ABSTRACT

In this era of digitization and automation, Internet of things (IoT) provides a platform that allows devices to be connected, sensed and controlled remotely across a network infrastructure. As an organization grows every year, new management problems and energy issues appear. Monitoring and controlling the unused devices that consume power during human absence is a major inability. In addition to this, coordinating the people participating in the daily activities of the laboratory is tedious when population of the usage of space out numbers a manageable threshold.

This work targets to develop a smart laboratory system based on IoT and mobile application technologies to monitor the overall activities of the lab including energy consumption and utilization of devices, thereby providing a smart environment to the campus with enhanced security, energy efficiency and comfort. The aim is to control and monitor the things such as lights, fans, projector and air-conditioner of IoT lab in CIT campus using Google assistant or chat bot.

The results of implementation, it is observed that the appliances in the lab were remotely monitored and controlled thereby reducing power consumed and human energy considerably. In recent years, Internet of Things (IOT) is being playing a major role in automation of Industries. Smart home systems are being employed in buildings to improve energy efficiency and reduce power wastage. Voice recognition is the current trend in automation.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	**iv**
1	**INTRODUCTION**	**3**
	1.1MOTIVATION	
	1.2 OBJECTIVES	
	1.3 AREA OF PROJECT	
	1.4 SCOPE OF IOT	
	1.5 APPLICATION OF IOT	
	1.6 MACHINE LEARNING	
2	**DIAGRAMATIC VIEW**	**14**
	2.1 PRINCIPLE	
	2.2 WORKING	
	2.3 BLOCK DIAGRAM	
	2.4 HARDWARE DIAGRAM	
3	**COMPONENTS**	
	REQUIRED	**17**

3.1 COMPONENTS

DESCRIPTION

3.2 ESP 8266

3.3 RELAY

3.4 BULBS

4 SOFTWARE

REQUIRED 32

4.1 ARDUINO IDE

SOFTWARE

4.2 IFTTT ANDWEBHOOKS

4.3 BLYNK

PROJECT IMPLEMENTATION 41

HARDWARE OUTPUT 42

CONCLUSION 42

APPENDIX 40

CHAPTER -1

Introduction

The rapid development of information technology (IT) has brought forward a hyper connected society in which objects are connected to mobile devices and the Internet and communicate with one another. The core component of this hyper connected society is IoT, which is also referred to as Machine to Machine (M2M) communication. The Internet of Things represents a hallucination in which the Internet extends into real world hypothesis. Physical objects are no longer detached from the virtual world but can be controlled remotely.

Better outcomes can be obtained if several elements can communicate with each other and provide the user a unified response. The network aspects are bringing online streaming services or network playback, while becoming a means to control the device functionality over the network. At the same time mobile devices ensure that customer has access to control the electronic devices connected to network.

The aim of the proposed system is to design a smart laboratory based on IOT and mobile application technologies that monitors the overall activities of the lab including energy consumption, human presence, devices and their usage via sensors and provides user understandable data through hand held devices about the status of the laboratory. Besides, it also provides a smart environment to CIT to help improve campus management in the Internet era with enhanced security, energy efficiency and comfort to the end user.

As campus grows every year, new management problems and energy issues appear. Managing the resources in the campus has become a real problem. Monitoring and controlling the unused devices that consume power during human absence is also a major inability.

1.1 Motivation

Home computerization brings about a more astute home and is utilized to give a higher and more beneficial way of life. The magnificence of a home computerization framework is that it is very versatile, adaptable and its abilities are constrained just by our creative ability. With the IOT unrest practically around the bend, it's about time that we move towards boundless selection of such a prototype.

1.2 objectives

The principle goal of this venture is to outline and execute a modest and open source home computerization framework that is equipped for controlling and mechanizing a large portion of the house machines. This application is a simple and reasonable web interface for client to run Home Mechanization Framework.

In this venture we have coordinated advances like Arduino with Wi-Fi to execute Home Mechanization Framework. The plan to take Wi-Fi as platform to communicate with all home gadgets in place of manual interference and delay. In this application, we utilized fans, bulbs and so on delineated Graphical User.

1.3 AREA OF PROJECT

Internet of things (IOT)

The Internet of things (IoT) describes physical objects (or groups of such objects) that are embedded with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

Internet of Things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things.

In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systemsand cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as Smartphone’s and smart speakers.

.

Internet of Things

The Internet of things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

IOT DEVICES

As we said earlier, there are many scenarios in which IoT can be employed and they all require different devices. Here, at the most basic level, we can speak of sensors (i.e. devices that sense things, such as temperature, motion, particles, etc.) and actuators (i.e. devices that act on things, such as switches or rotors). Rarely, though, will a smart solutionmake do with just one type of an IoT sensor or an actuator.

Consider running a smart farm for a plant to grow, its not just a matter of measuring the humidity of the soil, but also its fertility; it's also a matter of providing proper irrigation based on insulation, and much more. So, you need not just one, but many sensors and actuators that all have to work together. When speaking of devices essential for the IoT ecosystem, one cannot forget about IoT gateways. They are a piece of hardware that is capable of "translating" and facilitating the essential connection between devices or between devices and the network and work as a kind of relay for the two. Whichbrings us to the next element of our puzzle?

Whether they are in the cloud or not, IoT platforms are always thebinder for any IoT ecosystem. They are the quiet administrators that take care of device lifecycle management, so that you don't have to worry about them. They are also the hub that collects and aggregates the data, allowing you to make sense of it. With the variety of platforms offered on the market and the breadth of claims their providers make, the choice of the "ideal" IoT platform for a deployment is arguably the most significant, yet also the most difficult to make. It shouldn't be taken lightly, as it determines whether the IoT ecosystem will thrive or wither into oblivion.

The Internet of Things (IoT) is creating much buzz while it goes about transforming our lives.

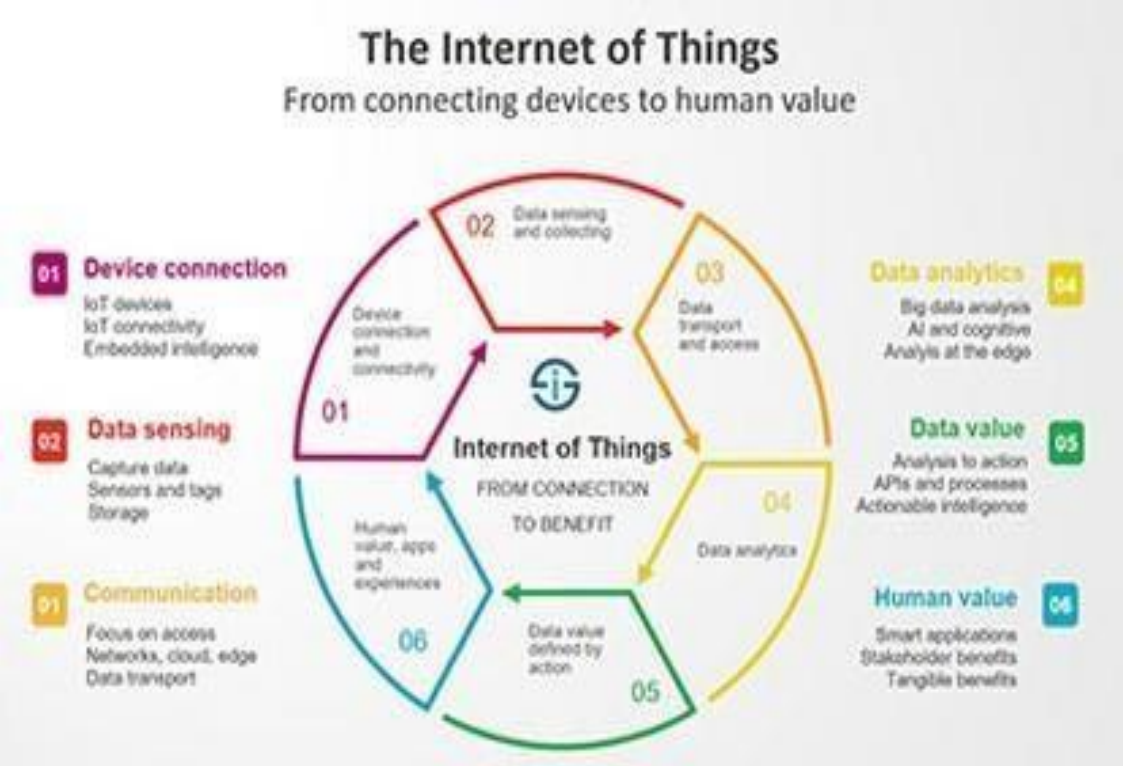
1.4 SCOPE OF IOT

Internet of Things has emerged as a leading technology around theworld. It has gained a lot of popularity in lesser time. Also, the advancements in Artificial Intelligence and Machine Learning have made the automation of IoT devices easy. Basically, AI and ML programs are combined with IoT devices to

give them proper automation. Due to this, IoT has also expanded its area of application in various sectors.

Here, in this section, we will discuss the applications and the future scope of IoT in healthcare, automotive, and agriculture industries. IoT has proved to be one of the best tools for the healthcare industry.

It helps provide advanced healthcare facilities to patients, doctors, and researchers. These facilities include smart diagnosis, wearable devices for tracking health, patient management, and many more. Furthermore, IoT devices have reduced unnecessary strain on the healthcare system .



Internet of Things from connection to benefit

1.5 APPLICATIONS OF IOT

- Smart City
- Self-driven Cars
- IOT Retail Shops

- Farming
- Wearable's
- Smart Grids
- Industrial Internet

1.6 MACHINE LEARNING

Machine Learning (ML) has proven to be one of the most game-changing technological advancements of the past decade. In the increasingly competitive corporate world, ML is enabling companies to fast-track digital transformation and move into an age of automation. Some might even argue that AI/ML is required to stay relevant in some verticals, such as digital payments and fraud detection in banking or product recommendations.

The eventual adoption of machine learning algorithms and its pervasiveness in enterprises is also well-documented, with different companies adopting machine learning at scale across verticals. Today, every other app and software all over the Internet uses machine learning in some form or the other. Machine Learning has become so pervasive that it has now become the go-to way for companies to solve a bevy of problems. Well dive deeper into what machine learning is, the basics of ML, types of machine learning algorithms, and a few examples of machine learning in action. We will also take a look at the difference between artificial intelligence and machine learning.

With machine learning algorithms, AI was able to develop beyondjust performing the tasks it was programmed to do. Before ML entered the mainstream, AI programs were only used to automate low-level tasks in business and enterprise settings. This included tasks like intelligent automation or simple rule-based classification. This meant that AI algorithms were restricted to only the domain of what they were processed for. However, with machine learning, computers were able to move past doing what they were

programmed and began evolving with each iteration. Machine learning is fundamentally set apart from artificial intelligence, as it has the capability to evolve.

Using various programming techniques, machine learning algorithms are able to process large amounts of data and extract useful information. In this way, they can improve upon their previous iterations by learning from the data they are provided. We cannot talk about machine learning without speaking about big data, one of the most important aspects of machine learning algorithms.

Any type of AI is usually dependent on the quality of its dataset for good results, as the field makes use of statistical methods heavily. Machine learning is no exception, and a good flow of organized, varied data is required for a robust ML solution. In today's online-first world, companies have accessto a large amount of data about their customers, usually in the millions. This data, which is both large in the number of data points and the number of fields, is known as big data due to the sheer amount of information it holds.

Big data is time-consuming and difficult to process by human standards, but good quality data is the best fodder to train a machine learning algorithm. The more clean, usable, and machine-readable data there is in a big dataset, the more effective the training of the machine learning algorithm will be. As explained, machine learning algorithms have the ability to improve themselves through training. Today, ML algorithms are trained using three prominent methods. These are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

CHAPTER -2

DIAGRAMATIC VIEW

2.1 principle

A. The whole apparatus connected to hand-off are controlled by Microcontroller.

B. ESP8266 is a Wi-Fi module interface to Microcontroller for remote correspondence.

C. We make a web server utilizing ESP8266 and screen all progressions using controlling transfers.

D. There is a site page which can be access through any gadget associated with home Wi-Fi. we can switch it by ON/OFF with that page can be access by putting IP address of ESP8266.

When we turn ON or OFF the catch from web server Wi-Fi switch offers summon to Wi-Fi module and after that it offers charge to Microcontroller

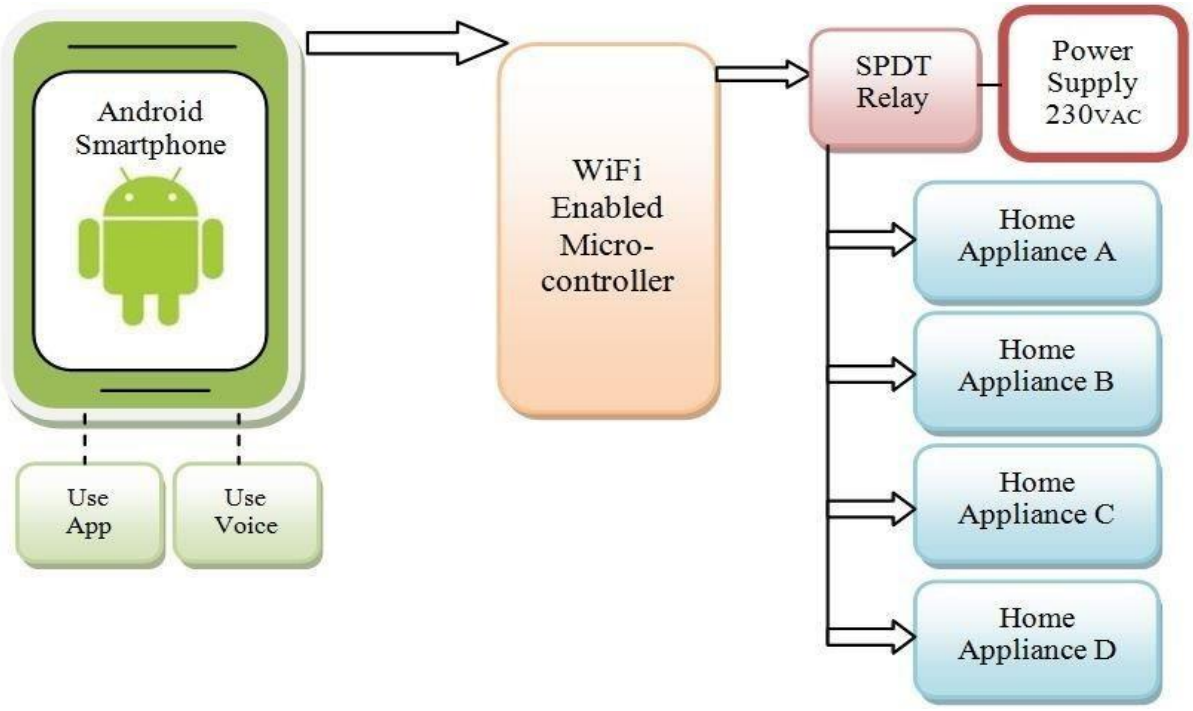
2.2 working

The home automation circuit is built around ESP8266, Blynk Android App, and a 4-channel relay board. The hardware set up should be according to the circuit diagram. AC mains appliances (Bulbs) will be connected to relays which are controlled by the ESP8266.

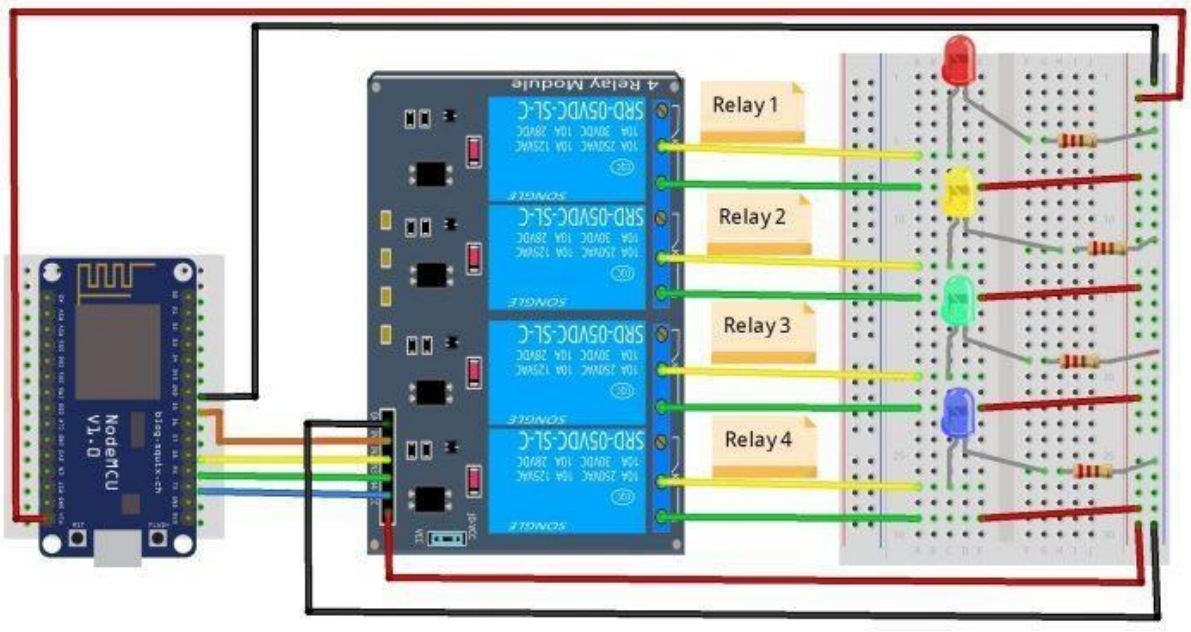
User has to install and configure the Blynk App and Alexa app as per the above instructions.

Whenever the user presses an icon in the app, then that information will be sent to ESP8266 via Wi-Fi. The ESP8266 analyses the received commands and turns ON/OFF of the respective device via 4-channel Relay board.

2.3 block diagram



2.1 Hardware Diagram



CHAPTER -3

3.Components required

SL.NO.	NAME OF THE COMPONENT	QUANTITY
1	ESP8266 (wifi module)	1
2	RELAY (4 channel)	1
3	BULB	3
4	COOLING FAN (12 volt)	1
5	POWER SUPPLY	-
6	5v ADAPTER	1

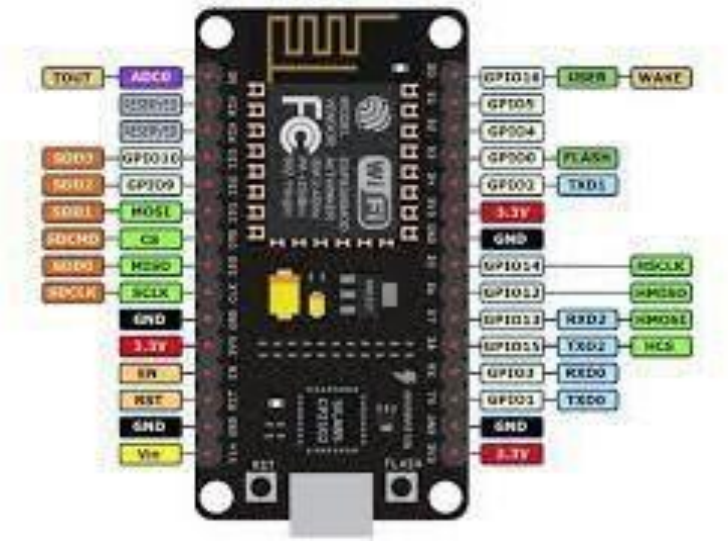
3.1. COMPONENTS DESCRIPTION:

3.2 ESP 8266

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi ability as a Wi-Fi Shield offers (and that's just out of

the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community



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

Power

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions.

The output of the regulator is also broken out to one of the sides of the board and labelled as 3V3. This pin can be used to supply power to external components

VIN- The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply powerto external components.

5V- The IO pins in input state (sink) are 5V tolerant, Yet the power supply to the chip must be 3.3V (Most boards come with a regulator for this soit should not be a problem). other models do not come with a regulator, and in such a case, you will need to add the regulator, but even then, you do not need a level shifter for the digital inputs

MEMORY

Processor: L106 32-bit RISC microprocessor core based on the Ten silica Diamond Standard 106Micro running at 80 MHz [5]

Memory: [citation needed]

32 KiB instruction RAM

32 KiB instruction cache RAM

80 KiB user-data RAM

16 KiB ETS system-data RAM

External QSPI flash: up to 16 MI is supported (512 KiB to 4 MI typically included)

INPUT AND OUTPUT

Processor: L106 32-bit RISC microprocessor core based on the Tensilica Diamond Standard 106Micro running at 80 MHz[5]

Memory:[citation needed]

32 KiB instruction RAM

32 KiB instruction cache RAM

80 KiB user-data RAM

16 KiB ETS system-data RAM

External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)

Just like a normal Arduino, the ESP8266 has digital input/output pins (I/O or GPIO, General Purpose Input/outputpins). As the name implies, they can be used as digital inputs to read a digital voltage, or as digital outputs to output either 0V (sink current) or 3.3V (source current).

Voltage and current restrictions

The ESP8266 is a 3.3V microcontroller, so its I/O operates at 3.3V as well. The pins are not 5V tolerant, applying more than 3.6V on any pin will kill the chip.

The maximum current that can be drawn from a single GPIO pin is 12mA.S

Usable pins

The ESP8266 has 17 GPIO pins (0-16), however, you can only use 11of them, because 6 pins (GPIO 6 - 11) are used to connect the flash memory chip. This is the small 8-legged chip right next to the ESP8266. If you try to use one of these pins, you might crash your program.

GPIO 1 and 3 are used as TX and RX of the hardware Serial port (UART), so in most cases, you can't use them as normal I/O while sending/receiving serial data.

Boot modes

As mentioned in the previous chapter, some I/O pins have a special function during boot: They select 1 of 3 boot modes:

GPIO15	GPIO0	GPIO2	Mode
0V	0V	3.3V	Uart Bootloader
0V	3.3V	3.3V	Boot sketch (SPI flash)
3.3V	x	x	SDIO mode (not used for Arduino)

Note: you don't have to add an external pull-up resistor to GPIO2, the internal one is enabled at boot.

We made sure that these conditions are met by adding external resistorsin the previous chapter, or the board manufacturer of your board added them for you. This has some implications, however:

GPIO15 is always pulled low, so you can't use the internal pull-up resistor. You have to keep this in mind when using GPIO15 as an input to read aswitch or connect it to a device with an open-collector (or open-drain) output, like I²C.GPIO0 is pulled high during normal operation, so you can't use it as a Hi-Z input.

GPIO2 can't be low at boot, so you can't connect a switch to it.

Internal pull-up/-down resistors

GPIO 0-15 all have a built-in pull-up resistor, just like in an Arduino. GPIO16 has a built-in pull-down resistor.

PWM

Unlike most Atmel chips (Arduino), the ESP8266 doesn't support hardware PWM, however, software PWM is supported on all digital pins. The default PWM range is 10-bits @ 1kHz, but this can be changed (up to >14-bit@1kHz).

Analog input

The ESP8266 has a single analog input, with an input range of 0 - 1.0V. If you supply 3.3V, for example, you will damage the chip. Some boards

like the NodeMCU have an on-board resistive voltage divider, to get an easier 0 - 3.3V range. You could also just use a trimpot as a voltage divider.

The ADC (analog to digital converter) has a resolution of 10 bits.

Communication

Serial

The ESP8266 has two hardware UARTS (Serial ports):

UART0 on pins 1 and 3 (TX0 and RX0 resp.), and UART1 on pins 2 and 8 (TX1 and RX1 resp.), however, GPIO8 is used to connect the flash chip. This means that UART1 can only transmit data.

UART0 also has hardware flow control on pins 15 and 13 (RTS0 and CTS0 resp.). These two pins can also be used as alternative TX0 and RX0 pins.

I²C

The ESP doesn't have a hardware TWI (Two Wire Interface), but it is implemented in software. This means that you can use pretty much any two digital pins. By default, the I²C library uses pin 4 as SDA and pin 5 as SCL. (The data sheet specifies GPIO2 as SDA and GPIO14 as SCL.) The maximum speed is approximately 450kHz.

SPI

The ESP8266 has one SPI connection available to the user, referred to as HSPI. It uses GPIO14 as CLK, 12 as MISO, 13 as MOSI and 15 as Slave Select (SS). It can be used in both Slave and Master mode (in software).

GPIO overview

GPIO	Function	State	Restrictions
0	Boot mode select	3.3V	No Hi-Z
1	TX0	-	Not usable during Serial transmission
2	Boot mode select		
TX1	3.3V (boot only)		Don't connect to ground at boot time
Sends debug data at boot time			
3	RX0	-	Not usable during Serial transmission
4	SDA (I²C)	-	-
5	SCL (I²C)	-	-
6 - 11	Flash connection	x	Not usable, and not broken out
12	MISO (SPI)	-	-
13	MOSI (SPI)	-	-
14	SCK (SPI)	-	-
15	SS (SPI)	0V	Pull-up resistor not usable
16	Wake up from sleep	-	No pull-up resistor, but pull-down instead
Should be connected to RST to wake up			

The ESP8266 as a microcontroller - Software

Most of the microcontroller functionality of the ESP uses exactly the samesyntax as a normal Arduino, making it really easy to get started

.

Digital I/O

Just like with a regular Arduino, you can set the function of a pin using pinMode(pin, mode); where pin is the GPIO number*, and mode can be either INPUT, which is the default, OUTPUT, or INPUT_PULLUP to enable the built-in pull-up resistors for GPIO 0-15. To enable the pull-down resistor for GPIO16, you have to use INPUT_PULLDOWN_16.

(*) NodeMCU uses a different pin mapping, read more here. To address a NodeMCU pin, e.g. pin 5, use D5: for instance: pinMode(D5, OUTPUT);

To set an output pin high (3.3V) or low (0V), use digitalWrite(pin, value); where pin is the digital pin, and value either 1 or 0 (or HIGH and LOW).

To read an input, use digitalRead(pin);

To enable PWM on a certain pin, use analogWrite(pin, value); wherepin is the digital pin, and value a number between 0 and 1023.

You can change the range (bit depth) of the PWM output by using analog WriteRange(new_range);

The frequency can be changed by using analog Writ eFreq(new _frequency);. New _frequency should be between 100 and 1000Hz.

Analog input

Just like on an Arduino, you can use analogRead(A0) to get the analog voltage on the analog input. (0 = 0V, 1023 = 1.0V).

The ESP can also use the ADC to measure the supply voltage (VCC).To do this, include ADC_MODE(ADC_VCC); at the top of your sketch, and use ESP.getVcc(); to actually get the voltage.

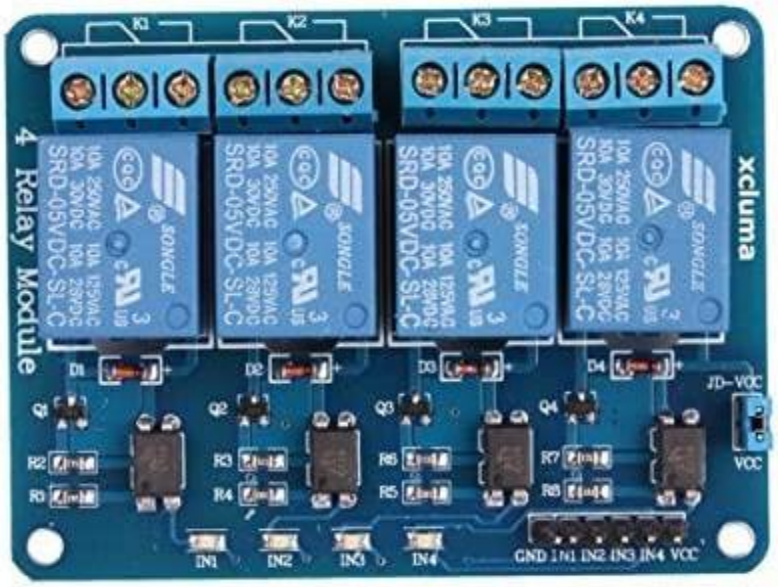
If you use it to read the supply voltage, you can't connect anything elseto the analog pin.

3.3 RELAY (4 channel)

This relay module is 5V active low.Low Active Means Relay will Get Trigger when Low Voltage/Signal Supplied to IN Pin. This Is A 5v 4-Channel Relay Interface Board,Be Able To Control Various Appliances, And Other Equipment's With Large Current.It Can Be Controlled Directly By Microcontroller(Arduino , 8051, Avr, Pic, Dsp, Arm, Arm, Msp430, Title Logic) .

5v 4-Channel Relay Interface Board, And Each One Needs 15-20ma DrivercurrentEquiped With High-Current Relay, Ac250v 10a ; Dc30v 10a The 8550 Transistor Drive, Drive Ability Working Voltage 5 V

Has The Fixed Bolt Hole And Easy Installation Small Board Pcb Size: 7.2 Cm * 4.8 Cm Power Indicator (Green), Two Ways Of Relay Status Indicator Light (Red) Standard Interface That Can Be Controlled Directly By Microcontroller (Avr, Pic, Dsp, Arm, Arm, Msp430, Title Logic) Indication Led'S For Relay Output Status.



APPLICATION

- Robotics
- Electronics projects
- Industrial controls
- Microwaves Oven
- Fans, DC Motor
- AC Lamp
- Solenoids Remote Controls

3.4 BULBS

A smart bulb is an internet-capable LED light bulb that allows lighting to be customized, scheduled and controlled remotely. Smart bulbs are among the most immediately successful offerings in the growing category of home automation and Internet of Things (IoT) products.

With the integration of Wi-Fi, Bluetooth, ZigBee or a proprietary connection for a home automation systems, smart bulbs can be controlled through a mobile app or a home/building automation hub and individual bulbs can be programmed to change output in a specific manner. The bulb's internet connectivity makes it possible for vendors to use edge computing and equip smart bulbs with additional features like built-in cameras, built-in speakers and presence-sensing capability. Many types of smart bulbs enable the home or building manager to control brightness as well as RBG colour.

As with all IoT products, smart bulbs can become an attack vector. This was demonstrated when researchers from security consultancy Context were able to gain a network's password at a 30-meter distance from the targeted smart bulb. As the researchers explained, "Armed with knowledge of the encryption algorithm, key, initialization vector and an understanding of the mesh network protocol we could then inject packets into the mesh network, capture the Wi-Fi details, and decrypt the credentials, all without any prior authentication or alerting of our presence."

CHAPTER -4

SOFTWARE REQURIED

4.1 ARDUINO IDE SOFTWARE

Programs written using Arduino Software (IDE) are called sketches. These sketchesare written in the text editor and are saved with the file extension. into the editor hasfeatures for cutting/pasting and for searching/replacing text.

The message area givesfeedback while saving and exporting and also displays error.

The console displaystext output by the Arduino Software (IDE), including complete error messages andother information.

The bottom righthand corner of the window displays the configuredboard and serial port..

The toolbar buttons allow you to verify and upload programs,create, open, and save sketches, and open the serial monitor.

4.1 IFTTT AND WEBHOOKS

IFTTT is the free way to get all your apps and devices talking to each other.Not everything on the internet plays nice.

IFTTT is a web-based service that allows various platforms, apps and gadgets to trigger responses in one another that otherwise wouldn't be possible.

Automate tasks that might otherwise be repetitive or unable to talk to each other. It works like this: users are guided through a process to make simple scripts, aka "applet," where some type of event in one device or service automatically triggers an action in another.

IFTTT is also completely free, and well supported. There are now more than 300 channels — which are what you reference when creating applets — spread across a range of devices and services, including social networks, smart appliances, smart home systems, and devices such as weather stations, audio systems, and wearables.

Categories of IFTTT Home Automation

IFTTT is divided into Services and Applets.

Services

Services aren't just services like Spotify, they also cover apps, whole platforms and devices. Basically anything you might want to control or be controlled by something else.

IFTTT Services include appliances, baby, communication, business tools, cloud storage, environment control, security and monitoring, lighting, photo and video and shopping.

Applets (Formerly known by Recipe**)**

Applets are composed of triggers and actions.

IFTTT stands for "If This Then That," where "This" represents a primary app and "That" represents a secondary connected app. When an **action** takes place on the primary app, the secondary connected app is automatically **triggered** to also take some sort of action.

Triggers

Triggers tell an Applet to start.

Actions

Amazon Alexa Actions are the end result of an Applet run

IFTTT is pretty damn comprehensive when it comes to smart home tech and the services you might want to use with it. There's lots of compatibility with smartphone-based triggers too.

Voice Assistant

- Google Assistant
- Microsoft Cortana
- InvoxiaTriby
- Jibo
- Samsung SmartThings

- Wink
- Figaro

Note: Siri isn't officially listed but you can set up Applets using Siri – Apple's own IFTTT-like feature is called Siri Shortcuts

Music

- Spotify
- Soundcloud
- Deezer
- Songkick
- Linn
- Songs
- TV and video
- TiVo
- AnyMote Smart Remote
- Comcast Labs
- YouTube
- Vimeo

Smart appliances

- iRobot
- Neato
- GE Appliances
- LG Washer, Dryer
- I Devices plugs and switches
- Samsung Washer, Samsung Robot Vacuum, Samsung Refrigerator
- Whirlpool Dryer, Whirlpool Washer, Whirlpool Refrigerator
- Rachio

Does IFTTT COST ?

The free service lets you run 750 flows per month and create unlimited flows. More advanced functionality costs $5 or $15 per user per month.

How to use IFTTT Applets?

IFTTT has a vast library of existing applets created by other users you can use with your own apps. Alternatively, you can create your own applet fromscratch.

3.1 BLYNK

Hey geeks, Welcome back to another new post. Today we gonna makehome automation using Blynk App and NodeMCU IoT development board. We can control the home appliances via the Blynk application. There is wireless communication between the Blynk app and the nodemcu.

There are some relay modules that we used in our project to switch on and off the devices. For more information regarding this project please visit the original post of this project and also bookmark TECHATRONIC.COM as allmy further projects and tutorials will be pre-uploaded there.

Working of the project

You have to provide the SSID and password of the router or a hotspot in the code below. Then set up the Blynk app, we have discussed the procedure in detail below. There are pushbuttons for the appliances that we have to control so you can switch them manually too. \

When we tap the buttons on the Blynk app then the relevant command is sent to the nodemcu board and it will further process it and control the appliances. There is a set of predefined commands which we give to the nodemcu so that it can generate the output accordingly

Project implementation

CONCLUSION

The IoT device market has undergone radical changes in only a few short years. Starting with disparate devices and no ecosystems to speak of, the market has now grown to encompass enterprise players working together to create ecosystems, tailored for mobile technology, which allows IoT devices to become interconnected.

Automaton of the home may have once seemed like a peculiar and unlikely concept, but as our devices become smarter and more investment is poured into the development of ioT consumer products, we are likely to see increased competition spur on further innovation in the field.

Appendix

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "vrM_Fq2hN8YL4DNl6rNmx95fomEGOqAu";//Enter your Auth token
charssid[] = "dheena";//Enter your WIFI name
char pass[] = "dheena12343";//Enter your WIFI password
void setup()
{
  // Debug console
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}
void loop()
{
Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
                                        // to avoid \delay() function!
```

REFERENCE

[1] https://www.toptal.com/designers/interactive/smart-home-domestic-internet-of-things#:~:text=IoT%20home%20automation%20is%20the,controlled%20by%20a%20mobile%20app.

[2] https://www.factoryforward.com/iot-home-automation-using-blynk-nodemcu/

[3] https://iotcircuithub.com/nodemcu-esp8266-blynk-home-automation/

[4] Dinesh Goyal, Dharm Singh Jat, S. Balamurugan, Sheng Lung Peng, 28 June 2019

[5] Ajit Singh, Smart Home Automation Using IoT, 20 June 2019

[6] Gerard O'Driscoll, Smart Home Automation Essential Guides - the Complete Series, 31 January 2017

[7] Dwight Spivey, Home Automation For Dummies, 9 February 2015