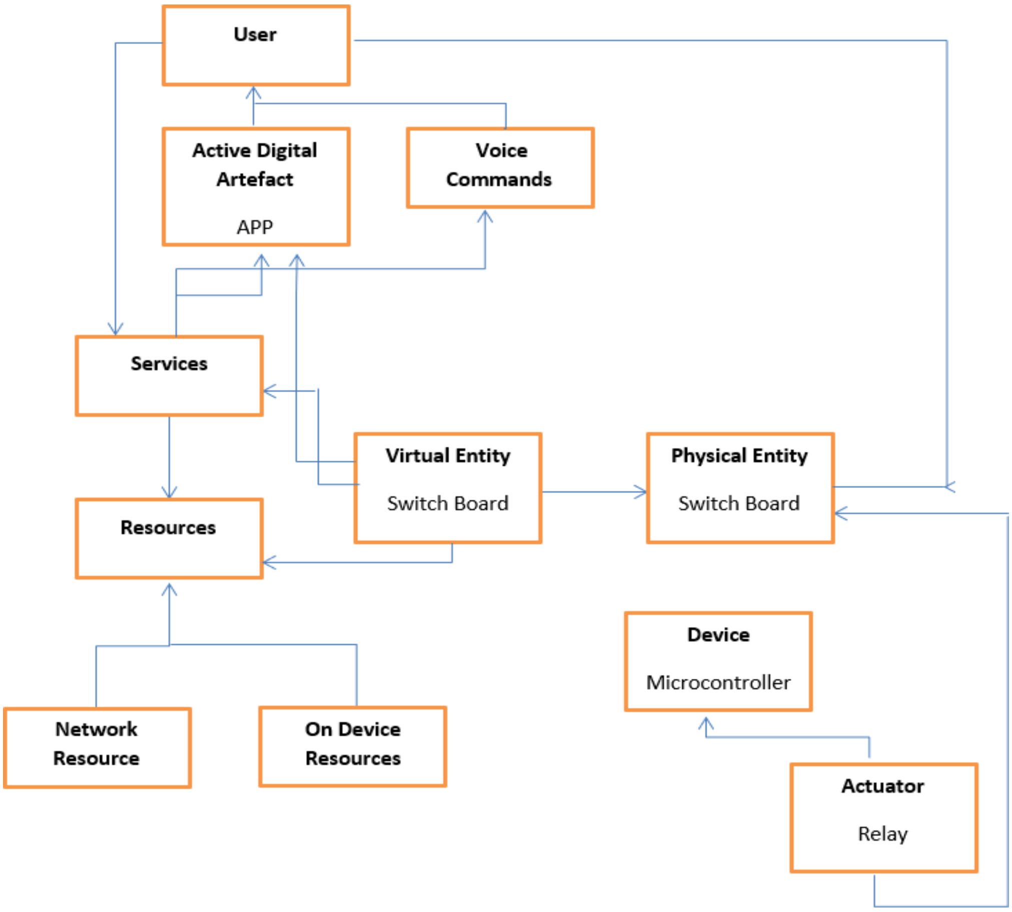


Fig 2.1: Domain level specification

The domain model describes the main concepts, entities and objects in the domain of IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed .For the domain model of this project, user is defined as an entity who can control the switch board with either a digital artifact i.e. an application or through his voice command. The switch board where the switches have to be controlled is another physical entity, but for design purposes and representation, its virtual counterpart is represented. The switch board needs to have access to resource i.e. power to function. It is controlled by a microcontroller which employs actuators known as relay to control the state of the switches on the board. Also, both the entities i.e. the user app & voice command devices as well as the switch board run on some services, which require either onboard resources or network resources.

IoT level specification

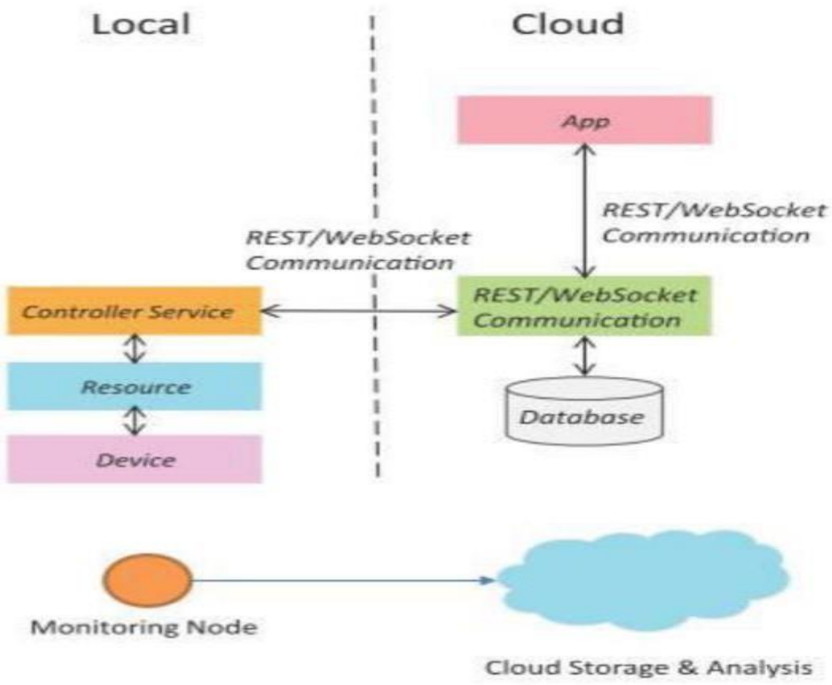


Fig 2.2: IoT level specification

Internet of Things level specification is step in IoT system design methodology which aims at breaking up the system in sub part and define which sub-system is local and which ones on the cloud.

The monitoring node is our system is the switch-board. It is locally placed but all the information about its state are on the cloud. The device i.e. the microcontroller & voice processing devices and the resources they use are also locally placed. The controller service or the program running on the microcontroller is local but it is interacting with the cloud through internet over HTTP or REST based communication. The Application which controls the switches on the board as well as the information about the state of switches, which we can call the database, are to be stored on the cloud in our system.

Operational view specification

The next step is to define the operation view of the system. In this specification, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc.

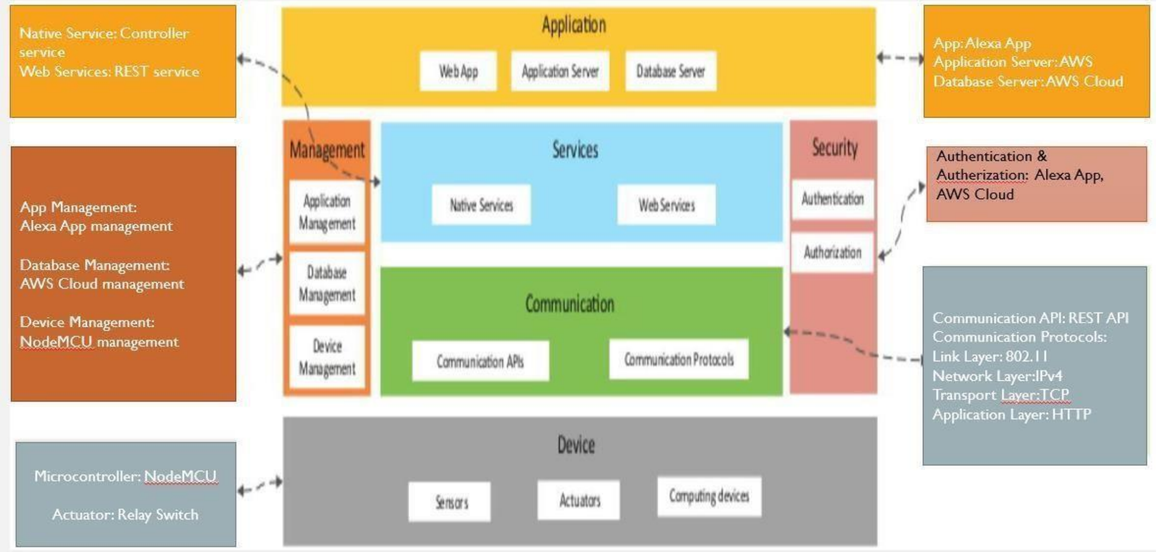


Fig 2.3: Operational view specification

The operational view specification of the system is shown in Fig 2.3. The device required in the system are nodeMCU as a microcontroller, an Alexa Echo dot for voice command listening and relay switches for controlling the switch board. AWS cloud authentication and authorization is used for this project, which runs on REST API as the communication protocols. AWS cloud is used as application as well as data cloud server. The mobile app deployed is the Alexa app service.

Solution design

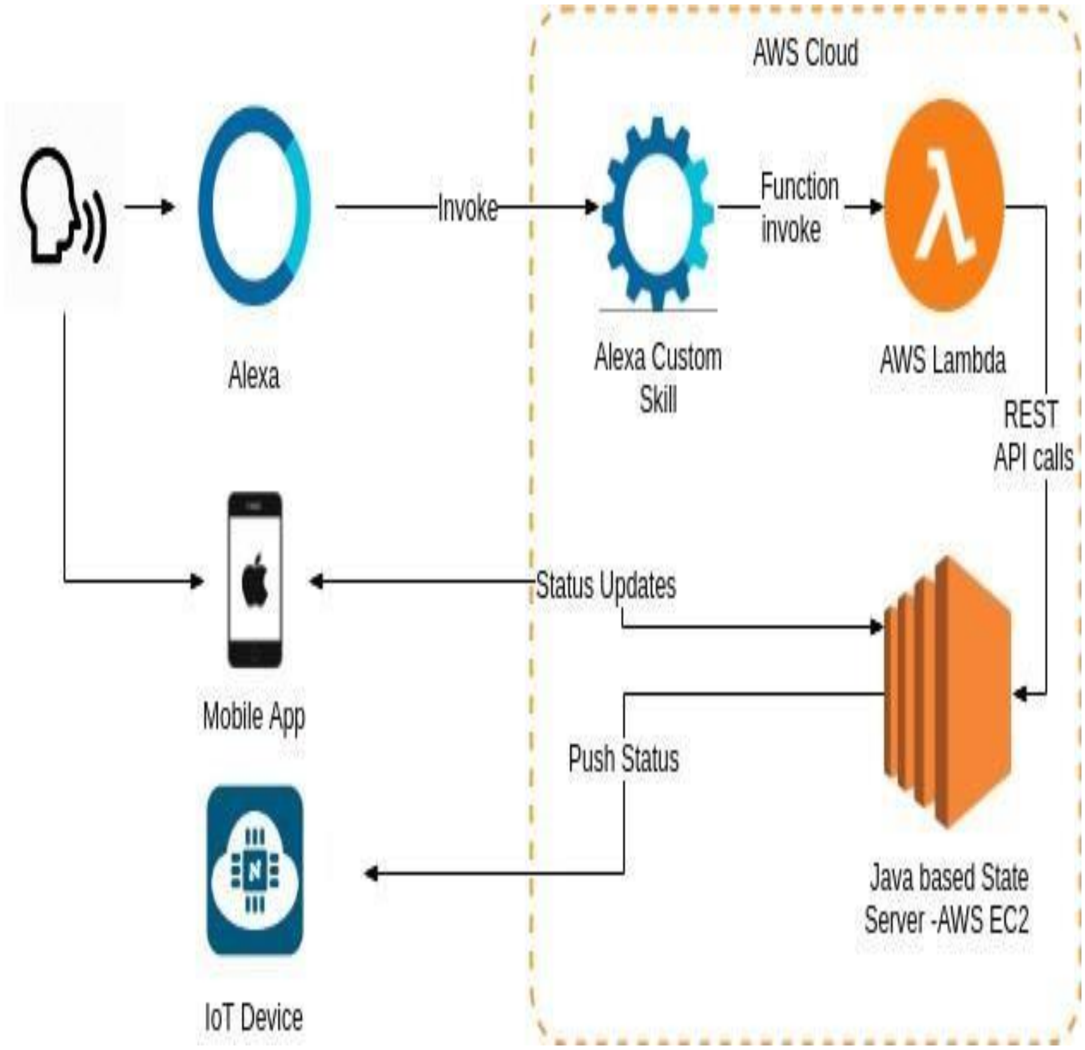


Fig 2.4: Solution design

To achieve the goal of controlling the switches using voice and mobile application, Alexa echo dot is used. The user can control the switch by giving a voice command to alexa echo dot. Alexa Voice Service is the intelligent voice control service that powers the device.Alexa uses natural language processing techniques trained by the developers and the user community of Amazon to process user requests and cater to their individual needs. The voice service can be triggered using the keyword “Alexa” the skill/application that we have developed can be triggered using the voice command, “Alexa, Turn the lights on”. Once Alexa is triggered, it runs a script on the cloud, which in-turn runs a subroutine on the NodeMCU to Switch on the light. Once computation is done in the NodeMCU, it sends the confirmation back to Alexa. Only the essential information is sent to Alexa which passes it on back to the user. Same can also be achieved using a mobile application interface, where toggle buttons to switch on or off the device are given and when the user toggles to turn on or off a switch, it switches on or off respectively.

CHAPTER 3

HARDWARE/ SOFTWARE TOOLS

In this chapter, the hardware & the software descriptions related to the project work is presented in brief. This project has a hardware as well as a software stack since its based on IoT. Below are the components used for this project

Hardware:

The hardware components used are:

1. Amazon Echo Dot 3rd Generation
2. ESP8266-12e (NodeMCU)
3. 8 channel Relay Board

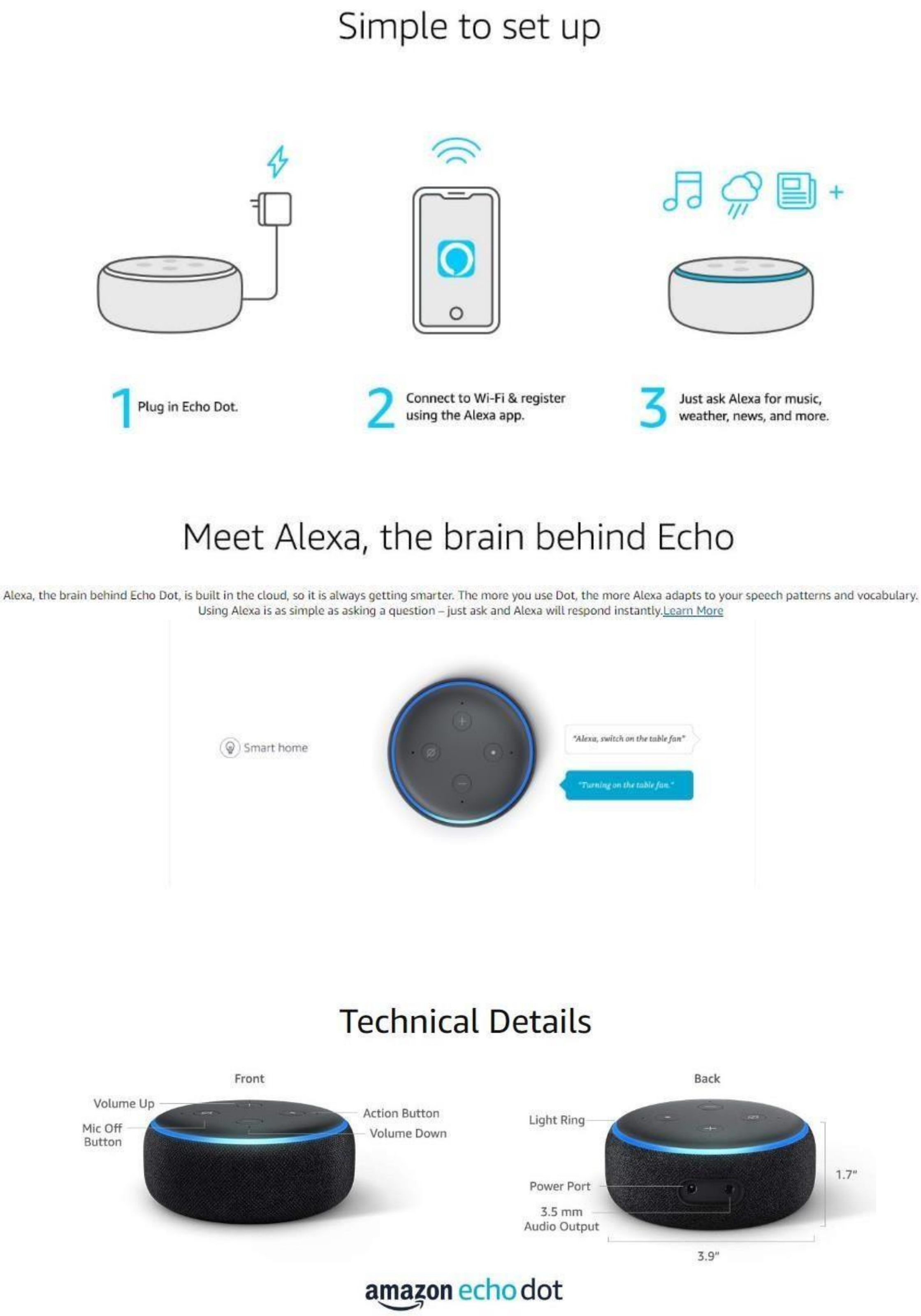
1. Amazon Echo Dot 3rd Generation:

Fig 3.1: Amazon Echo dot

The Echo Dot is Amazon’s small, puck-like smart speaker with the firm’s Alexa voice assistant built in. New for the third generation is a softer, more rounded aesthetic with fabric sides. It is also slightly larger in all directions, measuring 99mm in diameter and 43mm tall. At the back is a circular power socket, replacing the microUSB socket of the old Dot, and a standard 3.5mm analogue socket for connecting to a stereo. On top you have the volume buttons, action button and the microphone mute button. You can also use Bluetooth to stream from the Dot to other speakers or from your phone or other device to the Dot to use it as Bluetooth speaker.

Features:

- Echo Dot is our most popular voice-controlled speaker, now with an improved sound and a new design.
- Echo Dot connects to Alexa, a cloud-based voice service, to play music, answer questions, read the news, check the weather, set alarms, control compatible smart home devices, and more.
- Stream music from Amazon Prime Music, Saavn, and TuneIn – just ask for a song, artist, or genre. Fill your whole home with music with multiple Echo devices across different rooms.
- Control Echo Dot hands-free – it can hear you from across the room with 4 far-field microphones, even in noisy environments or while playing music.
- Call or message family and friends who have an Echo device or the Alexa App and use Alexa to make Skype calls. Use announcements like a one-way intercom to broadcast messages to all your Echo devices.
- Control compatible smart lights, plugs, and remotes from Philips, Syska, TP-Link and Oakter – just using your voice.
- Use the built-in speaker or connect to speakers through Bluetooth or audio cable for bigger sound.
- 15,000+ skills including ordering cab, food, jokes and games. Always getting smarter and adding new features.



Simple to set up

1 Plug in Echo Dot.

2 Connect to Wi-Fi & register using the Alexa app.

3 Just ask Alexa for music, weather, news, and more.

Meet Alexa, the brain behind Echo

Alexa, the brain behind Echo Dot, is built in the cloud, so it is always getting smarter. The more you use Dot, the more Alexa adapts to your speech patterns and vocabulary. Using Alexa is as simple as asking a question – just ask and Alexa will respond instantly.Learn More

Technical Details

Fig 3.2: Features & Technical details of Amazon Echo Dot 3

2. ESP8266-12e (NodeMCU):

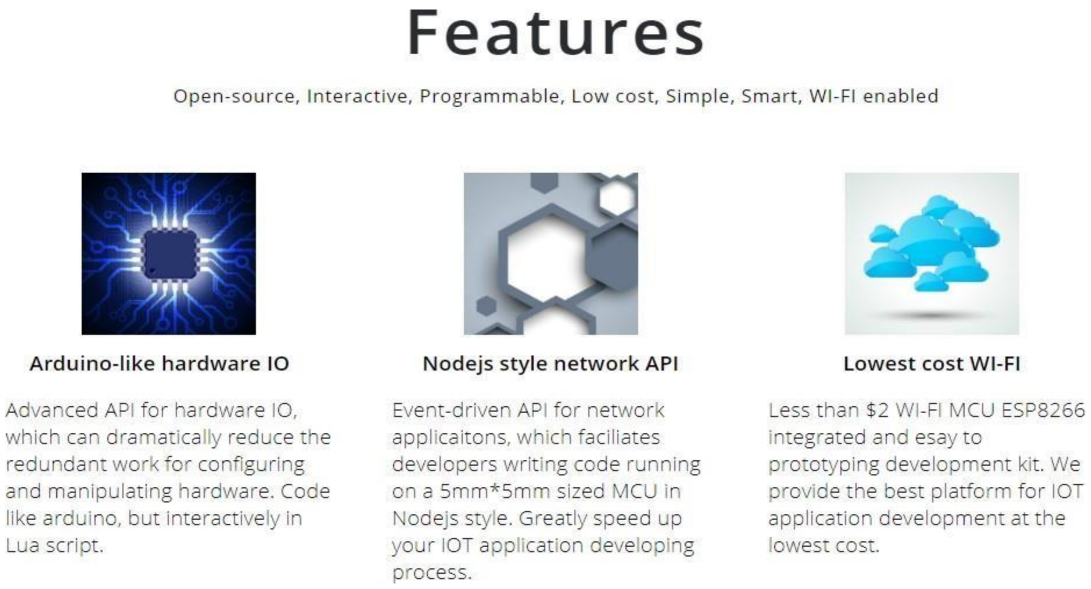
Fig 3.3: NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controllerbased design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. ESP8266EX also integrates an enhanced version of Tensilica’s L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

Features:

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified



Features

Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled

Arduino-like hardware IO

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like arduino, but interactively in Lua script.

Nodejs style network API

Event-driven API for network applicaitons, which faciliates developers writing code running on a 5mm*5mm sized MCU in Nodejs style. Greatly speed up your IOT application developing process.

Lowest cost WI-FI

Less than $2 WI-FI MCU ESP8266 integrated and esay to prototyping development kit. We provide the best platform for IOT application development at the lowest cost.

Fig 3.4: Features of NodeMCU

Ultra Low Power Technology:

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode. By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point. When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval. The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed. In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

Applications:

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics
- WiFi Location-aware Devices
- Security ID Tags
- WiFi Position System Beacons

Internal Block Diagram:

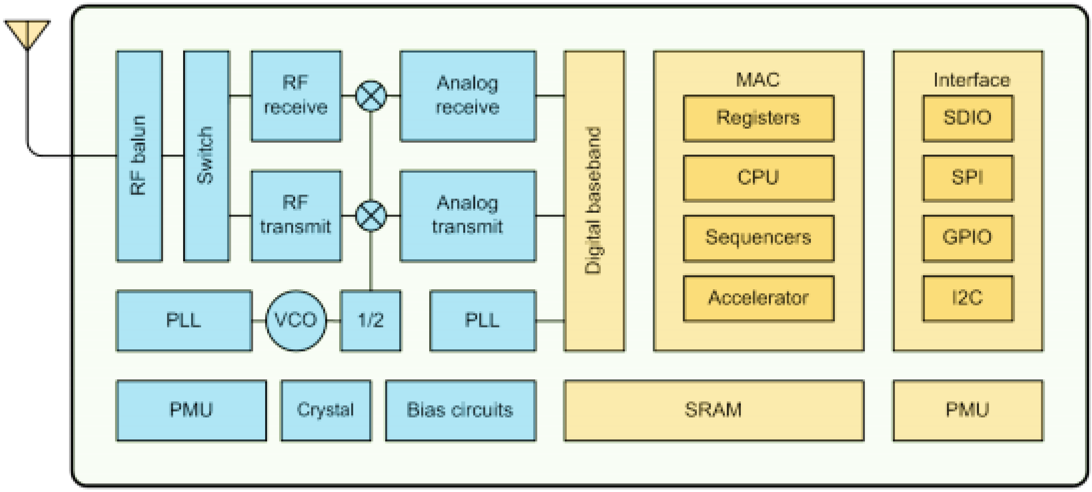


Fig 3.5: NodeMCU internal block diagram

Pin Configuration:

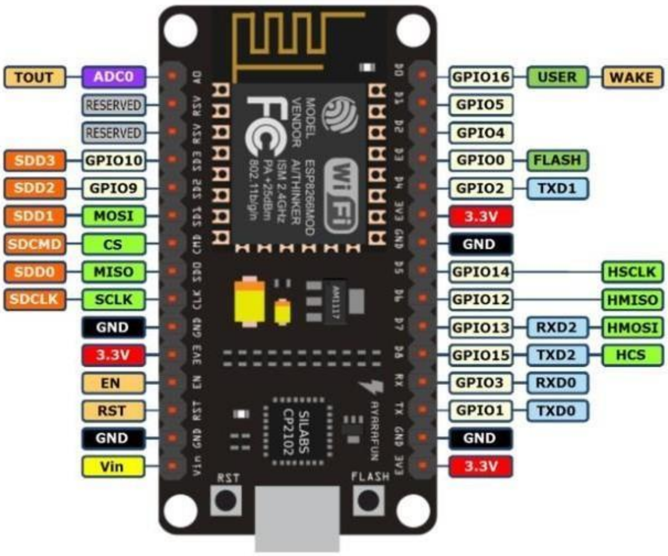
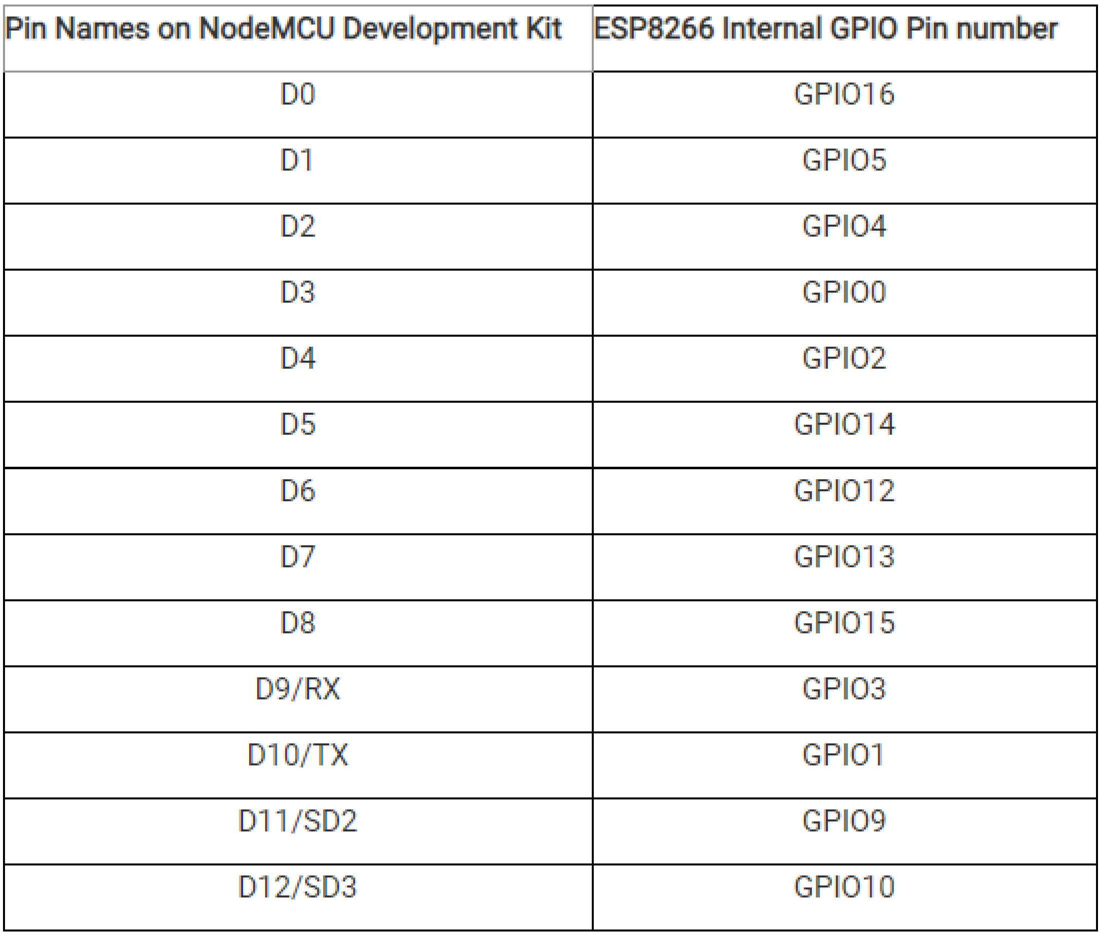


Fig 3.6: NodeMCU Dev-Kit GPIO

Below table gives NodeMCU Dev Kit IO pins and ESP8266 internal GPIO pins mapping:



Pin Names on NodeMCU Development Kit	ESP8266 Internal GPIO Pin number
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9/RX	GPIO3
D10/TX	GPIO1
D11/SD2	GPIO9
D12/SD3	GPIO10

Table 3.1: Internal GPIO specifications of NodeMCU

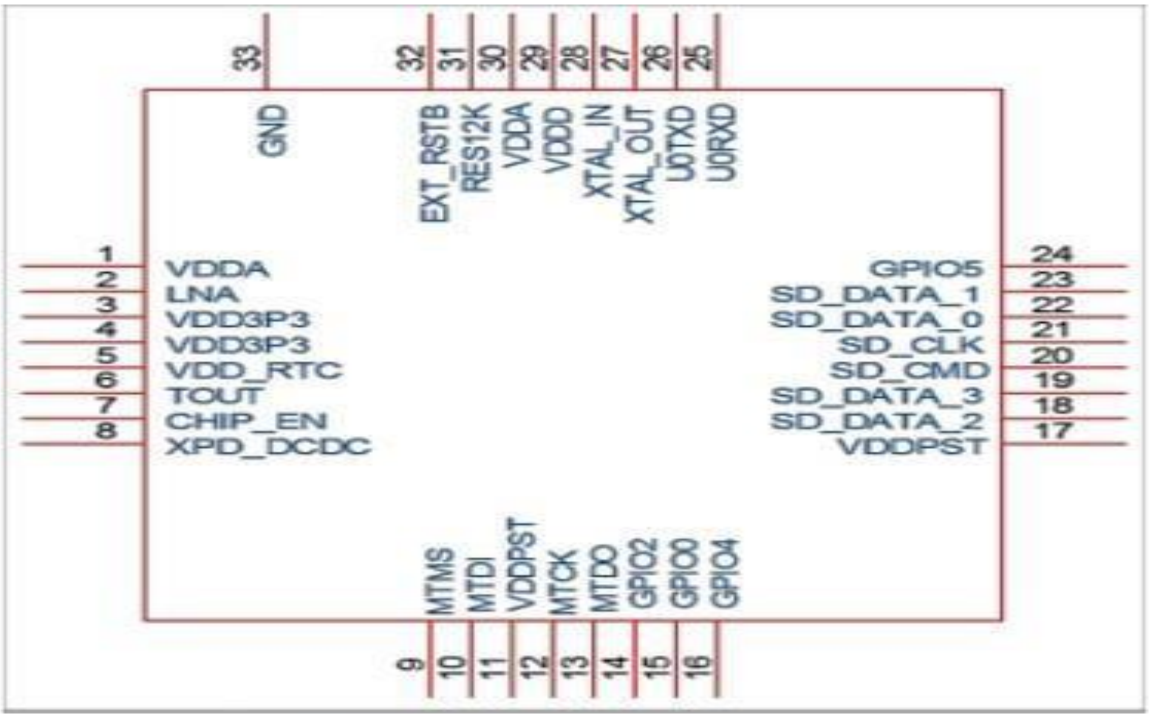
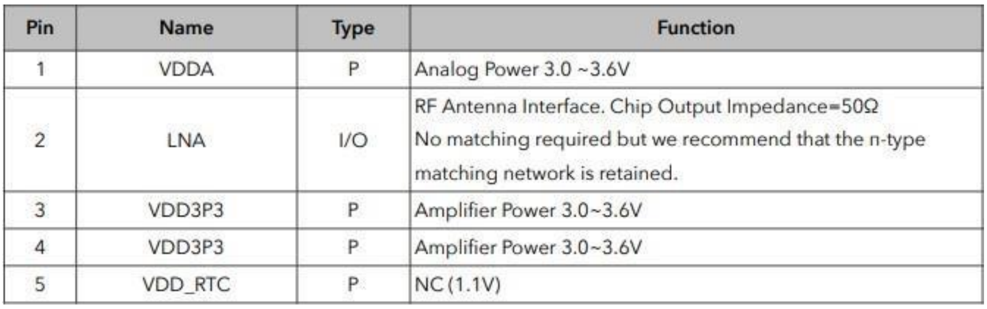
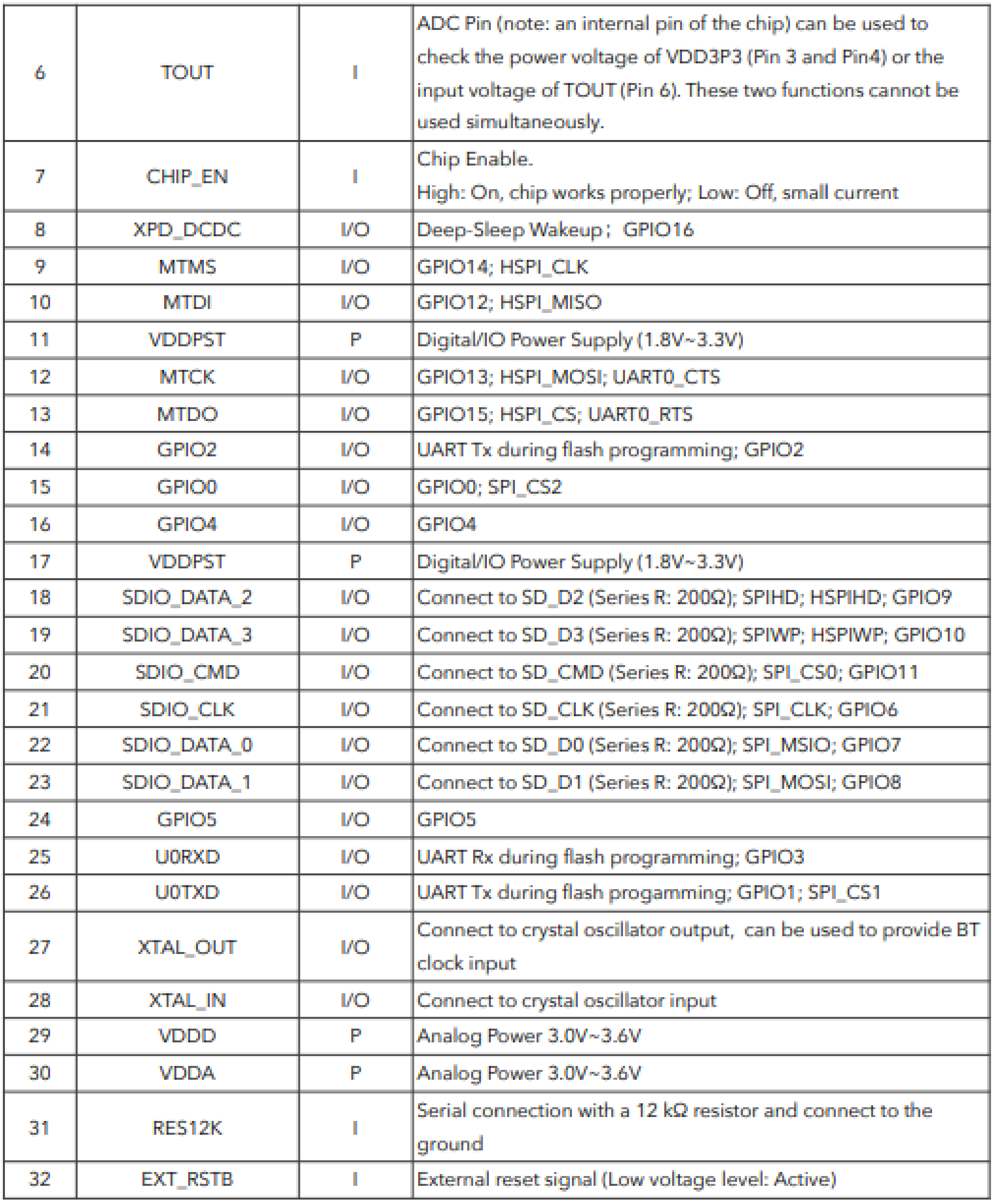


Fig 3.8: Pin assignment for 32-pin QFN package



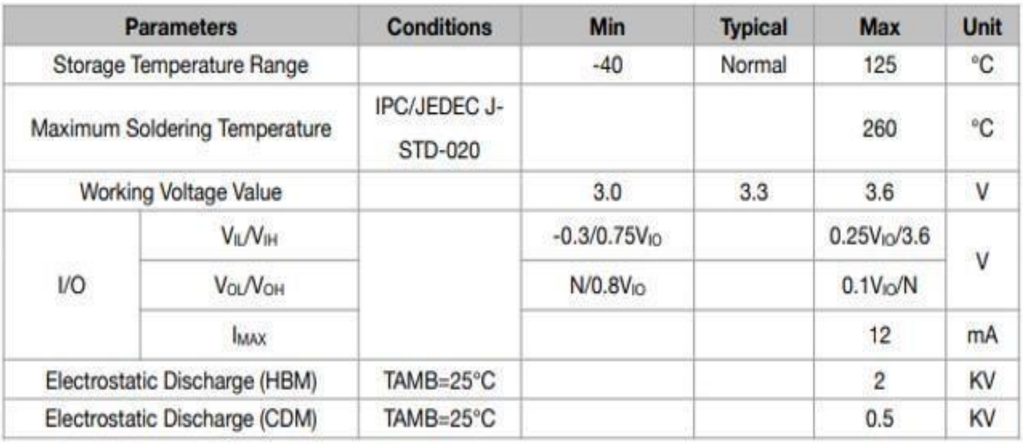
Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)



6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup； GPIO16
9	MTMS	I/O	GPIO14; HSPI_CLK
10	MTDI	I/O	GPIO12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash progamming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V~3.6V
30	VDDA	P	Analog Power 3.0V~3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)

Table 3.2: Functionalities of each pin of a 32-bit QFN package

Electrical Characteristics:

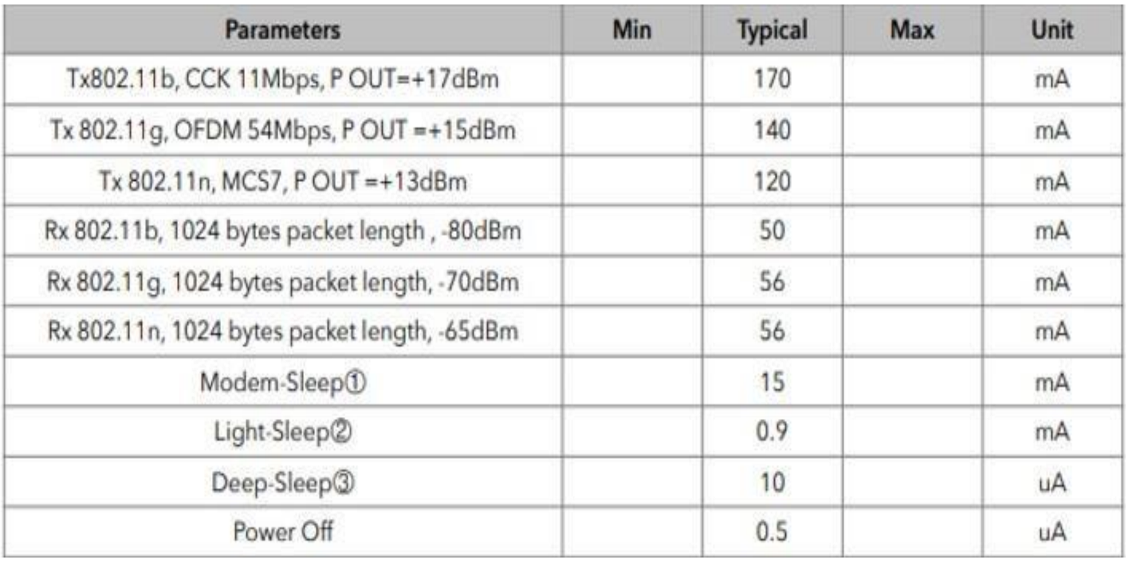


Parameters		Conditions	Min	Typical	Max	Unit
Storage Temperature Range			-40	Normal	125	°C
Maximum Soldering Temperature		IPC/JEDEC J-STD-020			260	°C
Working Voltage Value			3.0	3.3	3.6	V
I/O	V_{IL}/V_{IH}		-0.3/0.75V_{IO}		0.25V_{IO}/3.6	V
	V_{OL}/V_{OH}		N/0.8V_{IO}		0.1V_{IO}/N	
	I_{MAX}				12	mA
Electrostatic Discharge (HBM)		TAMB=25°C			2	KV
Electrostatic Discharge (CDM)		TAMB=25°C			0.5	KV

Table 3.3: ESP8266EX Electrical charachterstics

Power Consumptions:

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.



Parameters	Min	Typical	Max	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm		140		mA
Tx 802.11n, MCS7, P OUT =+13dBm		120		mA
Rx 802.11b, 1024 bytes packet length , -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA

Table 3.4: Description on Power Consumption

3. 8 Channel Relay Board:

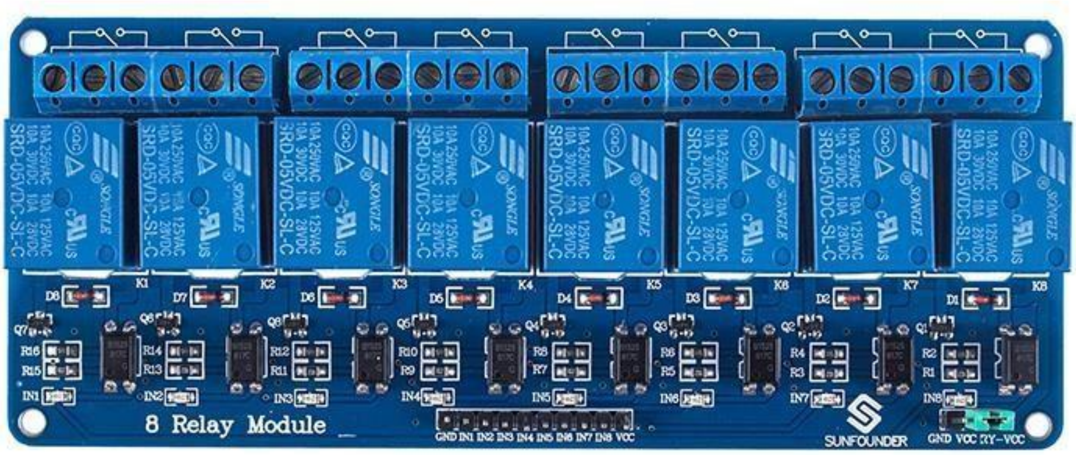


Fig 3.9: Channel Relay Board

This is a LOW Level 5V 8-channel relay interface board, and 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 microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.

Brief Data:

- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- 8 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.
- Module Board: 138 x 56 mm.

Operating Principle:

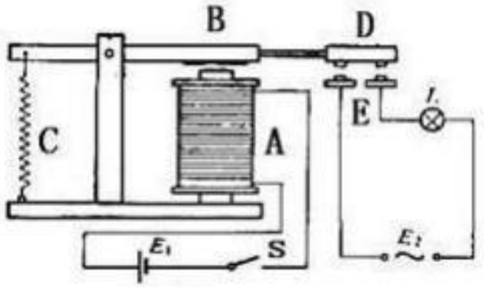


Fig 3.10: Operation diagram

A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on. Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts result in power on and off of the circuit

Software:

1. Arduino IDE:
2. Amazon Alexa Skill Kit
3. AWS Lambda

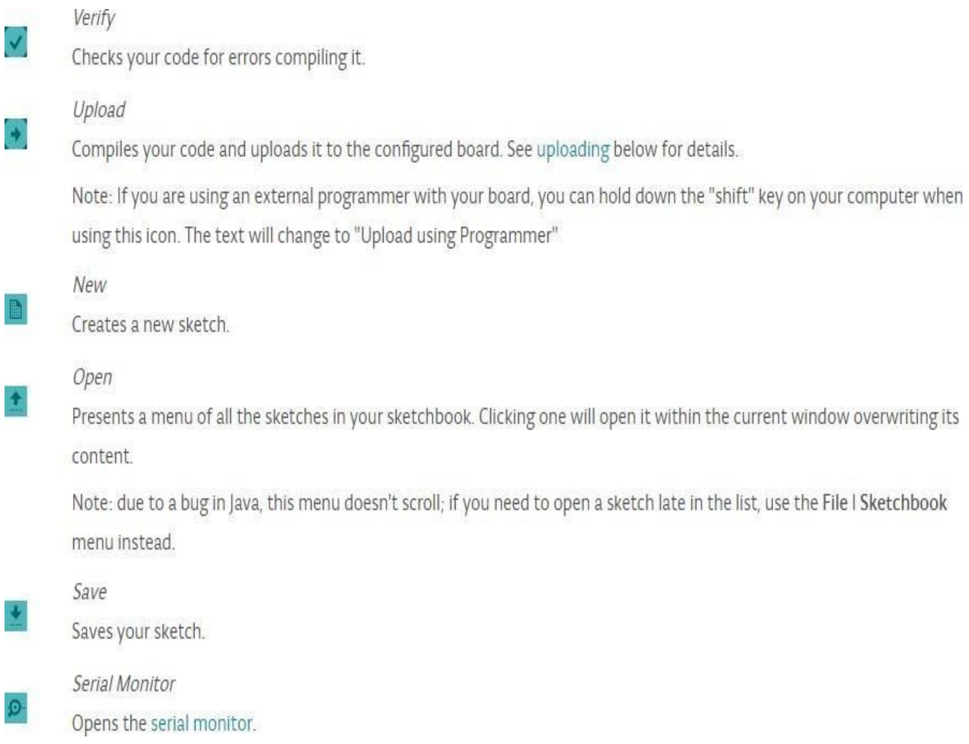
1. Arduino IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Keys functionalities on Arduino IDE:



Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the **File | Sketchbook** menu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

Fig 3.11: Key functionalities of Arduino IDE

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the **File** > **Sketchbook** menu or from the **Open** button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the **Preferences** dialog.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the **Tools** > **Board** and **Tools** > **Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1** (for a Duemilanove or earlier USB board), or **/dev/**tty.USA19QW1b1P1.**1** (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, itshould be **/dev/ttyACMx** , **/dev/ttyUSBx** or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch** > **Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include**statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to **Serial.begin** in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

2. Alexa Skill Kit:

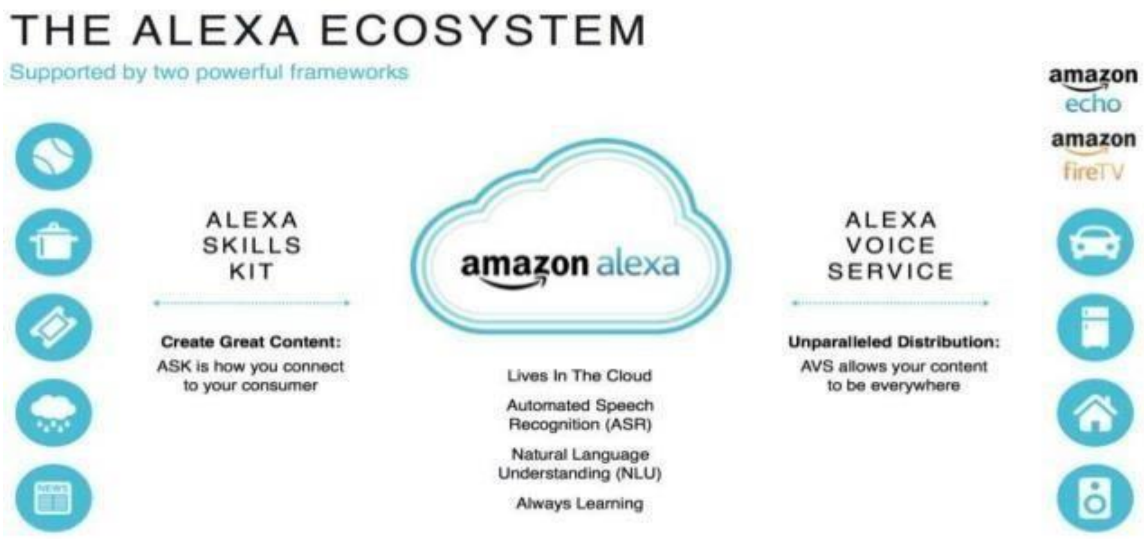

THE ALEXA ECOSYSTEM
Supported by two powerful frameworks
ALEXA SKILLS KIT
Create Great Content:
ASK is how you connect to your consumer
amazon alexa
Lives In The Cloud
Automated Speech Recognition (ASR)
Natural Language Understanding (NLU)
Always Learning
ALEXA VOICE SERVICE
Unparalleled Distribution:
AVS allows your content to be everywhere
amazon echo
amazon fireTV

Fig 3.11: Alexa skill set

What Is an Alexa skill?

Alexa is Amazon’s voice service and the brain behind millions of devices including Amazon Echo. Alexa provides capabilities, or skills, that enable customers to create a more personalized experience. There are now thousands of skills available in India from brands like Saavn, Zomato, Times of India, and Goibibo as well as many other innovative designers and developers.

What Is the Alexa Skills Kit?

With the Alexa Skills Kit (ASK), designers, developers, and brands can build engaging skills and reach new customers. ASK is a collection of self-service APIs, tools, documentation, and code samples that make it fast and easy for you to add skills to Alexa. With ASK, you can leverage Amazon’s knowledge and pioneering work in the field of voice design.

What you can build with ASK?

- Custom Skills

- Flash Briefing Skills
- Smart Home Skills

Brief Working of ASK:

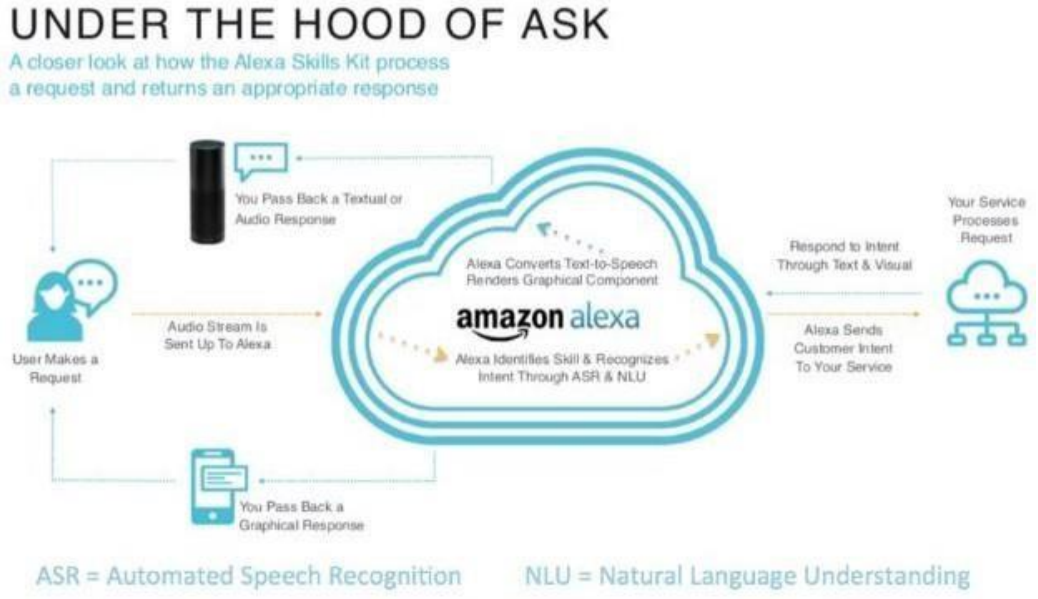
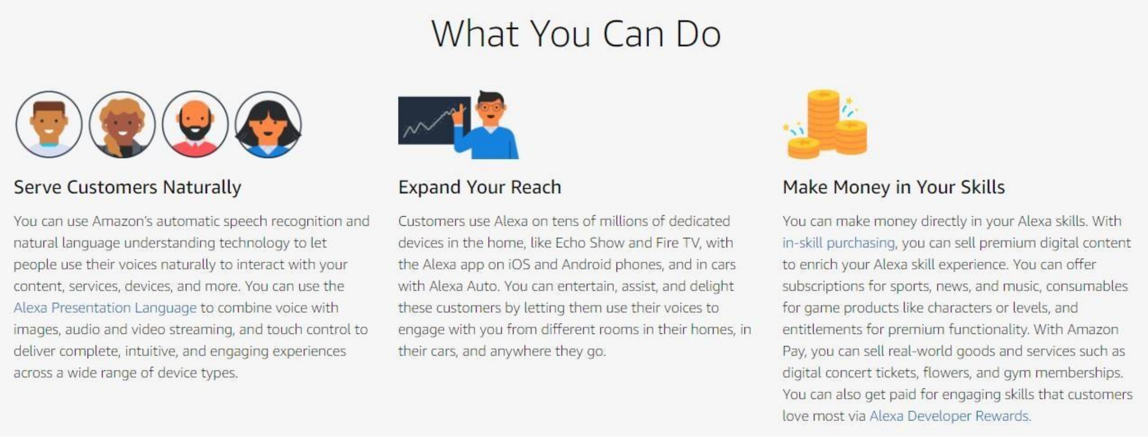

UNDER THE HOOD OF ASK
A closer look at how the Alexa Skills Kit process a request and returns an appropriate response
You Pass Back a Textual or Audio Response
Alexa Converts Text-to-Speech Renders Graphical Component
amazon alexa
Alexa Identifies Skill & Recognizes Intent Through ASR & NLU
User Makes a Request
Audio Stream Is Sent Up To Alexa
Your Service Processes Request
Respond to Intent Through Text & Visual
Alexa Sends Customer Intent To Your Service
You Pass Back a Graphical Response
ASR = Automated Speech Recognition
NLU = Natural Language Understanding

Fig 3.12: Brief working about working of Alexa Skill Kit

Why Build Alexa Skills?

Alexa is Amazon's cloud-based voice service and the brain behind tens of millions of devices including the Echo family of devices, FireTV, Fire Tablet, and third-party devices with Alexa built-in. You can build voice experiences, or skills, that make everyday tasks faster, easier, and more delightful for customers.

Tens of thousands of developers have built skills using the Alexa Skills Kit (ASK), a collection of self-service APIs, tools, documentation, and code samples. With ASK, anyone can leverage Amazon's knowledge in voice design to build quickly and easily. Start building today to reimagine your customer experience for voice and reach customers where they are.



What You Can Do

Serve Customers Naturally

You can use Amazon's automatic speech recognition and natural language understanding technology to let people use their voices naturally to interact with your content, services, devices, and more. You can use the Alexa Presentation Language to combine voice with images, audio and video streaming, and touch control to deliver complete, intuitive, and engaging experiences across a wide range of device types.

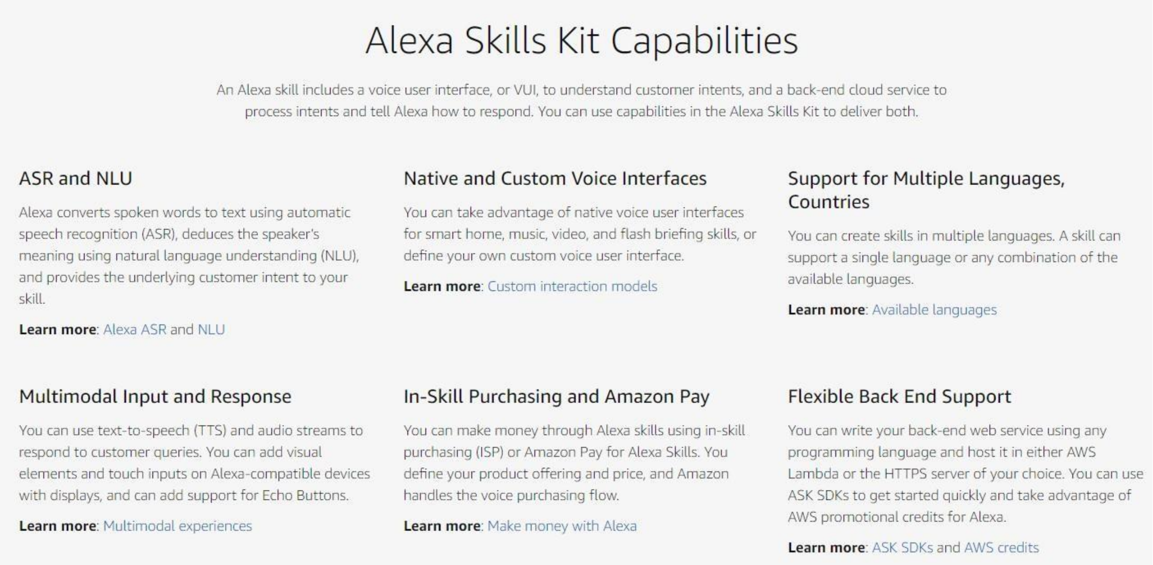
Expand Your Reach

Customers use Alexa on tens of millions of dedicated devices in the home, like Echo Show and Fire TV, with the Alexa app on iOS and Android phones, and in cars with Alexa Auto. You can entertain, assist, and delight these customers by letting them use their voices to engage with you from different rooms in their homes, in their cars, and anywhere they go.

Make Money in Your Skills

You can make money directly in your Alexa skills. With in-skill purchasing, you can sell premium digital content to enrich your Alexa skill experience. You can offer subscriptions for sports, news, and music, consumables for game products like characters or levels, and entitlements for premium functionality. With Amazon Pay, you can sell real-world goods and services such as digital concert tickets, flowers, and gym memberships. You can also get paid for engaging skills that customers love most via Alexa Developer Rewards.

Fig 3.13: What you can do with ASK



Alexa Skills Kit Capabilities

An Alexa skill includes a voice user interface, or VUI, to understand customer intents, and a back-end cloud service to process intents and tell Alexa how to respond. You can use capabilities in the Alexa Skills Kit to deliver both.

ASR and NLU

Alexa converts spoken words to text using automatic speech recognition (ASR), deduces the speaker's meaning using natural language understanding (NLU), and provides the underlying customer intent to your skill.

Learn more: Alexa ASR and NLU

Native and Custom Voice Interfaces

You can take advantage of native voice user interfaces for smart home, music, video, and flash briefing skills, or define your own custom voice user interface.

Learn more: Custom interaction models

Support for Multiple Languages, Countries

You can create skills in multiple languages. A skill can support a single language or any combination of the available languages.

Learn more: Available languages

Multimodal Input and Response

You can use text-to-speech (TTS) and audio streams to respond to customer queries. You can add visual elements and touch inputs on Alexa-compatible devices with displays, and can add support for Echo Buttons.

Learn more: Multimodal experiences

In-Skill Purchasing and Amazon Pay

You can make money through Alexa skills using in-skill purchasing (ISP) or Amazon Pay for Alexa Skills. You define your product offering and price, and Amazon handles the voice purchasing flow.

Learn more: Make money with Alexa

Flexible Back End Support

You can write your back-end web service using any programming language and host it in either AWS Lambda or the HTTPS server of your choice. You can use ASK SDKs to get started quickly and take advantage of AWS promotional credits for Alexa.

Learn more: ASK SDKs and AWS credits

Fig 3.14: Alexa Skill kit Capabilities

3. AWS Lambda

Fig 3.15: AWS Lambda

What is AWS Lambda?

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running. With AWS Lambda, you can run code for virtually any type of application or backend service - all with zero administration. AWS Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. All you need to do is supply your code in one of the languages that AWS Lambda supports.

You can use AWS Lambda to run your code in response to events, such as changes to data in an Amazon S3 bucket or an Amazon DynamoDB table; to run your code in response to HTTP requests using Amazon API Gateway; or invoke your code using API calls made using AWS SDKs. With these capabilities, you can use Lambda to easily build data processing triggers for AWS services like Amazon S3 and Amazon DynamoDB, process streaming data stored in Kinesis, or create your own back end that operates at AWS scale, performance, and security.

You can also build serverless applications composed of functions that are triggered by events and automatically deploy them using CodePipeline and AWS CodeBuild. For more information, see AWS Lambda Applications.

For more information about the AWS Lambda execution environment, see AWS Lambda Runtimes. For information about how AWS Lambda determines compute resources required to execute your code, see AWS Lambda Function Configuration.

Reasons for choosing AWS Lambda

AWS Lambda is an ideal compute platform for many application scenarios, provided that you can write your application code in languages supported by AWS Lambda, and run within the AWS Lambda standard runtime environment and resources provided by Lambda.

When using AWS Lambda, you are responsible only for your code. AWS Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources. This is in exchange for flexibility, which means you cannot log in to compute instances, or customize the operating system or language runtime. These constraints enable AWS Lambda to perform operational and administrative activities on your behalf, including provisioning capacity, monitoring fleet health, applying security patches, deploying your code, and monitoring and logging your Lambda functions.

If you need to manage your own compute resources, Amazon Web Services also offers other compute services to meet your needs.

- □ Amazon Elastic Compute Cloud (Amazon EC2) service offers flexibility and a wide range of EC2 instance types to choose from. It gives you the option to customize operating systems, network and security settings, and the entire software stack, but you are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.
- □ Elastic Beanstalk offers an easy-to-use service for deploying and scaling applications onto Amazon EC2 in which you retain ownership and full control over the underlying EC2 instances.

CHAPTER 4

FLOWCHART

The flowchart of how a switch can be controlled is shown below in Fig 3.1

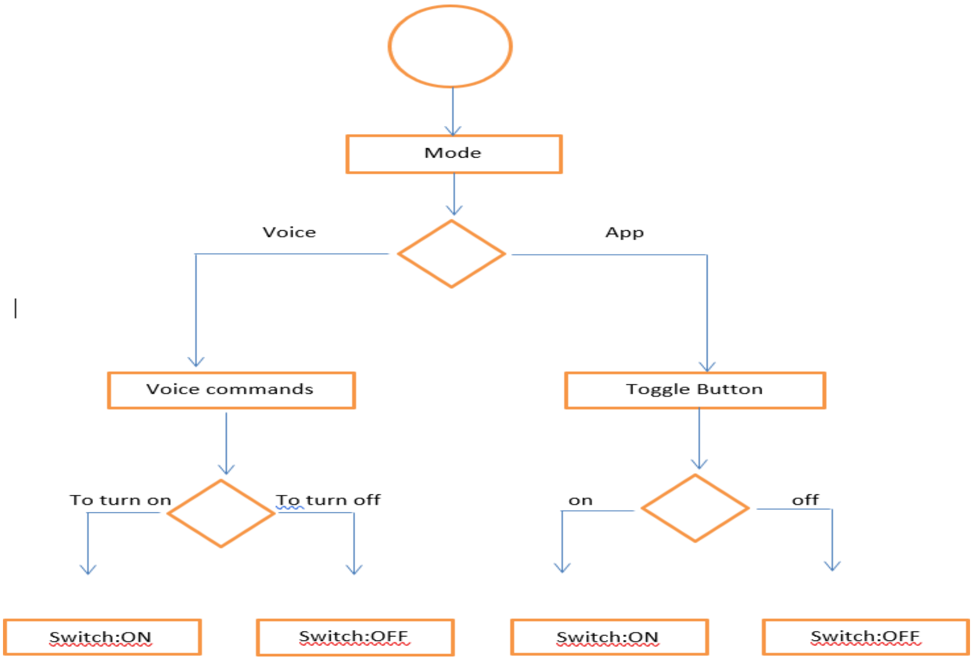


Fig. 4.1: Flow-chart of the methodology used

- □ The switch can be controlled in two ways i.e. mode. The two modes are voice control and mobile app.
- □ In voice control, the user gives voice commands to Alexa echo dot.
- If the voice command is to turn a switch on, it switches it on and if it's to turn off, its switches off
- □ In the mobile app, the interface has toggle button to set the switch state
- □ If the button is toggled to on, its switches on and if its toggled to off, it will switch off the switch
- □ This flowchart represents the methodology for a single switch. Same flow is present for all switches and can be replicated
- □ Apart from switching on and off, the switches can be scheduled to turn on for a specified time using the mobile app

CHAPTER 5

RESULTS AND DISCUSSIONS

The results or the outcome of the project work could be summarized based on the functionality achieved:

1. 8 channel Freedom Control

All the 8 channels in the board can be controlled individually or as a group for switching.

In the below demonstration we gave Alexa command "**TURN ON SWITCH 5**" and the Specific switch 5 was turned ON and the internal circuit picture after command execution of the board is attached below.

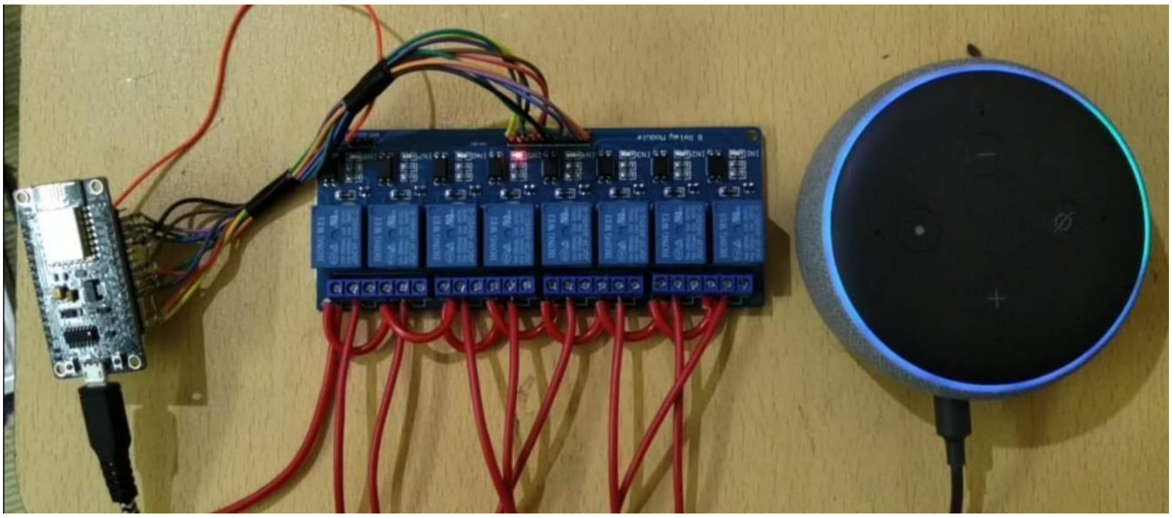
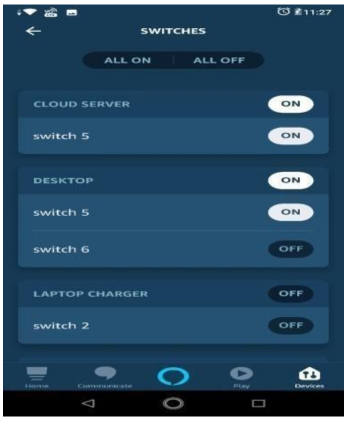


Fig 5.1: Switching Individual channel

2. Mobile App Control

The board can be controlled by a mobile app over the internet for remote switching on and off. As the figure below demonstrate the Interface of the Alexa app to access the board remotely.



CLOUD SERVER ON
switch 5 ON

DESKTOP ON
switch 5 ON
switch 6 OFF

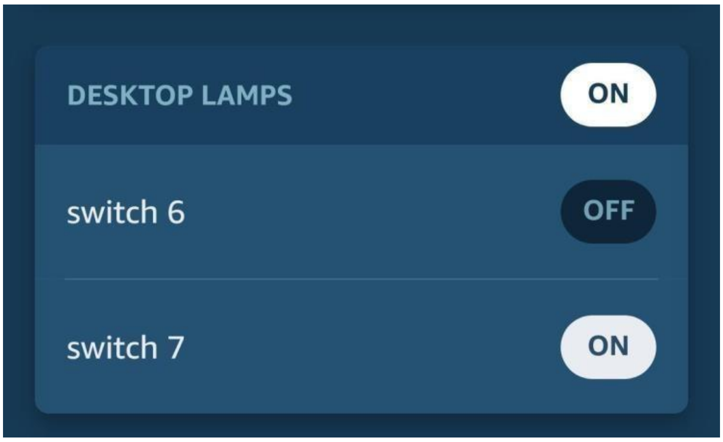
LAPTOP CHARGER OFF
switch 2 OFF

Fig 5.2: Mobile app

3. Group Control:

The switches can be controlled individually as well as in a group and defining the name of the groups based on user preference.

The below figure has a group Named as “DESKTOP LAMPS” which contains switch 6 and switch 7 and the Command to Alexa can be given as “TURN ON DESKTOP LAMPS ” to simultaneously access switch 6 and switch 7.



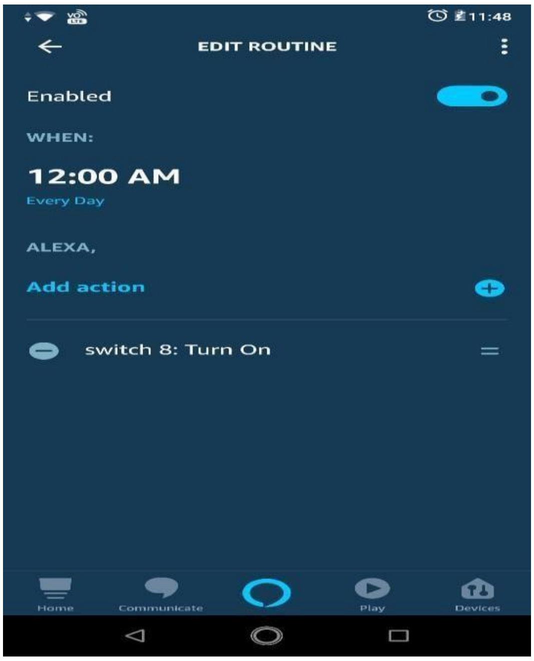
DESKTOP LAMPS ON
switch 6 OFF
switch 7 ON

Fig 5.3: DESKTOP LAMPS GROUP

4. Time scheduling:

Time scheduling to keep a switch ON at a specified time duration and can be managed easily and have full access to user to specify time and the interface for the timing specification is as shown in the figure below

The figure describes the action for the “SWITCH 8” to turn on at 12:00 AM as shown in the figure. This functionality can be can be used for any switch or the group created by the user.



Enabled

WHEN:

12:00 AM

Every Day

ALEXA,

switch 8: Turn On

Fig 5.4: Time scheduling for switch 8

CHAPTER 6

APPLICATIONS, ADVANTAGES AND LIMITATIONS

The project can be used for various application ranging from residential, office and industrial use. Voice controlling of switches can be a very large section of the society. Below are some of the applications

1. It can be used in sync with home automation as a house has many switches and plugs.
2. This project is a boon for visually impaired and physically challenged people, for whom it is inconvenient to perform the task of switching on or off of plugs.
3. It can be used to control switches in industrial units, where the switch boards are located at inaccessible locations. Since the project can control the switches remotely through voice and app, it can conveniently used
4. It can be used in offices where a supervisor is in charge of electrical switches. This project will give a centralized hub to control and monitor the switches in an institution
5. The scheduling option, which offers to switch on a plug for a specified time can prove to be very crucial for saving power consumption.
6. The scheduling option can also be used in scenarios with time constraints. For example, it can be used to turn on a computer being used for exam for a specified amount of time
7. The project makes it easier for monitoring and finding faults if any in the switch system of a place, particularly the ones with huge number of switches, since all the switches are either numbered or have names, which makes it easier to finding faults.

Advantages

1. The project can monitor the state of switches and with scheduling, it can be a precursor for reducing the power consumptions
2. The switches and groups can be named according to the user's choice.
3. It is easy to scale up the device since adding new board would only require relays and a microcontroller. The same app and alexa echo dot can handle the scaled-up versions
4. The system is easy to integrate since it's a stand-alone device and can be employed for home, office and industrial use
5. Compared to the products present in the market, this system is cheaper.

Limitations

1. The system is can be compromised by physical access or theft
2. The Alexa voice service is susceptible to relay attacks
3. The commands for voice control are only in English language, which can be a hassle for non-English speaking users.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

. The project aimed to design a simple and cost-effective switch board which could be controlled through voice as well as mobile app over the internet. This was achieved by using Alexa echo dot, which accepts voice command and by combining Alexa skills and nodeMCU, the objective was achieved. A system with 8 switches was designed. Each of the switch can be controlled individually through voice and app. Two or more switches can be grouped and named by the user to control them concurrently. Furthermore, the switches can be scheduled to turn on for a specific time period using the scheduling option. Overall, all the objectives of the system design were completed. The system designed is sixteen times cheaper than the present ones in the market. Hence, a cost-effective smart switch board was designed and successfully tested.

However, every system has flaws. No matter how hard we try to make the system robust, there are few shortcomings. The major drawback this system has security issues. If the voice commands are recorded and played back, the Alexa falls victim to replay attack. This can be improved in the future version with voice recognition and a more robust recognition. Also, as of now, Alexa echo dot support only English as a language to give commands in this system. Integration of vernacular languages to broaden the user base would be much needed upgrade in future.

REFERENCES

[1]. Vinay sagar KN & Kusuma SM , “Home automation using internet of things”, International Research Journal of Engineering Technology, 2015.

[2]. V. Këpuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 99-103.

[3]. A. M. D. Celebre, A. Z. D. Dubouzet, I. B. A. Medina, A. N. M. Surposa and R. C. Gustilo, "Home automation using raspberry Pi through Siri enabled mobile devices," *2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, Cebu City, 2015, pp. 1-6. doi: 10.1109/HNICEM.2015.7393270

[4] Alshu'Eili, H., Gupta, G. S., & Mukhopadhyay, S. (2011). Voice recognition based wireless home automation system. In *2011 4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11 - Conference Proceedings* Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE)

[5] A. A. Mohammed and E. Erçelebi, "Development of embedded system for making plugs smart," *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*, Ankara,2017pp.148-152.doi: 10.1109/ICEEE2.2017.7935810

[6]. Prof B.P Kulkarni, Aniket V Joshi, Vaibhav V Jadhav, Akshaykumar T Dhamange, “IoT Based Home Automation Using Raspberry PI”, International Journal of Innovative Studies in Sciences and Engineering Technology, Vol. 3 Issue 4 April 2017

[7].J.Chandramohan, R.Nagarajan, K.Satheeshkumar, N.Ajithkumar, P.A.Gopinath, S.Ranjithkumar6, “Intelligent Smart Home Automation and Security System Using Arduino and Wi-fi”, International Journal Of Engineering And Computer Science, Vol. 6 Issue 3 March 2017

[8]. Tatjana Eric, Sandra Ivanovic, Suncica Milivojsa, “Voice control for smart home automation: Evaluation of approaches and possible architectures”, Internet of Things Journal IEEE, Sept 2017

[9]. Kalyani Pampattiwar, Mit Lakhani, Rinisha Marar and Rhea Menon, “Home Automation using Raspberry Pi controlled via an Android Application”, International Journal of Current Engineering and Technology, 2017

[10]. Shivani V Devoor, Syed Sha Qutub & Mallikarjuna Swamy M.S ,"Design of Sensor Network for Home Automation System", International Journal of Engineering and Innovation technology,2015.

[11]. B. Selva Priya & R. Geethamani , " Design and Implementation of Home Automation using Power Electronic Switches", International Journal of Advanced Information and Communication Technology , 2016

[12]. Sirsath N. S, Dhole P. S, Mohire N. P, Naik S. C & Ratnaparkhi Vidyanagari "Home Automation using Cloud Network and Mobile Devices" ,The Institute for Research and Development India, vol.1,issue.2,2018.

[13]. Fei Ding, Aiguo Song, En Tong, and Jianqing Li,"A Smart Gateway Architecture for Improving Efficiency of Home Network Applications,", Journal of Sensors, vol. 2016.

[14]. Norhafizahbt Aripin, M. B. Othman, "Voice control of home appliances using Android", Electrical Power Electronics Communications Controls and Informatics Seminar (EECCIS) 2014, pp. 142-146, 2014.

[15]. N. Sriskanthan and Tan Karand. "Bluetooth Based Home Automation System",Journal of Microprocessors and Microsystems, Vol. 26, pp.281-289, 2002.

[16]. Arduino IDE- Home: https://www.arduino.cc/en/guide/environment

[17]. ESP8266 Community Forum: http://www.esp8266.com/

[18]. NodeMCU: https://www.electronicwings.com/nodemcu/introduction-to-nodemcu

[19]. 8-channel relay: https://www.makerlab-electronics.com/product/8-channel-5v-relay-module/

[20].Alexa skills:https://developer.amazon.com/blogs/post/Tx205N9U1UD338H/Introducing-the-Alexa-Skills-Kit-Enabling-Developers-to-Create-Entirely-New-Voic

[21]. AWS Lambda: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

[22]. Arshdeep Bahga,Vijay Madisetti, "Internet of Things (A Hands-on-Approach)", Publisher: Vijay Madisetti , 12th September 2014.

[23]. Kumar, M., and Shimi, S. L. "Voice Recognition Based Home Automation System of Paralyzed People System", Vol. 4, No. 10, 2015

APPENDIX

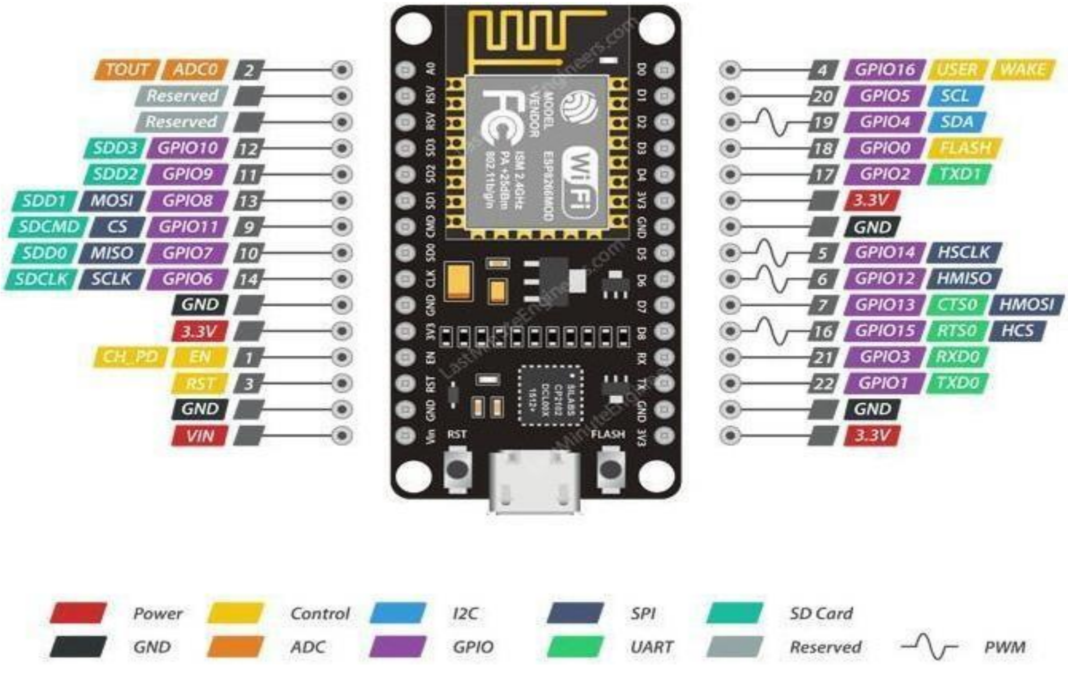


Fig 9.1: Pin diagram of NodeMCU

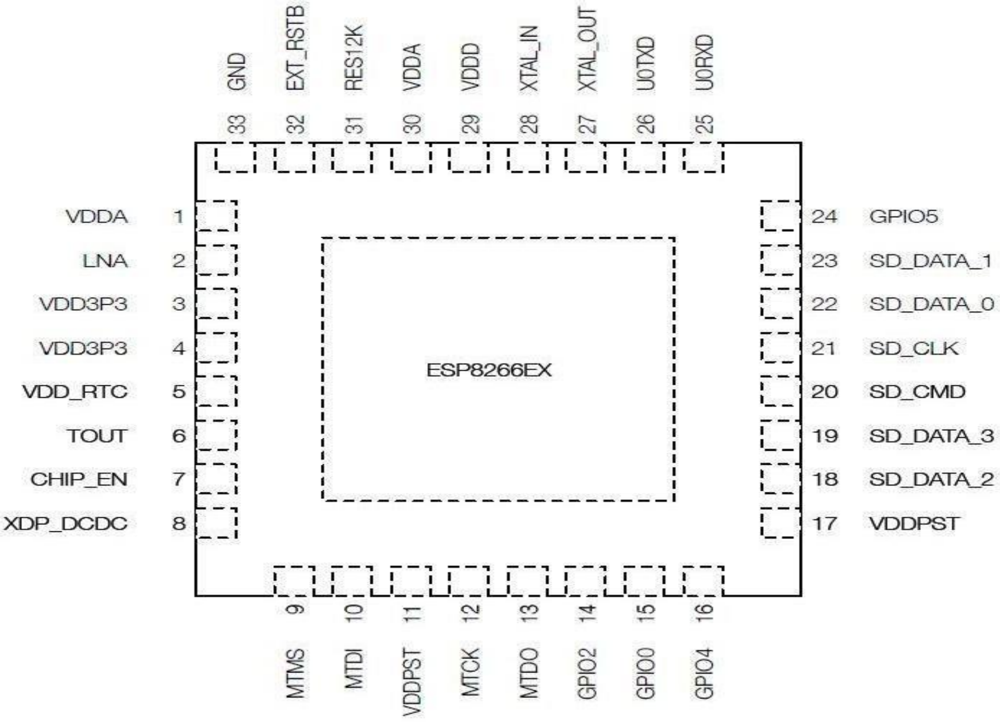


Fig 9.2: Pin layout (Top View) of NodeMCU

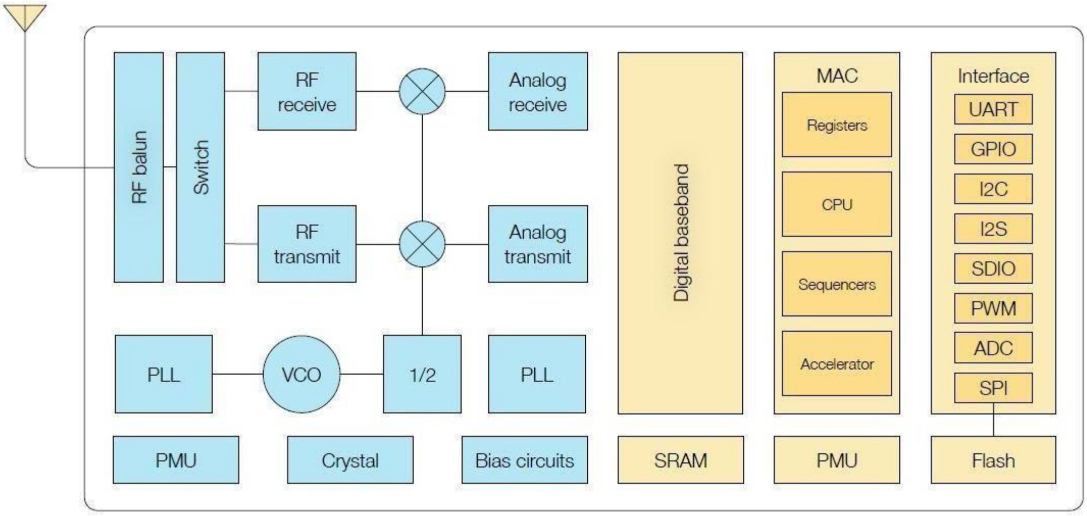


Fig 9.3: Functional Block Diagram of NodeMCU

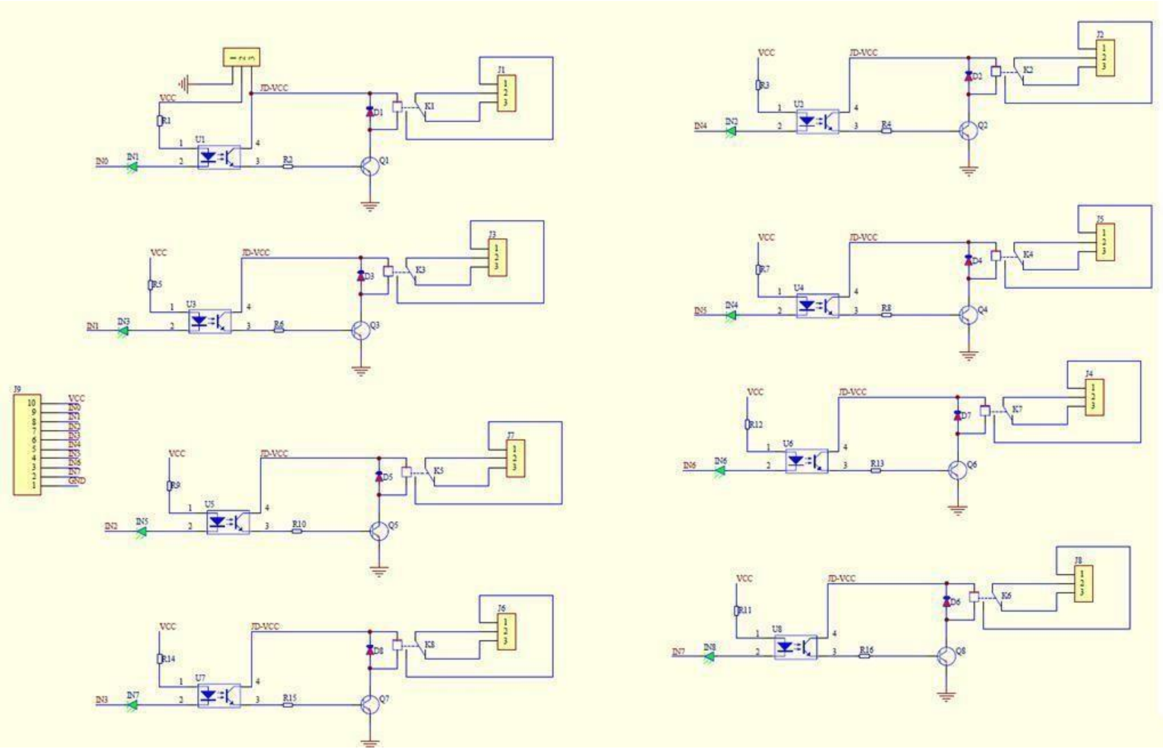


Fig 9.4: 8-channel relay module schematic

PROJECT PHOTOS