



Devang Patel Institute of Advance Technology and Research

(A Constitute Institute of CHARUSAT)

Certificate

This is to certify that

Mr. / Mrs. Het Bipinkumar Patel

of Depstar - CE Class,

ID. No. 24DCCE091 has satisfactorily completed

his/ her term work in ITUS - 101 for

the ending in Dec 2025 / 2026

Date : 28/11/2026

Sign. of Faculty

Head of Department



Devang Patel Institute of Advance Technology and Research

(A Constitute Institute of CHARUSAT)

Name: Patel Het Bipinkumar

ID. No. 24DCIE091

INDEX

No.	DATE	SUBJECT	PAGE	MARKS	SIGNATURE
1)		Introduction to computer hardware types of memory / types of motherboard / types of processor			
2)		Assembly of computer			
3)		Installation of Linux and windows using Vmware			
4)		Learning Linux and windows command			
5)		mooc on google workshop			
6)		Workshop in github			
7)		playing with computer Network			
8)		Install the Arduino IDE - connect the led and buzzer to arduino uno			
9)		Develop Arduino IVC sketch for soil moisture sensor & DHT11			
10)		Enable Arduino IOT, connect to Arduino Uno / ESP32. Display IDE and upload			
11)		connect water level sensor to Arduino Uno / ESP32. Display . Pro			
12)		Develop Arduino UNO C sketch for ultra sonic sensor & buzzer in flame sensor and buzzer			
13)		develop Arduino C sketch to LED on and off from Arduino cloud and monitor DHT11 sensor			
14)		create a powerpoint presentation using AI Technology			

Experiment No 1.

AIM: Introduction to computer Hardware, Types of Memory, Types of Motherboard, Types of Processors.

Objectives: To Know About Different Components of Computer Hardware.

1. Components : CPU:

	A CPU or Processor, is a circuit board inside a computer that executes instructions on behalf of programs. Modern computer processors can process millions of instructions in a second. Processors are considered the main chip on a computer.
---	--

2. Memory (RAM):

	RAM is fast, temporary computer memory that helps the processor work efficiently by storing active data for quick access. It's crucial for multitasking and running applications, but it's wiped when the computer turns off.
---	---

3. Storage:

	Storage refers to a computer's long-term memory where data, files, and programs are stored even when the computer is powered off. It's used for keeping information over time, such as documents, photos, and software. Unlike RAM, storage retains data permanently until intentionally deleted or modified.
---	---

4. Input Devices:

Sr. No.	Image	Description
1.	 MOUSE	A mouse is a hand-held input device for computers that allows users to control the cursor on the screen by moving the device across a surface. It typically has buttons that are used for clicking and interacting with elements on the screen. The mouse is an essential tool for navigating and interacting with graphical user interfaces and software.
2.	 KEYBOARD	A keyboard is an input device for computers consisting of a set of keys that are used to input text and commands. Each key represents a specific letter, number, symbol, or function. Users press the keys to communicate with the computer, type text, enter commands, and interact with software applications.
3.	 JOYSTICK	A joystick is a manual input device used for controlling motion on a computer or a gaming system. It typically consists of a lever-like stick that can be moved in different directions and buttons for additional actions. Joysticks are commonly used in gaming and simulations to provide precise control over movement and actions within a digital environment.

4.	 MONITOR	A monitor is a display screen for computers and other devices, showing visual output such as images, text, and videos. It allows users to interact with digital content and view the results of their actions. Monitors come in various sizes and resolutions, and they play a crucial role in providing visual feedback and enabling user interaction with computers.
----	---	--

Conclusion: we get to know about computer Hardware, Types of Memory, Types of Motherboard, Types of Processors.

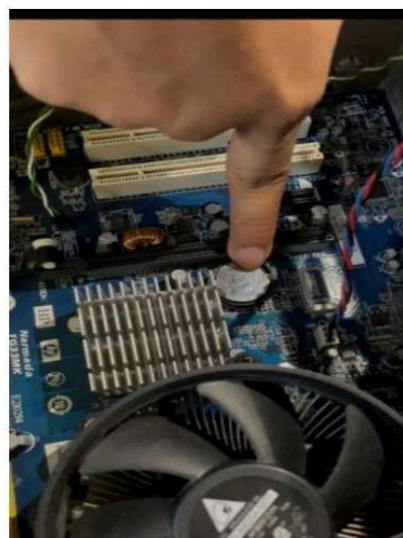
PRACTICLE 2

AIM: **Assembling of computer.**

DRIVE LINK OF VIDEO:

https://drive.google.com/file/d/1lCe6TkkKLh8XaZR_oGUubNfjLf1-n-y/view?usp=drive_link

SCREENSHOTS OF IMPLEMENTATION:



OBSERVATIONS: Motherboard is securely fastened to the case. Storage drives are connected to the motherboard and power supply. RAM modules are installed in the correct slots. CPU cooler is installed correctly. I have assembled the computer's CPU perfectly.

CONCLUSION: Assembling a computer is basic understanding of computer hardware components. By following step-by-step approach, any

individual can successfully build a custom computer that meets their specific needs.

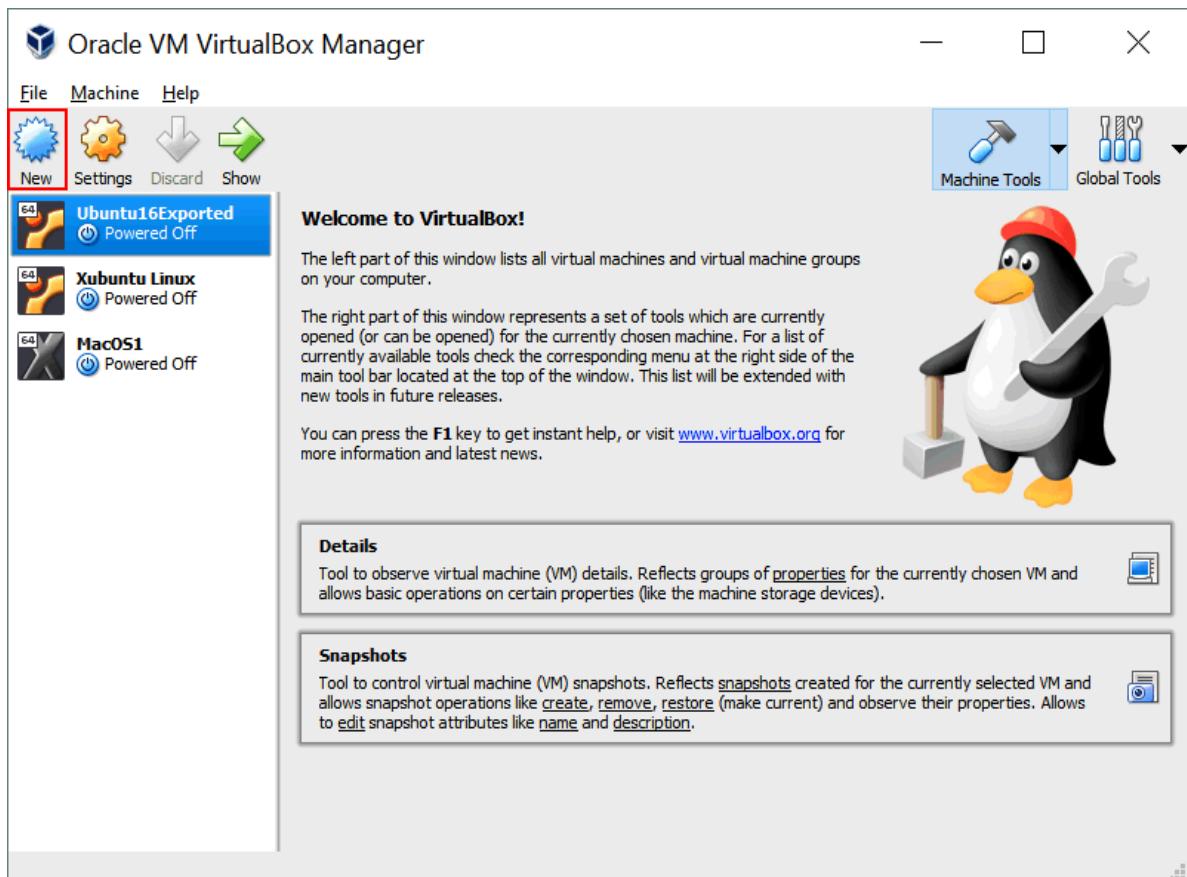
Experiment No 3.

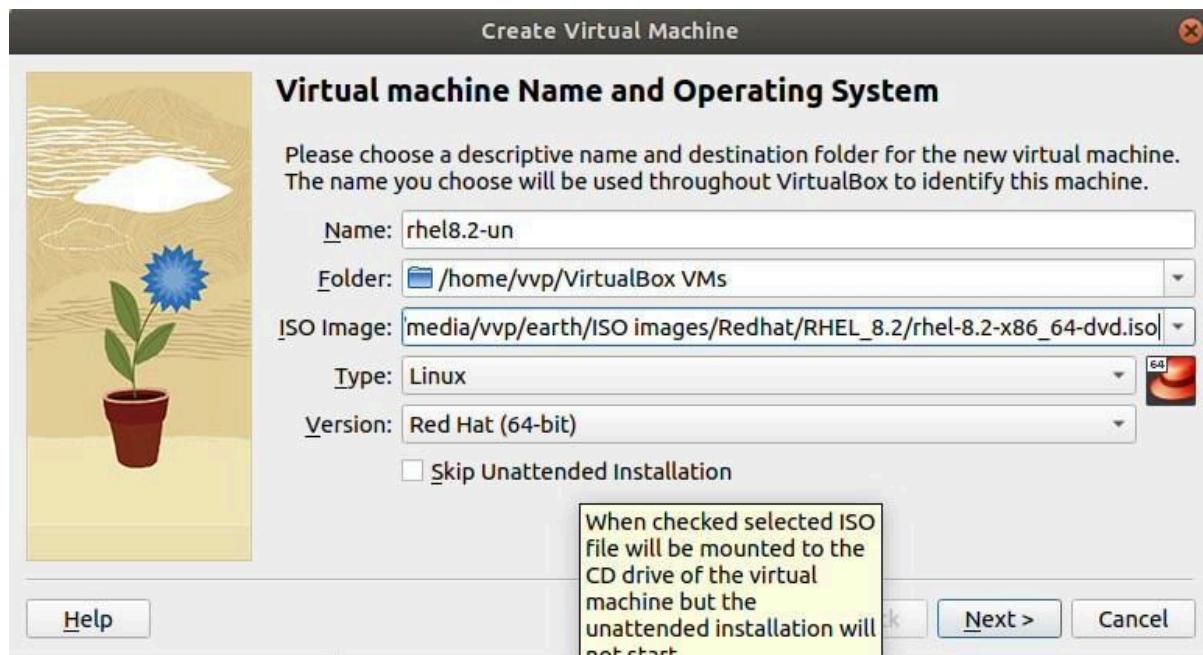
AIM: Installation of Linux and windows using VMWare.

Objectives: The objective of installing Linux and Windows using VMware is to create virtualized environments for both operating systems within a single physical machine, enabling efficient testing, development, and resource management.

Components: virtual Machine box , ISO file of Linux , Command prompt.

Image description:





Experiment No 4.

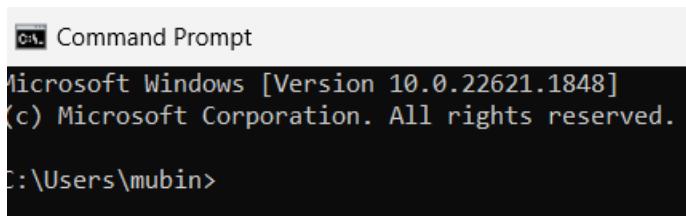
AIM: Learning linux and windows commands.

Objectives:

The objective of learning Linux and Windows commands is to gain the ability to effectively interact with and manage the operating systems through the command-line interface, enabling efficient system administration, troubleshooting, automation, and task execution.

Image Description:

>>>Click start Type cmd & press enter.

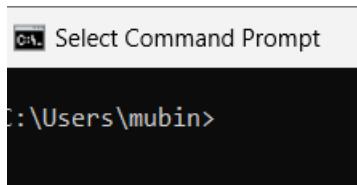


```
cmd Command Prompt
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mubin>
```

1. 1) CLS (Clear Screen)

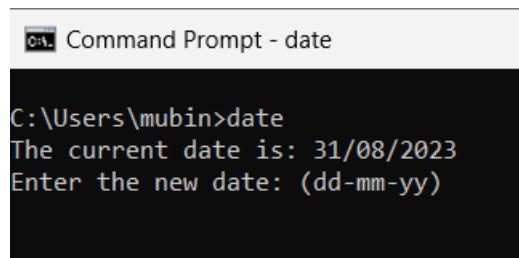
Running the cls command at the command prompt would clear your screen of all previous text and only return the prompt. It does not clear the user's history of commands.



```
cmd Select Command Prompt
C:\Users\mubin>
```

2. DATE

The date command can be used to look at the current date of the computer as well as change the date to an alternate date.



```
cmd Command Prompt - date
C:\Users\mubin>date
The current date is: 31/08/2023
Enter the new date: (dd-mm-yy)
```

3. TIME

Time command is used to display and set the current system time of the operating system.

```
cmd Command Prompt - time

C:\Users\mubin>time
The current time is: 22:09:52.40
Enter the new time:
```

4. VER

(Version) VER will tell you what version of DOS you are using

```
C:\Users\mubin>ver

Microsoft Windows [Version 10.0.22621.1848]

C:\Users\mubin>
```

5. VOL (Volume)

Displays the volume information about the designated drive.

```
C:\Users\mubin>vol
Volume in drive C is Windows
Volume Serial Number is 90C6-01AA

C:\Users\mubin>
```

6. DIR (Directory)

The dir command allows you to see the available files and directories in the current directory.

The dir command also shows the last modification date and time, as well as the file size.

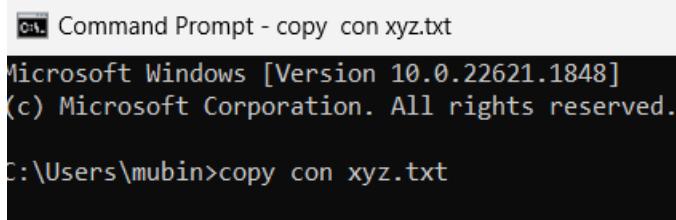
```
Directory of C:\Users\mubin

24/08/2023  17:08    <DIR>      .
22/04/2023  15:32    <DIR>      ..
25/08/2023  13:03    <DIR>      .VirtualBox
29/07/2023  13:31    <DIR>      ansel
22/04/2023  15:15    <DIR>      Contacts
23/08/2023  20:55    <DIR>      Desktop
19/08/2023  15:36    <DIR>      Documents
31/08/2023  22:02    <DIR>      Downloads
22/04/2023  15:15    <DIR>      Favorites
22/04/2023  15:15    <DIR>      Links
22/04/2023  15:15    <DIR>      Music
22/04/2023  15:17    <DIR>      OneDrive
02/05/2023  09:40    <DIR>      Pictures
22/04/2023  15:15    <DIR>      Saved Games
22/04/2023  15:32    <DIR>      Searches
22/04/2023  18:21    <DIR>      Videos
                  0 File(s)          0 bytes
                 16 Dir(s)  197,151,940,608 bytes free
```

If you want to create new .txt file in C:\ using CMD then type
 dir> filename.txt

7. Copy con

Copy con is an MS-DOS and Windows command line command that allows the creation of a file through the command prompt



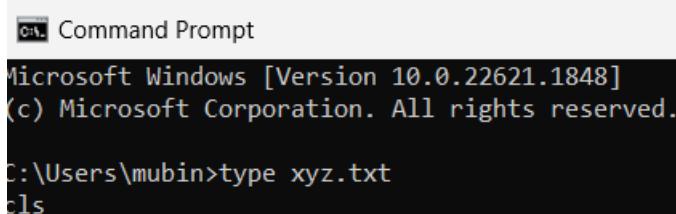
```
Command Prompt - copy con xyz.txt
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mubin>copy con xyz.txt
```

8. TYPE

Allows the user to see the contents of a file. To edit the files, the user would need to use either edit or copy con.

Type [drive:][path]filename



```
Command Prompt
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mubin>type xyz.txt
cls
```

9. MKDIR

Allows you to create your own directories in MS-DOS

MKDIR [drive:]path

MD [drive:]path

```
C:\Users\mubin>mkdir mubin1
```

```
C:\Users\mubin>
```

10. CHDIR (CD)

Chdir (Change Directory) is a command used to switch directories in MS DOS

```
C:\Users\Administrator>chdir My Documents  
C:\Users\Administrator\My Documents>chdir..  
C:\Users\Administrator>
```

11. RMDIR

Removes an empty directory in MS-DOS.

rmdir dirname

```
C:\Users\Administrator>rmdir Nehal_Patel  
C:\Users\Administrator>
```

12. RENAME

Used to rename files and directories from the original name to a new name.

rename c:\computer hope

```
C:\Users\Administrator>rename d:\nehal sneh  
C:\Users\Administrator>
```

13. DEL

Del is a command used to delete files from the computer.

```
C:\Users\Administrator>del nehal.txt  
C:\Users\Administrator>
```

14. MOVE

Allows you to move files or directories from one folder to another, or from one drive to another

```
C:\Users\Administrator>move hello.docx d:  
      1 file(s) moved.  
  
C:\Users\Administrator>move 1.py c:\TC  
      1 file(s) moved.  
  
C:\Users\Administrator>
```

15. COPY

Allows the user to copy one or more files to an alternate location.

```
copy *.txt d:\
```

```
C:\Users\Administrator>copy *.txt d:\  
NEHAL.txt  
PATEL.txt  
XYZ.TXT  
      3 file(s) copied.  
  
C:\Users\Administrator>
```

16. DOSKEY

Doskey is an MS-DOS utility that allows a user to keep a history of commands used on the computer. Doskey allows frequently used commands to be used without having to type the command each time its needed

```
C:\Users\Administrator>DOSKEY /HISTORY  
CLS  
DOSKEY /HISTORY  
  
C:\Users\Administrator>
```

17. PATH

Path is used to specify the location where MS-DOS looks when using a command. Typing "path" by itself shows the current path information.

```
C:\Users\Administrator>PATH  
PATH=C:\Program Files (<x86>)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (<x86>)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\ MATLAB\R2012a\runtime\win64;C:\Program Files\ MATLAB\R2012a\bin;C:\Program Files\MiKTeX 2.9\miktex\bin\x64\;c:\users\administrator\appdata\local\enthought\canopy\user\scripts;C:\Users\Administrator\AppData\Local\Enthought\Canopy\User;C:\Users\Administrator\AppData\Local\Enthought\Canopy\User\Scripts  
C:\Users\Administrator>
```

External Commands

1. ATTRIB

Attrib allows a user to change the attributes of a file or files.

R	Read-only file attribute.
A	Archive file attribute.
H	Hidden file attribute.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator>attrib /?
Displays or changes file attributes.

ATTRIB [+R | -R] [+A | -A] [+S | -S] [+H | -H] [+I | -I]
[drive:]|[path][filename] [/S [/D] [/L]]

+ Sets an attribute.
- Clears an attribute.
R Read-only file attribute.
A Archive file attribute.
S System file attribute.
H Hidden file attribute.
I Not content indexed file attribute.
[drive:]|[path][filename]
Specifies a file or files for attrib to process.
/S Processes matching files in the current folder
and all subfolders.
/D Processes folders as well.
/L Work on the attributes of the Symbolic Link versus
the target of the Symbolic Link
```

```
C:\Users\Administrator>cd\
C:\>cd nehal
C:\nehal>attrib -h hello.txt
C:\nehal>attrib +h hello.txt
C:\nehal>
```

2. FORMAT

Format is used to erase information off of a computer diskette or fixed drive.

```
C:\>format g:/q
Insert new disk for drive G:
and press ENTER when ready...
The type of the file system is FAT32.
QuickFormatting 7435M
Initializing the File Allocation Table <FAT>...
Volume label (11 characters, ENTER for none)?
```

3. CHKDSK

Chkdsk is a utility that checks the computer's hard drive status for any crosslinked or any additional errors with the hard drive.

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>chkdsk
The type of the file system is NTFS.

WARNING! F parameter not specified.
Running CHKD SK in read-only mode.

CHKD SK is verifying files (stage 1 of 3)...
File verification completed.
CHKD SK is verifying indexes (stage 2 of 3)...
Index verification completed.
CHKD SK is verifying security descriptors (stage 3 of 3)...
Security descriptor verification completed.
CHKD SK is verifying Usn Journal...
Usn Journal verification completed.

16609288 KB total disk space.
15005300 KB in 85352 files.
    57336 KB in 10426 indexes.
        0 KB in bad sectors.
    209688 KB in use by the system.
        65536 KB occupied by the log file.
    1336964 KB available on disk.

        4096 bytes in each allocation unit.
    4152322 total allocation units on disk.
        334241 allocation units available on disk.

C:\Documents and Settings\Administrator>

```

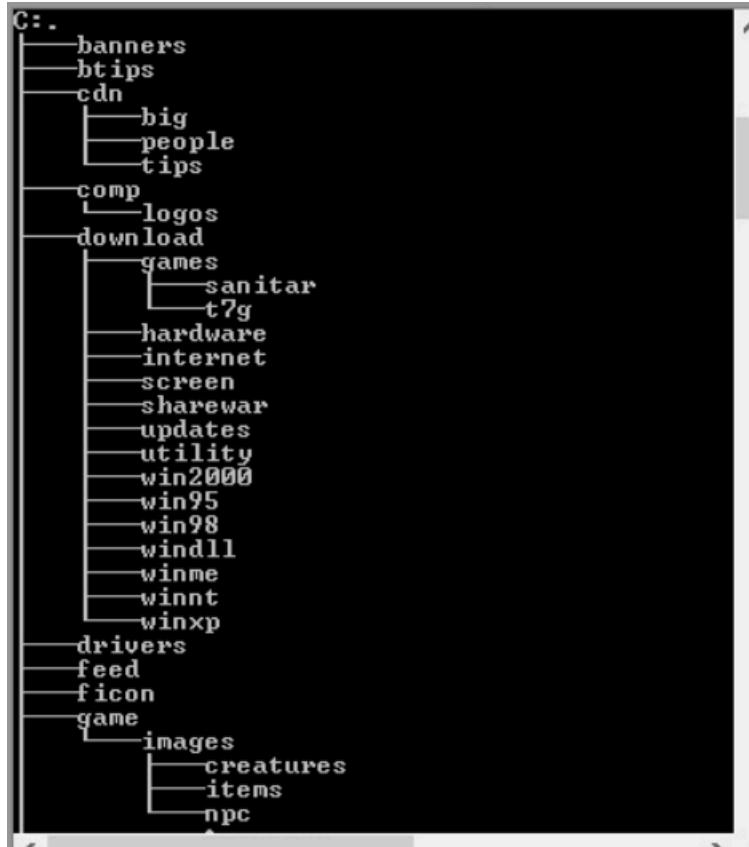
4. SCANDISK

Microsoft ScanDisk was first introduced with MS-DOS 6.2 and is a software utility capable of checking the hard drive and floppy diskette drive for any disk errors.

5. TREE

Allows the user to view a listing of files and folders in an easy to read listing.

Drive:\Path	Drive and directory containing disk for display of directory structure.
/F	Displays file names in each directory.
/A	ext characters used for linking lines, instead of graphic characters. /a is used with code pages that do not support graphic characters and to send output to printers that do not properly interpret graphic characters.



6. XCOPY

Xcopy is a powerful version of the copy command with additional features; has the capability of moving files, directories, and even whole drives from one location to another.

```
C:\>xcopy patel.txt d:\  
C:PATEL.txt  
1 File(s) copied  
C:\>
```

Move one folder from one drive to another.

```
C:\>xcopy IT /s d:  
IT\New Text Document.txt  
1 File(s) copied  
C:\>
```

Linux commands

1.mkdir

```
ubuntu@ubuntu:~$ mkdir mubin1
```

2.vi

```
ubuntu@ubuntu:~$ vi mmk
```

hello world

"mmk" [New File]

3.ls

```
ubuntu@ubuntu:~$ ls
Desktop   Downloads  Mmk    Mubin   Music    Public    Videos
Documents mmk       mubin  mubini1 Pictures Templates
```

4.hostname

```
ubuntu@ubuntu:~$ hostname
ubuntu
ubuntu@ubuntu:~$
```

5.date

```
ubuntu@ubuntu:~$ date
Tue 21 Nov 2023 09:32:53 AM UTC
```

6.ifconfig

```
ubuntu@ubuntu:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
```

7.remove directory(rm -rf <directory name>)

```
ubuntu@ubuntu:~$ rm -rf Mubin
ubuntu@ubuntu:~$ ls
Desktop   Downloads  Mmk    mubini1 Pictures Templates
Documents mmk       mubin  Music    Public    Videos
```

8.top

```
ubuntu@ubuntu:~$ top

top - 09:34:11 up 19 min,  1 user,  load average: 0.01, 0.12, 0.16
Tasks: 177 total,   1 running, 176 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.0 us,  2.3 sy,  0.0 ni, 92.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 1987.4 total,    150.4 free,    703.9 used,   1133.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.  1043.8 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
4243 ubuntu    20   0 3773992 361140 123296 S  3.3 17.7  0:38.15 gnome-
+ 4097 ubuntu    20   0  531620  62740  39784 S  1.7  3.1  0:09.95 Xorg
5440 ubuntu    20   0  823284  51184  38520 S  0.7  2.5  0:01.01 gnome-
+ 1 root       20   0 102356  11848   8572 S  0.0  0.6  0:00.76 systemd
2 root       20   0      0      0      0 S  0.0  0.0  0:00.00 kthrea+
3 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp
4 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_pa+
5 root      20   0      0      0      0 I  0.0  0.0  0:00.98 kwork+
6 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kwork+
8 root      20   0      0      0      0 I  0.0  0.0  0:00.13 kwork+
9 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 mm_per+
```

9.Rename files(`mv <old file name> <new name>`)

```
ubuntu@ubuntu:~$ mv mmk Mubin14
ubuntu@ubuntu:~$ ls
Desktop  Downloads  mubin  Mubin14  Pictures  Templates
Documents  Mmk          mubini  Music    Public    Videos
```

10.word count(`wc <file name>`)

```
ubuntu@ubuntu:~$ wc mmk
 3  2 14 mmk
```

11.clear

```
ubuntu@ubuntu:~$ clear
```

```
ubuntu@ubuntu:~$
```

12. Display size of files and Directories(`du -h <somepath>`)

```
ubuntu@ubuntu:~$ du -h Mubin14
4.0K    Mubin14
```

13.echo

```
ubuntu@ubuntu:~$ echo hello everyone
hello everyone
```

14.head of file(head <filename>)

```
ubuntu@ubuntu:~$ head Mubin14
hello world
```

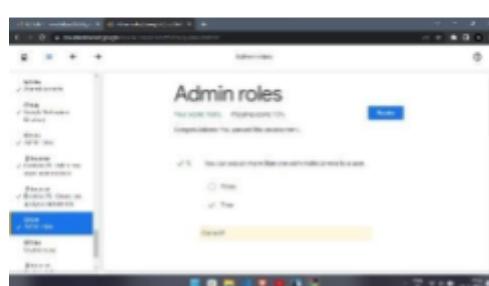
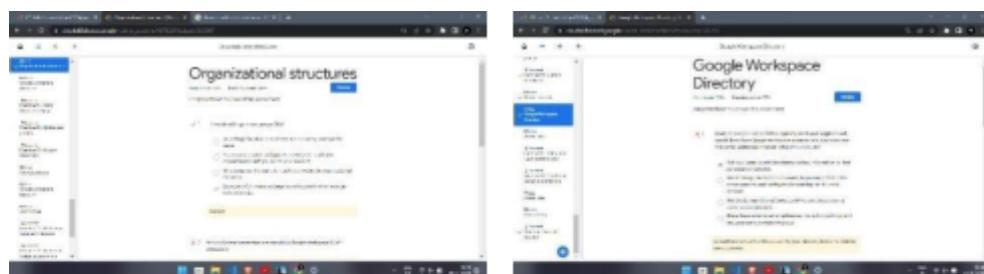
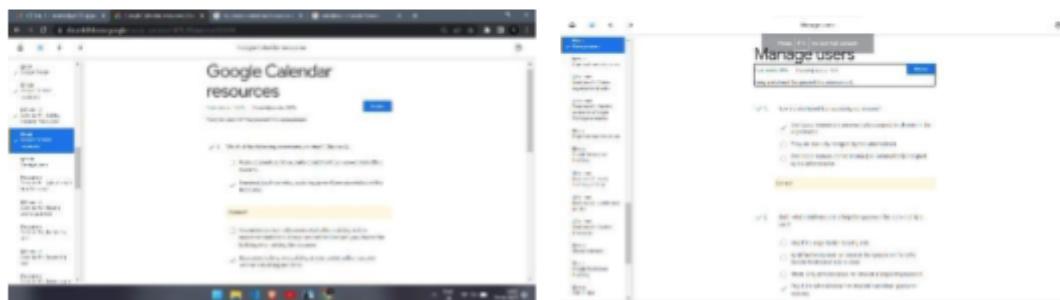
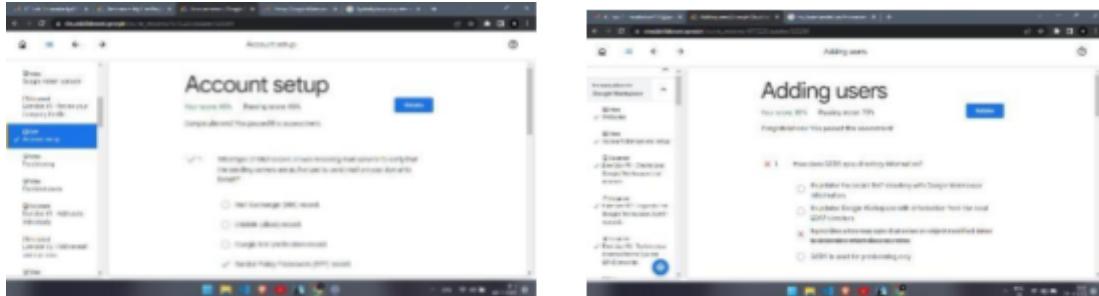
15. Display kernel message log(dmesg)

```
ubuntu@ubuntu:~$ dmesg
[    0.000000] Linux version 5.8.0-43-generic (buildd@lcy01-amd64-018) (gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #49~20.04.1-Ubuntu SMP Fri Feb 5 09:57:56 UTC 2021 (Ubuntu 5.8.0-43.49~20.04.1-generic 5.8.18)
[    0.000000] Command line: file=/cdrom/preseed/ubuntu.seed initrd=/casper/initrd quiet splash --- maybe-ubiquity
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Hygon HygonGenuine
[    0.000000]   Centaur CentaurHauls
[    0.000000]   zhaoxin Shanghai
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[    0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,
[    0.000000]   using 'standard' format.
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000007ffeffff] usable
[    0.000000] BIOS-e820: [mem 0x0000000007ffff0000-0x0000000007ffffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc00000-0x00000000ffffffff] reserved
```

Experiment-5

Aim: To learn about and get Basic Introduction of Google Workspace.

Objective: SCREENSHOTS OF QUIZZES/ASSIGNMENTS SUBMITTED DURING MOOC:



Google Cloud

Introduction to Google Workspace Admin

COMPLETION BADGE



Certificate no: UC-35fa61f6-e330-446a-8275-33e107168a50

Certificate url: ude.my/UC-35fa61f6-e330-446a-8275-33e107168a50

Reference Number: 0004

CERTIFICATE OF COMPLETION

Complete Google Workspace (G Suite), Beginner - Advanced

Instructors **Laurence Svekis, Paul Ogier**

Het Patel

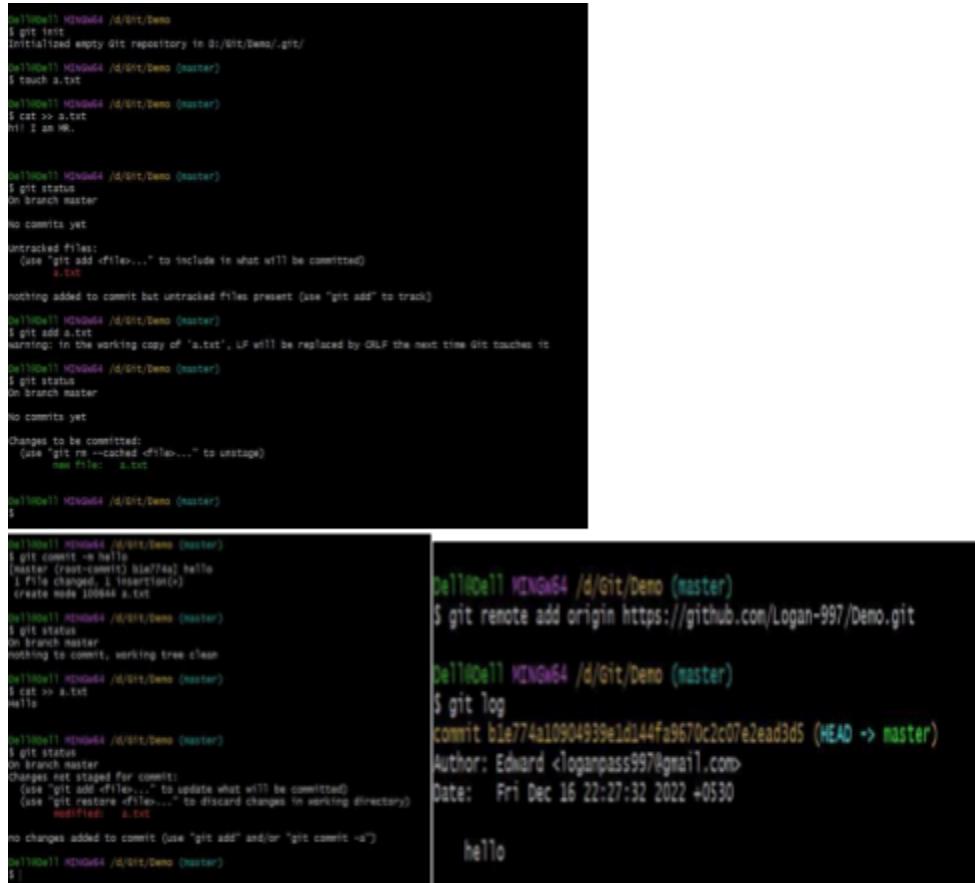
Date **Nov. 30, 2024**

Length **21.5 total hours**

Experiment-6

Aim: To learn about Git & Github

Objective: SCREENSHOTS OF Task during the Session:



```

$ cd /d/Git/Demo
$ git init
Initialized empty Git repository in /d/Git/Demo/.git/
$ touch a.txt
$ cat >> a.txt
hi I am HI.

$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .txt

nothing added to commit but untracked files present (use "git add" to track)

$ git add a.txt
warning: The working copy of 'a.txt', LF will be replaced by CRLF the next time Git touches it

$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: a.txt

$ git commit -m hello
[master (root-commit) b1e774a] hello
 1 file changed, 1 insertion(+)
 create mode 100644 a.txt

$ git status
On branch master
nothing to commit, working tree clean

$ cat >> a.txt
Hello

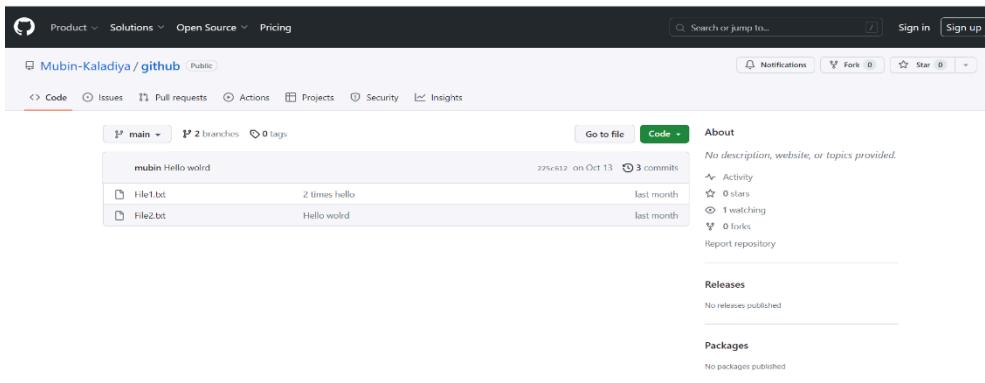
$ git status
On branch master
Changes to be committed:
  (use "git add <file>..." to update what will be committed)
    new file: a.txt

no changes added to commit (use "git add" and/or "git commit -a")
$ git remote add origin https://github.com/Logan-997/Demo.git
$ git log
commit b1e774a10904939e1d144fa9670c2c07e2ead35 (HEAD -> master)
Author: Edward <loganspass997@gmail.com>
Date:   Fri Dec 16 22:27:32 2022 +0530

    hello

```

SCREENSHOTS OF GITHUB ACCOUNT:



Mubin-Kaladiya / github Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

[main](#) [2 branches](#) [0 tags](#) [Go to file](#) [Code](#)

mubin Hello world · 225612 · on Oct 13 · 3 commits

File	Content	Last Commit
file1.txt	2 times hello	last month
file2.txt	Hello world	last month

About
No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases
No releases published

Packages
No packages published

Experiment-7

DCN Components (ICT)

Sr. No.	Components Name & Picture	Description of Component
1	Ethernet Cables 	An Ethernet cable is also referred as a network cable. It is a cord that runs from a router, modem or a network switch to your computer, giving your device access to a local area network. Ethernet connection are almost always faster than a Wi-Fi connections, and are usually more stable. You will need to connect one end of ethernet cable to router and another end to your computer.
2	Clamping Tool 	A clamping tool is the tool used to deform the material and create the connection. Crimping is commonly used in electrical work, to attach wires together or wire to another connectors.
3	Console Cable	The console cable is used for the serial connect between your computer's serial port and your console port on your TP-link switch or router to access the CLI (Command line interface) of the device. There are many types of console cable in the market. The cable shown in the given diagram is RS232 DB9 female head <--> RJ45 connector.

4	 Cable Tester	<p>A cable tester is an electronic device used to verify the electrical connections in a signal cable or other wired cable. Basic cable tester are continuity tester that verify the existence of a conductive path between the ends of cable, and verify the correct wiring of connectors of cable. More advanced cable tester can measure the signal transmission properties of cable such as its resistance, signal attenuation, noise and interference.</p>
---	---	---

5	Manageable Switch 	<p>A managed network switch is a technology that allows Ethernet devices to communicate with each other, and that contains feature to configure, manage and monitor traffic on Local Area Network (LAN). A managed network switch provides more control over how data travels over the network and who can access it.</p>
6	Normal Switch 	<p>It is also known as network switch is networking hardware that connects devices on a computer network by using packet switching to receive and forward data to destination device.</p>

7	Router 	<p>A router is a device that connects two or more packet switched networks or sub-networks. It serves two primary functions: managing traffic between these networks by forwarding data packets to their intended IP addresses, and allowing multiple devices to use the same internet connection.</p>
8	Chromecast 	<p>Chromecast is a device that allows users to wirelessly “cast” their content from other devices, so they can watch their favorite movies, TV shows, photos, websites, and more on their TV.</p>
9	Passive Extender 	<p>Passive extender is designed to extend the range of any IP device beyond 300ft limit of Ethernet. The unit does not filter out any protocols, codes, or applications, ensuring compatibility with any IP camera and its software.</p>

10	Active Extender		It ensures you get the best performance and maximum data integrity when you transfer information between your computer and devices
11	Converter		The work of converter is to convert the ethernet into different purpose. The ethernet cable of router comes inside converter and with the help of other ethernet cable we can get ethernet in more than one computer.
12	Normal ethernet extender		An Ethernet extender is a device used to extend an Ethernet or network segment beyond its inherent distance limitation which is approximately 100 meters (330ft) for most common forms of twisted pair Ethernet.

13	Passive Splitter 	A splitter is a device used to split the cable signal to two or more devices
14	USB to Ethernet Cable 	A USB to Ethernet cable is a device that is capable of connecting a USB port to an Ethernet cable. USB to Ethernet cable allows users to connect multiple devices together via an Ethernet cable rather than using an USB cable, which is generally shorter and less reliable.
15	POE Splitter 	POE splitters are the devices that are used together with POE switches and POE injectors. Instead of taking both data input and power input and turning them into singular output, they supply power from splitting the power from data and feeding it into a separate input.

16	NIC Card for optical fiber 	It is known as fiber LAN card or fiber server adapter. It is like a circuit board which is designed to allow computers or servers to communicate with a network. It has a unique physical network address, also referred as MAC address.
17	NIC Card for wireless 	Wireless network NIC cards consists of a small antenna integrated onto a card, where the communication between the various devices is set up wirelessly using the routers and various network protocols.
18	NIC Card for wireline 	A network interface card (NIC) is a Hardware component, typically a circuit board or chip, which is installed on a computer so that it can connect to a network.

19	Faceplate Punch Tool	 <p>To use a punch tool such as this, a worker presses the sharp end against a work piece, after which he or she hits the blunt end with a hammer or mallet. When the hammer or mallet hits the punch tool, the tool's sharp end will cut into the work piece.</p>
20	Rack	 <p>A rack server, also known as a rack mount server, rack-mounted server or rack mount computer, is a computer designed to be situated in a rectangular structure called a server rack.</p>

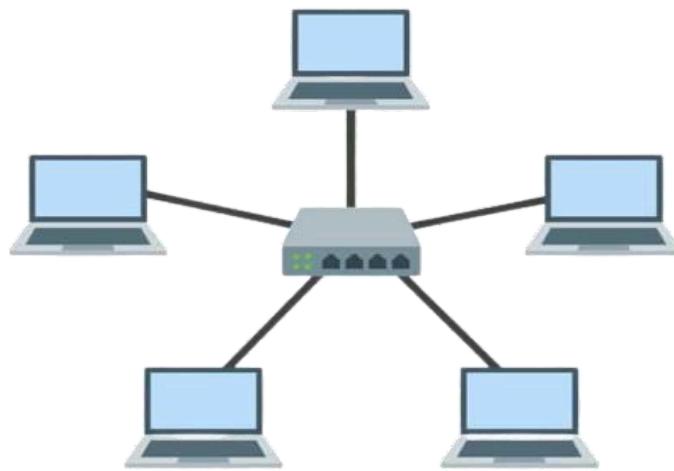
Assign IP Address to each Network:

1. BUS Topology

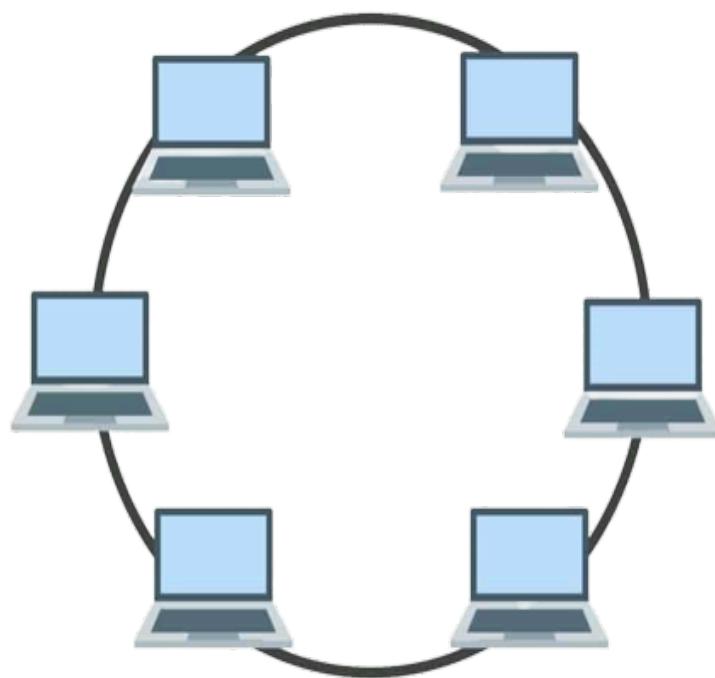


Page No:14

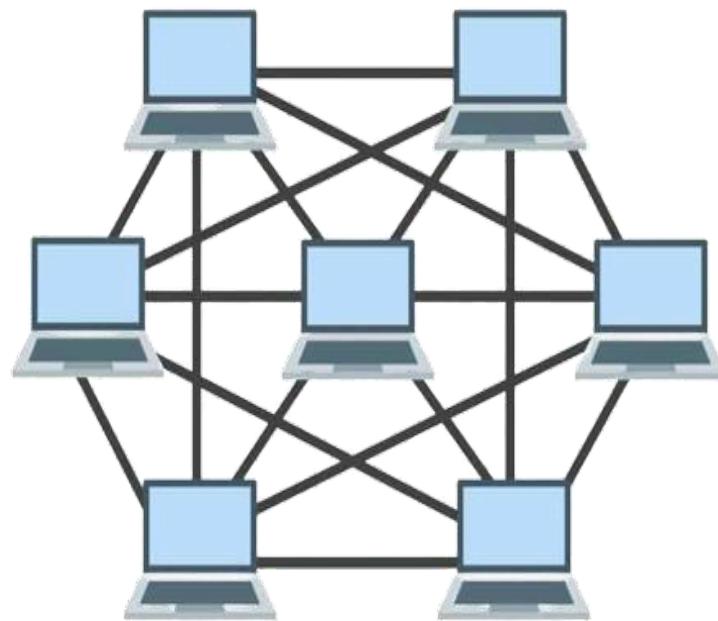
2. Star technology



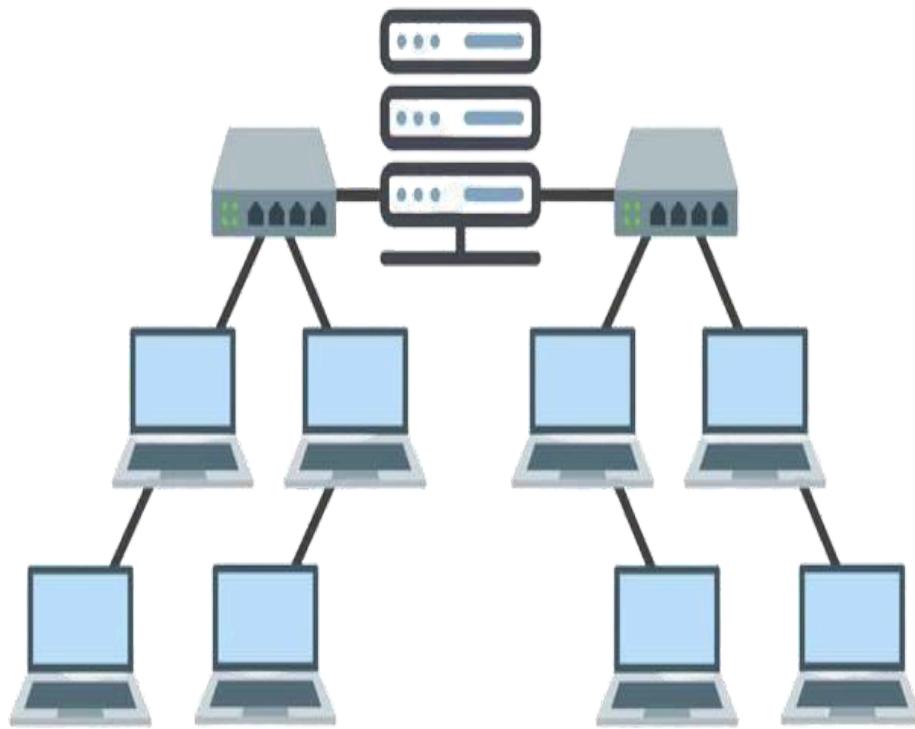
3. Ring Topology



4. Mesh topology



5. Tree Topology



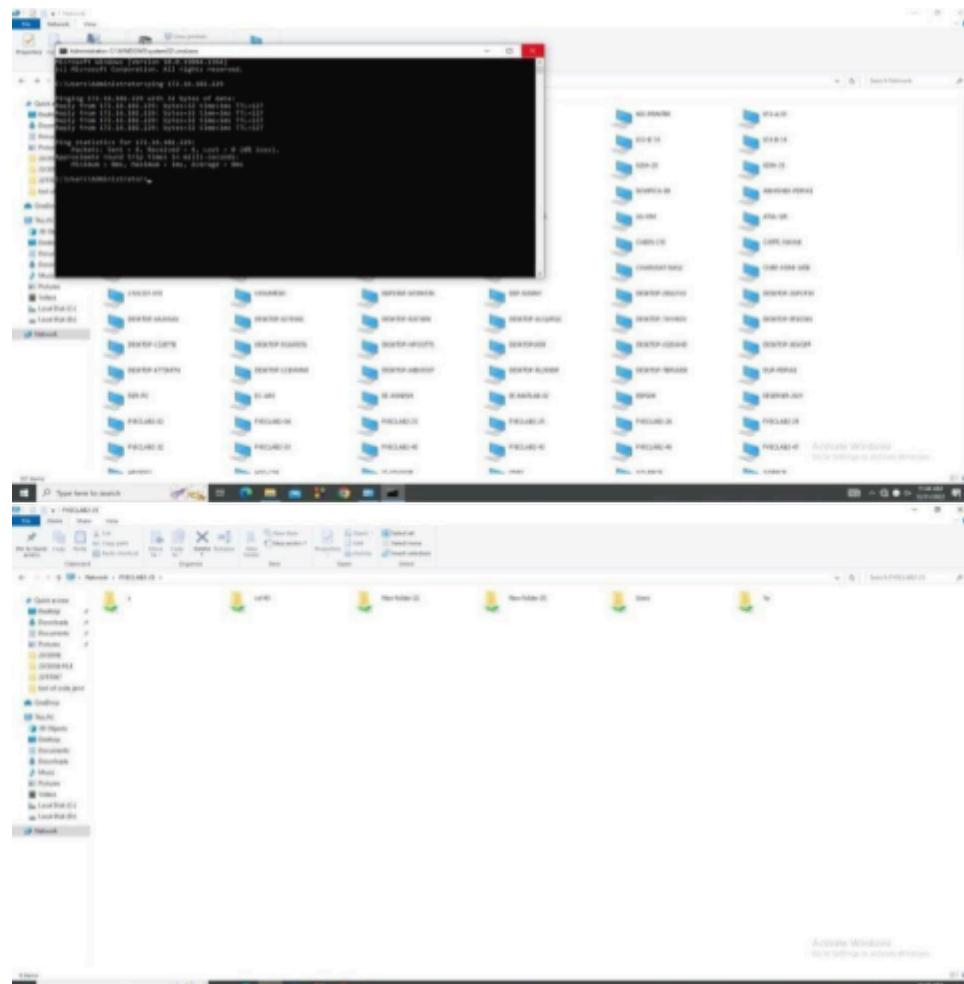
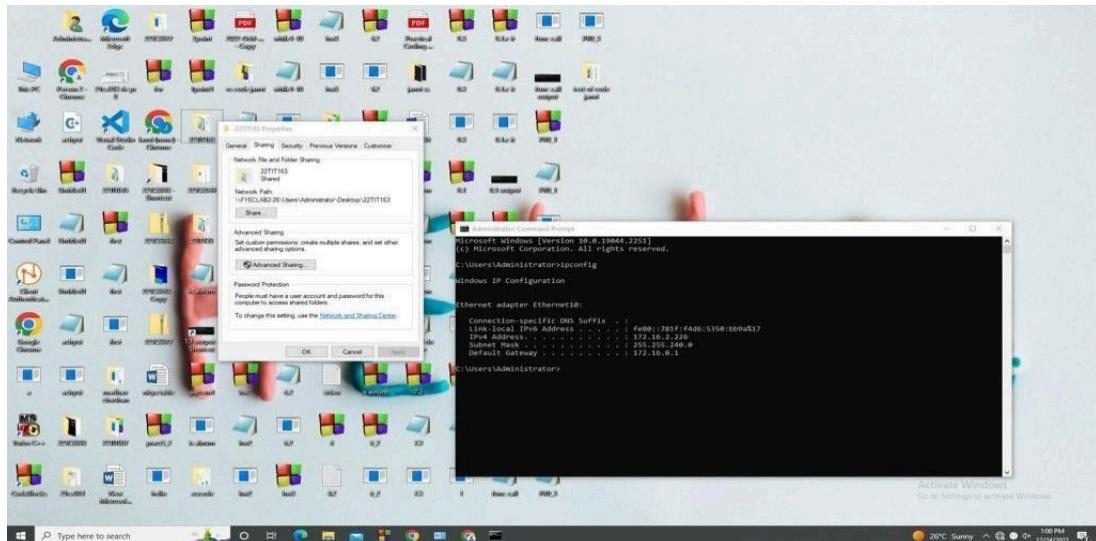
Experiment-8

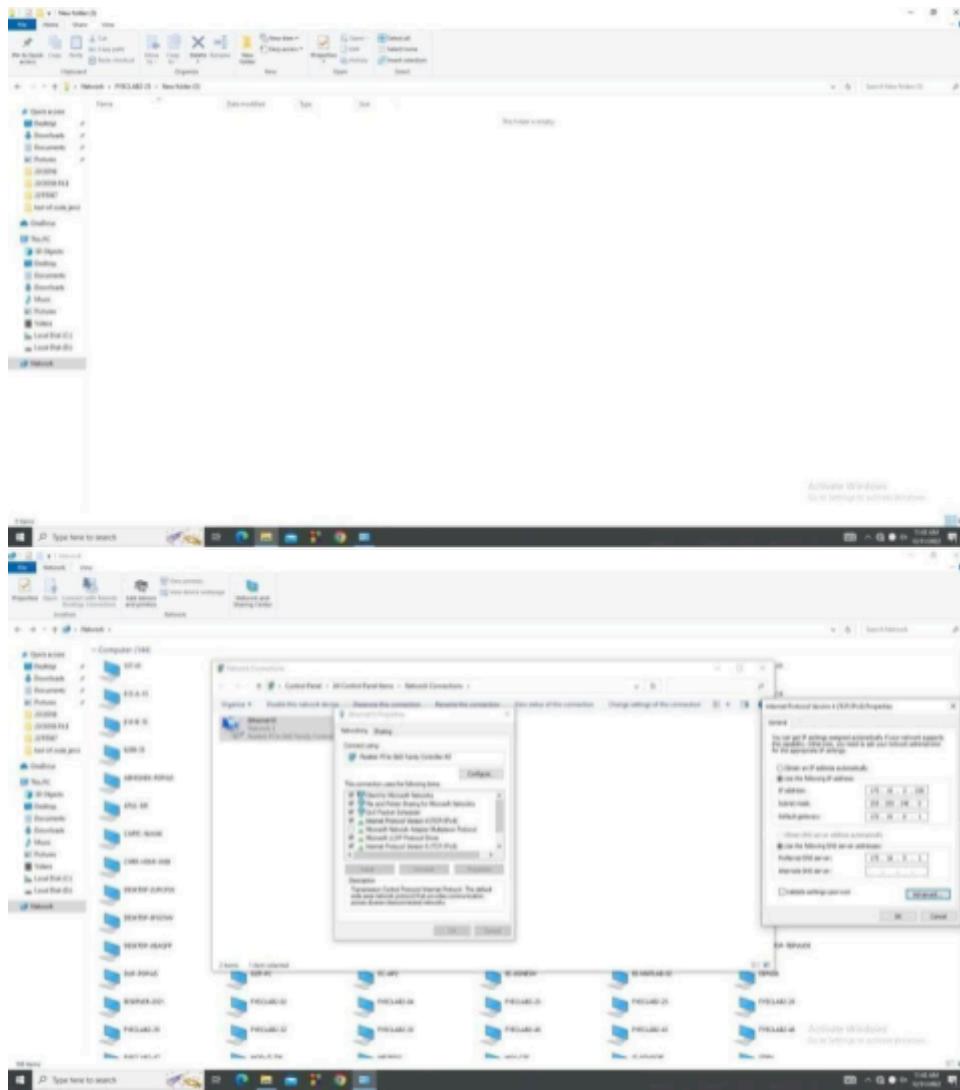
AIM: Understanding basic concepts of networking and studying the details of components involved in networking to use them efficiently.

- 1. Bits and Bytes:** A bit is the smallest unit of storage. A bit stores just a 0 or 1 "In the computer it's all 0's and 1's" ... bits, anything with two separate states can store 1 bit. In a chip: electric charge = 0/1. In a hard drive: spots of North/South magnetism = 0/1. A bit is too small to be much use. One byte = collection of 8 bits. e.g., 0 1 0 1 1 0 1 0. One byte can store one character, e.g. 'A' or 'x' or '\$'
- 2. Types of Network Cables:** There are three types of network cable:
 - i. Wire cable ii. Fiber Optics cable
 - iii. Wireless cable
- 3. Ports and network cable:** The ports can vary from network port, ethernet port, LAN (Local Area Network) port or WAN (Wide Area Networks) port. And the network cable is the ethernet cable having 8 wires in it where 2 pairs are connected for future purposes.
- 4. Types of ethernet:** There are three types of ethernet -
 - i. Ethernet with 10 mbp/s ii. Fast ethernet with 100 mbp/s iii. Gigabyte ethernet with 1000 mbp/s
- 5. Transfer of hello world from one pc to other:** It can be done by sharing through USB cable or through wireless Bluetooth or by op address.
- 6. Importance of speed of data:** Performance plays a key role in the success of any online venture. High-performing sites engage and retain users better than low-performing ones.

7. Network cards: A network interface card (NIC) is a hardware component, typically a circuit board or chip, which is installed on a computer so it can connect to a network. Modern NICs provide functionality to computers, such as support for I/O interrupt, direct memory access (DMA) Interfaces, data transmission, network traffic engineering and partitioning.

8. IP Address:



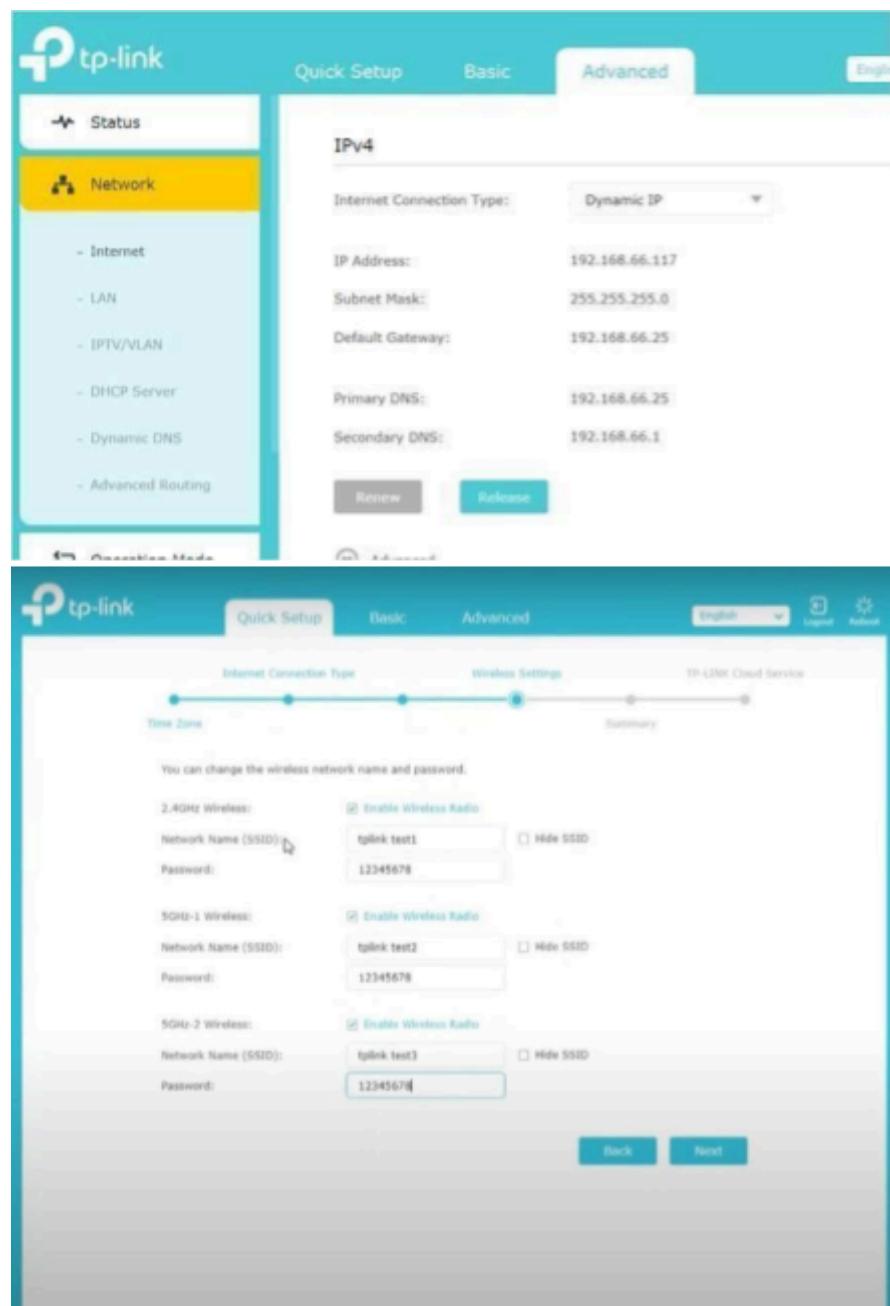


9. Configuring Wi-Fi Router:

- Using your web browser, enter the router's default IP address into the address bar, then press Enter. Your router's instructions should include this information, but some of the most common addresses include 192.168.0.1, 192.168.1.1, and 192.168.2.1.
- The router's sign-in page will appear. Again, the exact sign-in details should be included with your router's instructions, but most routers use a standard user name and password combination, such as admin and password.

- Your router's settings page will appear. Locate and select the Network Name setting, then enter a unique network name.
- Locate and select the Network Password setting, and choose an Encryption option. There are several types of encryptions you can use, but we recommend WPA2, which is the most secure.
- Enter your desired password. Make sure to use a strong password to help ensure no one else can access your network.

Locate and select the Save button to save your settings.



OBSERVATIONS:

As discussed in the class we learnt about

- How bytes and bits works at ground level for different purposes like image processing, Video, etc
- Types of Network cables required, How they work, Using which one gives benefit to the system
- Using different ports and then using Different devices to Split and couple the Cables
- Types of Ethernet Cable and How to Get more speed ie: Giga Ethernet
- Importance of Network cards and how do they support the hardware
- Learnt Ip addressing and last but not the least Configuring Routers
- In Routers We Learnt
- Configuring Subnets, User id, Pass
- Configuring Isp, Dns servers
- Using Parental controls and Much more
- This all Concluded that one who has Knowledge in the field of Networking can easily work with computers and he/she can practically understand How things are working and can make own servers and all stuffs..!

Experiment-9

AIM: Introduction to Arduino/ESP32 & implement sample programs of

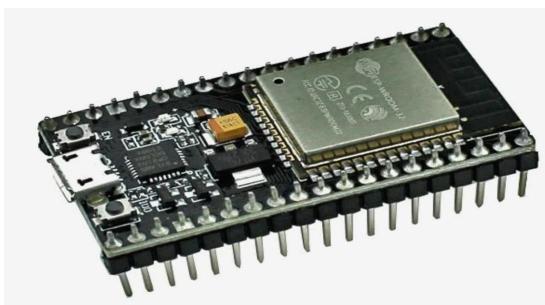
- 9.1: LED Blink
 - 9.2: LED with timer
 - 9.3: LED control from Switch

OBJECTIVES:

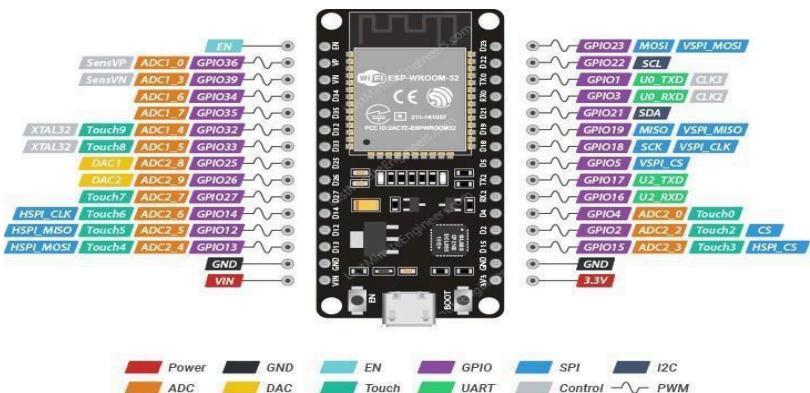
- 9.1: Blink the LED light using Arduino and ESP32
 - 9.2: LED light glow with timer using Arduino and ESP32
 - 9.3: LED light control with Switch using Arduino and ESP32

COMPONENTS:

ESP32:



The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities.



- ### 1) USB CABLE:



USB cable is used to connect the Computer to ESP32

2) LED:



A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it.

4) Button:



Grove-Button is a momentary push button. It contains one independent "momentary on/off" button. "Momentary" means that the button rebounds on its own after it is released. The button outputs a HIGH signal when pressed, and LOW when released.

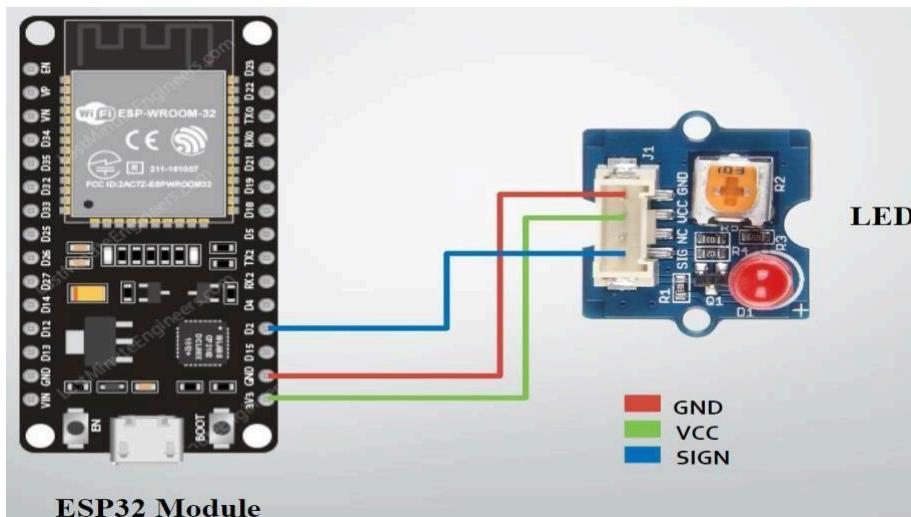
5) Jumper Wires:



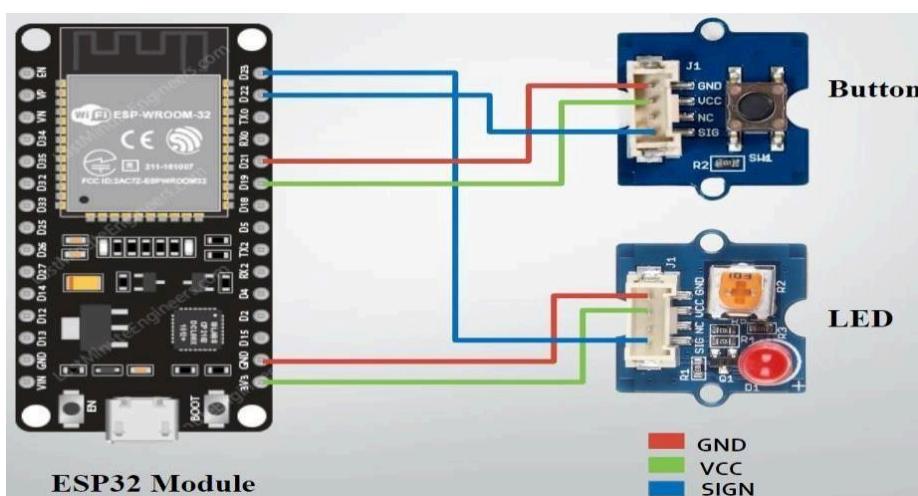
Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins.

CONNECTION DIAGRAM:

a) 9.1 and 9.2



9.3



CODES:

9.1: const int ledpin=2; void setup()

```

{

pinMode(ledpin,OUTPUT);

} void loop()

{

digitalWrite(ledpin,HIGH); delay(1000) digitalWrite(ledpin,LOW); delay(1000);

} 9.2:

int led = 23; int brightness = 0; int fadeAmount = 5; void setup()

{

pinMode(led,OUTPUT);

} void loop() { analogWrite(led,brightness); brightness=brightness + fadeAmount; if

(brightness <=0 || brightness >= 255)

{

fadeAmount = -fadeAmount;

}

delay(30);

}

9.3: const int buttonPin

= 22; const int ledPin

= 23; const int suppl

= 19; const int groun

= 21; int buttonState

= 0; void setup()

{

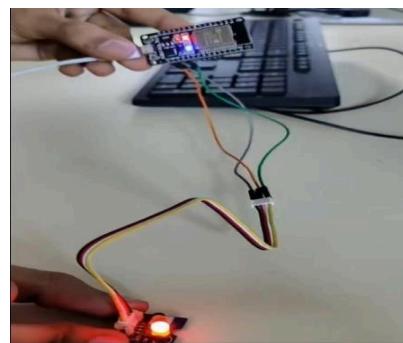
pinMode(ledPin, OUTPUT); pinMode(buttonPin, INPUT); pinMode(suppl, OUTPUT);

pinMode(groun, OUTPUT); digitalWrite(suppl, HIGH); digitalWrite(groun, LOW);

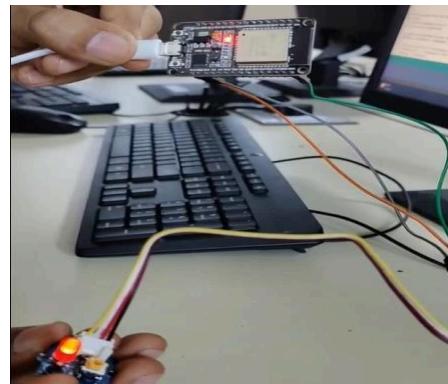
```

```
    } void loop()
    {
        buttonState = digitalRead(buttonPin); if (buttonState == HIGH)
        {
            digitalWrite(ledPin, HIGH);
        } else
        {
            digitalWrite(ledPin, LOW);
        }
    }
```

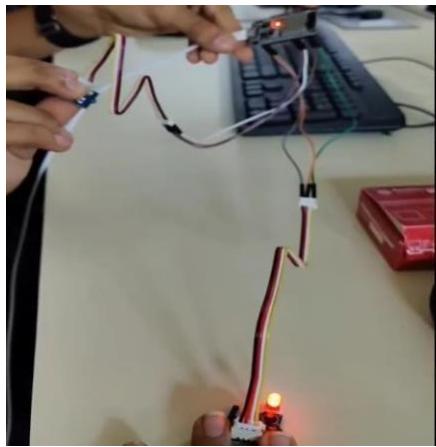
9.1:



9.2:



9.3:



Experiment-10

Aim: • Implement Sensor interfacing with Arduino / ESP32.

10.1: Ultrasonic Sensor

10.2: Temperature with Humidity

10.3: Light Sensor with LED and Buzzer

OBJECTIVES:

10.1: Find the distance to the closest object using Ultrasonic Sensor, Arduino and ESP32

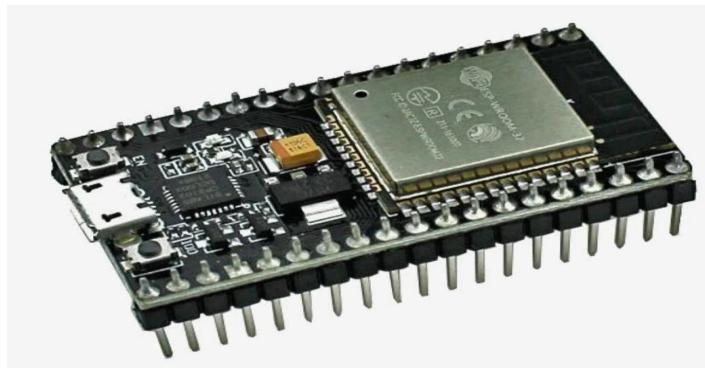
10.2: Measure the Temperature and Humidity using Temperature and Humidity Sensor, Arduino and ESP32

10.3.1: Turn LED On/Off using Light Sensor, Arduino and ESP-32

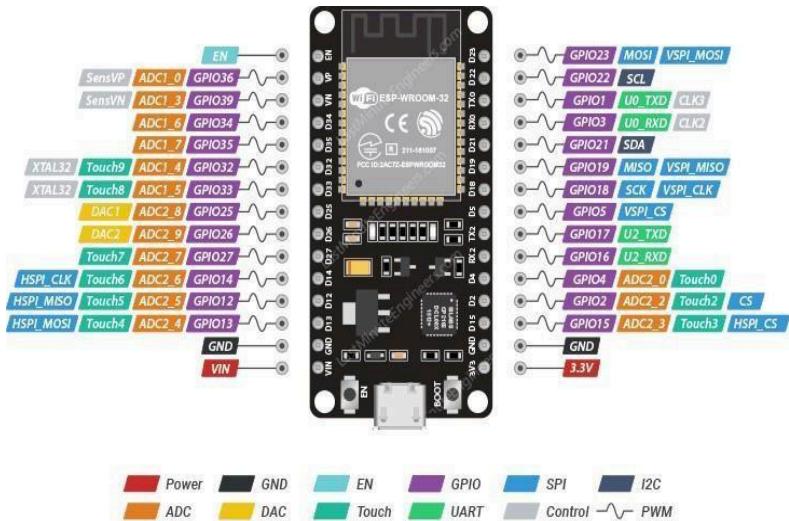
10.3.2: Turn Buzzer On/Off using Light Sensor, Arduino and ESP-32

COMPONENTS:

1) ESP32:



The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities.



2) USB CABLE:



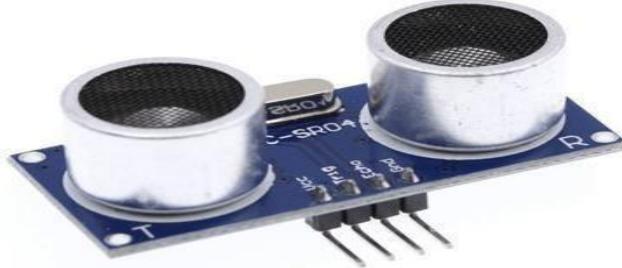
USB cable is use to connect the Computer to ESP32

3) LED:



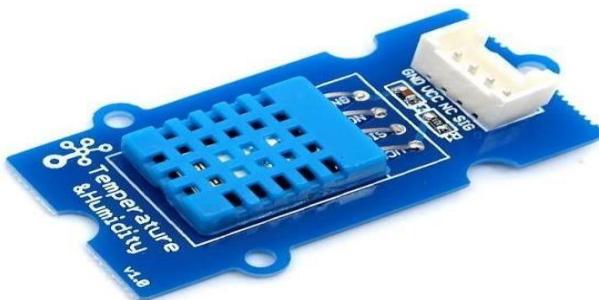
A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it.

4) Ultrasonic Sensor:



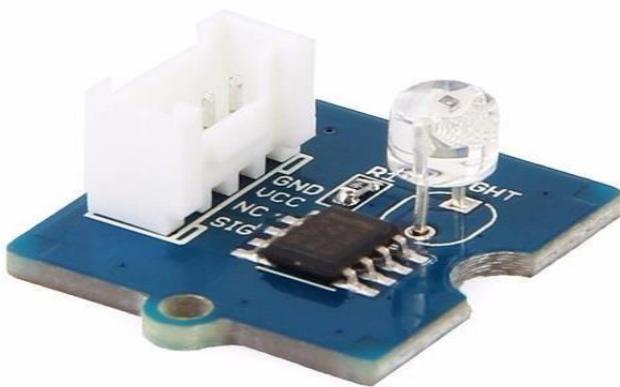
An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal.

5) Temperature and Humidity Sensor:



This is a multifunctional sensor that gives you temperature and relative humidity information at the same time. It utilizes a DHT11 sensor that can meet measurement needs of general purposes.

6) Light Sensor:



The Grove - Light sensor integrates a photo-resistor(light dependent resistor) to detect the intensity of light. The resistance of photo-resistor decreases when the intensity of light increases.

7) Buzzer:



The Grove - Buzzer module has a [piezo buzzer](#) as the main component. The piezo can be connected to digital outputs, and will emit a tone when the output is HIGH. Alternatively, it can be connected to an analog pulse-width modulation output to generate various tones and effects.

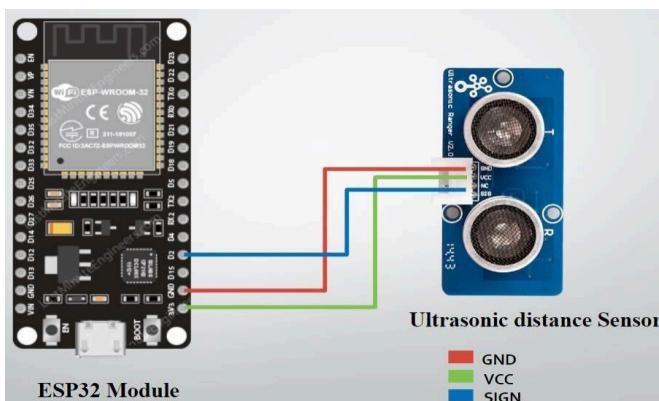
8) Jumper Wires:



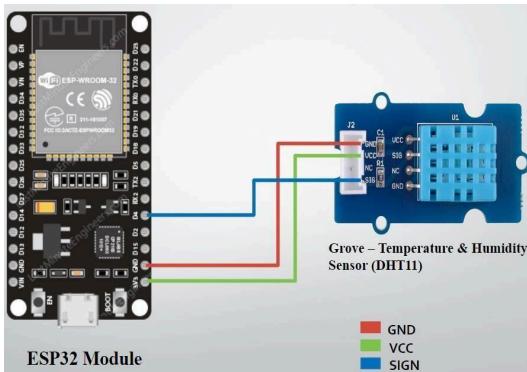
Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins.

CONNECTION DIAGRAM:

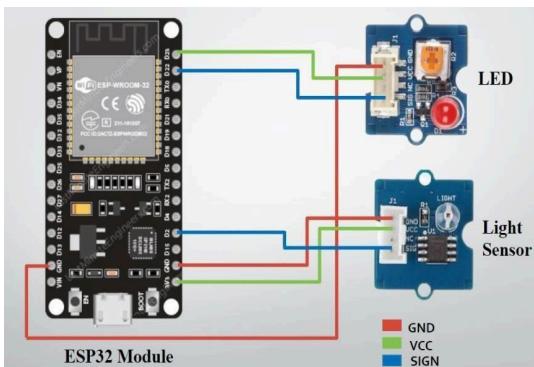
a) 10.1



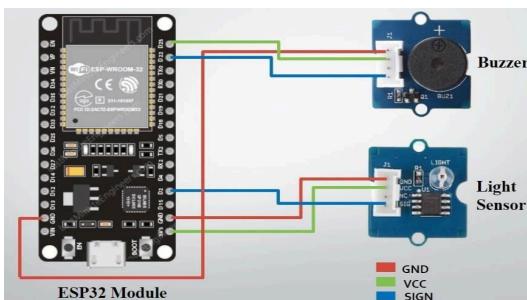
b) 10.2



c) 10.3 1



2)



CODES:

10.1

```
#include "Ultrasonic.h"

Ultrasonic ultrasonic(2); // Output Pin
void setup() {
    Serial.begin(9600); //9600 MHz band
} void loop() {    long
RangeInInches;    long
RangeInCentimeters;
```

```

Serial.println("The distance to
obstacles in front is: ");

RangeInInches = ultrasonic.MeasureInInches();

Serial.print(RangeInInches);//0~157 inches
Serial.println(" inch"); delay(2000);

RangeInCentimeters = ultrasonic.MeasureInCentimeters(); // two measurements should keep
an interval

```

```
Serial.print(RangeInCentimeters);//0~400cm
```

```
Serial.println(" cm"); delay(2000);
```

```
}
```

10.2

```
#include "DHT.h"
```

```
#define DHTPIN 4 // Digital pin connected to the DHT sensor
```

```
#define DHTTYPE DHT11 // DHT 11 DHT
```

```
dht(DHTPIN, DHTTYPE);
```

```
void setup() {
```

```
Serial.begin(9600);
```

```
Serial.println(F("DHTxx test!"));
```

```
dht.begin();
```

```
}
```

```
void loop() {
```

```
delay(2000);
```

```
float h = dht.readHumidity(); float
```

```
t = dht.readTemperature(); float f =
```

```
dht.readTemperature(true); if
```

```
(isnan(h) || isnan(t) || isnan(f)) {
```

```
Serial.println(F("Failed to read from DHT sensor!")); return;
```

```
}
```

```
float hif = dht.computeHeatIndex(f, h);
```

```
float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print("% Temperature: ");
Serial.print(t);
Serial.print(F("°C"));

Serial.print(f);
Serial.print(F("Heat index: "));
Serial.print(hic);
Serial.print(F("°C"));

Serial.print(hif);
Serial.println(F("°F"));

}
```

10.3

```
#include <Digital_Light_ISL29035.h>

#include <Digital_Light_TSL2561.h>

int sensorPin = 2; // select the input pin for the potentiometer int

ledPin = 15; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor int

suppl = 23;

void setup() {
pinMode(ledPin, OUTPUT);
pinMode(suppl,OUTPUT);
Serial.begin(115200);

digitalWrite(suppl,HIGH);

}

void loop() { sensorValue =
analogRead(sensorPin);
```

```

Serial.println(sensorValue); if
(sensorValue<1500)

{
    digitalWrite(ledPin, HIGH);
}
else
{
    digitalWrite(ledPin, LOW);
}

delay(1000);

}

```

OUTPUTS:

10.1:

```

10 cm
The distance to obstacles in front is:
3 inch
391 cm
The distance to obstacles in front is:
153 inch
3 cm
The distance to obstacles in front is:
204 inch
0 cm
The distance to obstacles in front is:
2 inch
3 cm
The distance to obstacles in front is:
76 inch
19 cm
The distance to obstacles in front is:
5 inch
5 cm
The distance to obstacles in front is:
77 inch
194 cm
The distance to obstacles in front is:
76 inch

```

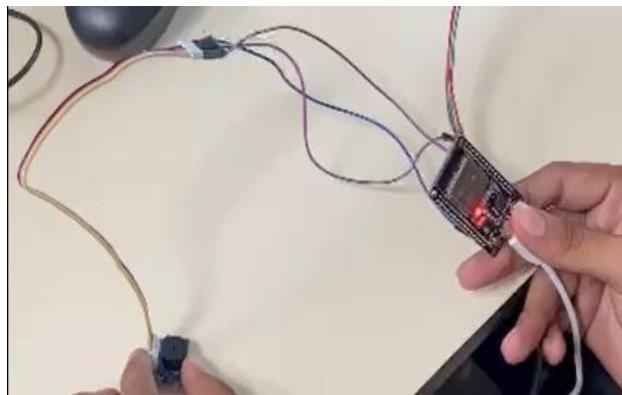
10.2:

```

Humidity: 58.00% Temperature: 30.30°C 86.54°F Heat index: 32.99°C 91.39°F
Humidity: 58.00% Temperature: 30.30°C 86.54°F Heat index: 32.99°C 91.39°F
Humidity: 58.00% Temperature: 30.20°C 86.36°F Heat index: 32.81°C 91.05°F
Humidity: 58.00% Temperature: 30.20°C 86.36°F Heat index: 32.81°C 91.05°F
Humidity: 58.00% Temperature: 30.10°C 86.18°F Heat index: 32.62°C 90.72°F
Humidity: 58.00% Temperature: 30.10°C 86.18°F Heat index: 32.62°C 90.72°F
Humidity: 58.00% Temperature: 30.10°C 86.18°F Heat index: 32.62°C 90.72°F
Humidity: 58.00% Temperature: 30.10°C 86.18°F Heat index: 32.62°C 90.72°F
Humidity: 58.00% Temperature: 30.00°C 86.00°F Heat index: 32.44°C 90.39°F
Humidity: 60.00% Temperature: 30.00°C 86.00°F Heat index: 32.83°C 91.10°F
Humidity: 61.00% Temperature: 30.10°C 86.18°F Heat index: 33.23°C 91.81°F
Humidity: 62.00% Temperature: 30.10°C 86.18°F Heat index: 33.44°C 92.19°F

```

10.3:



Experiment-11

Aim: Create app using MIT App Inventor

OBJECTIVES:

11.1. Tap to Speech App

11.2. Ball Bounce App

11.3. Digital doodle app

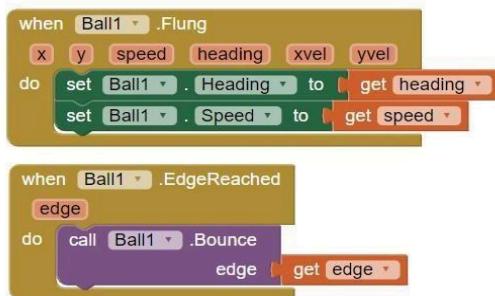
11.4. Bluetooth Client App

CODES:

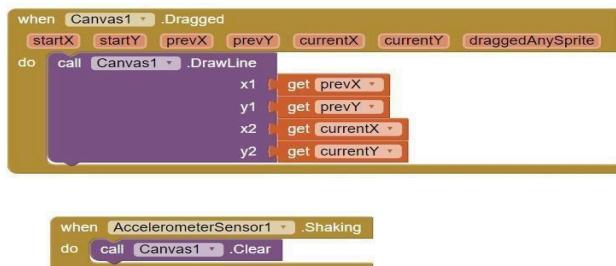
11.1:



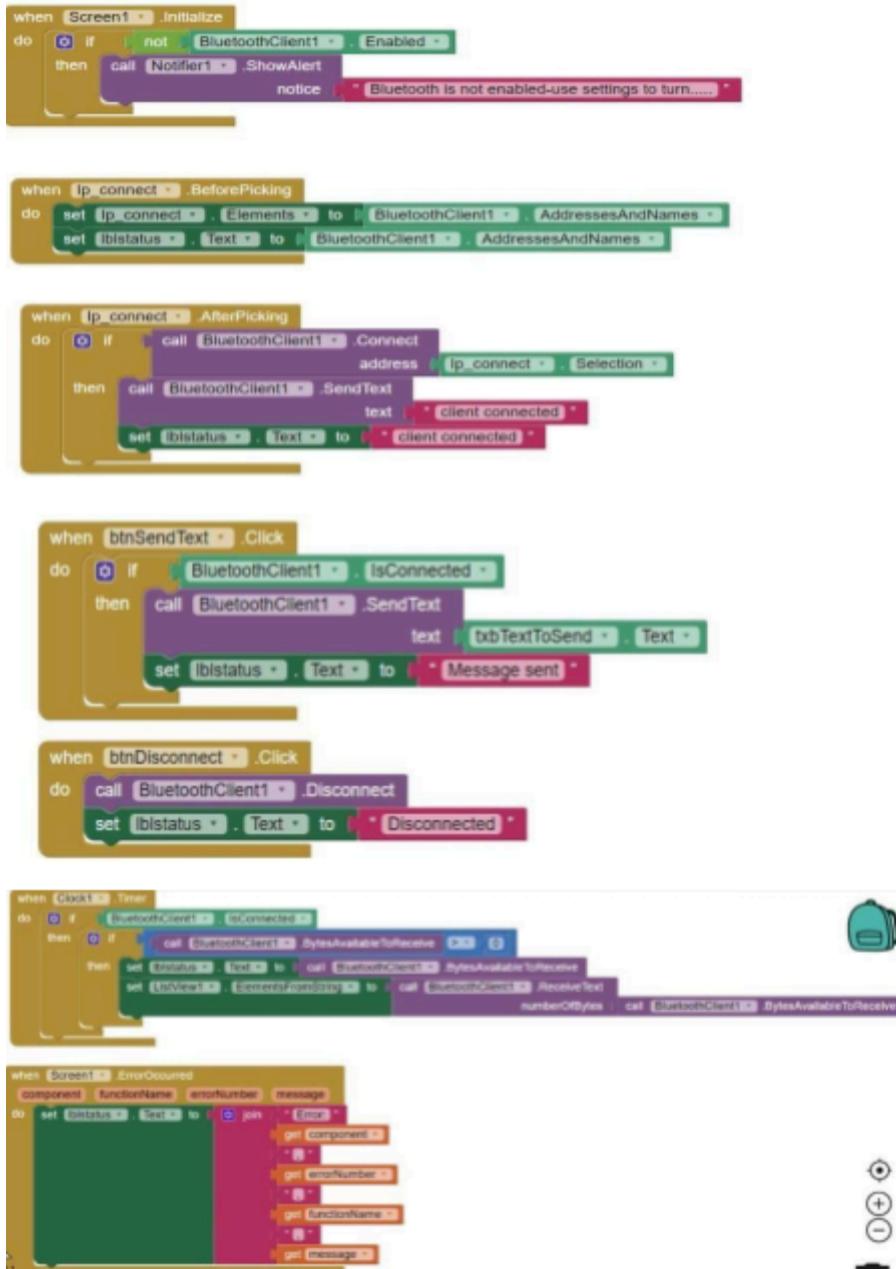
11.2:



11.3:



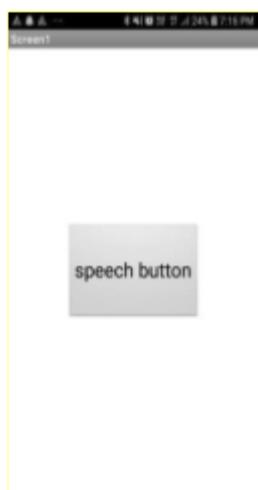
11.4:



OUTPUTS:

1)

Talk to Me

OUTPUTS:**1. Talk to Me****2. Ball Bounce****3. Digital Doodle App**

22'AI
042

A handwritten digital drawing in black ink on a white background. The text "22'AI" is written in a cursive style above a horizontal line. Below this line, the number "042" is written in a larger, more stylized cursive font, with a horizontal line underneath it.**3. Blutooth Client App**



Experiment-12

Aim: Learn to utilize Bluetooth Module on ESP32 and implement codes to.

12.1 Scan Bluetooth devices in range

12.2 Connect Bluetooth device and transfer data from and to ESP32 with APP

12.3 Control LED on ESP32 through Bluetooth app

OBJECTIVES:

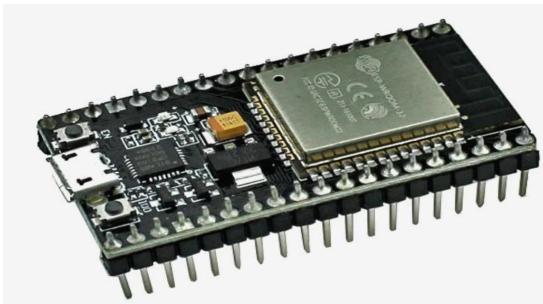
12.1: To Scan for all the Bluetooth Devices in Range

12.2: To Connect ESP32 with a Bluetooth Device and Transfer Data

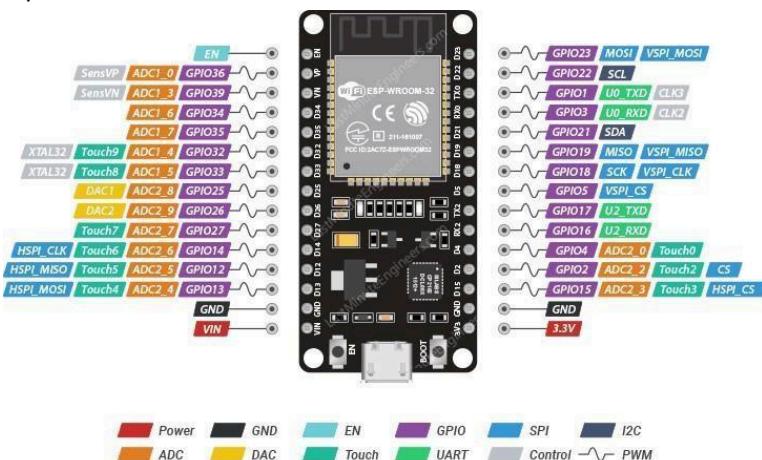
12.3.: Turn LED On/Off using Bluetooth App, Arduino and ESP-32

COMPONENTS:

1) ESP32:



The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities.





USB cable is use to connect the Computer to ESP32

3) LED:



A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it.

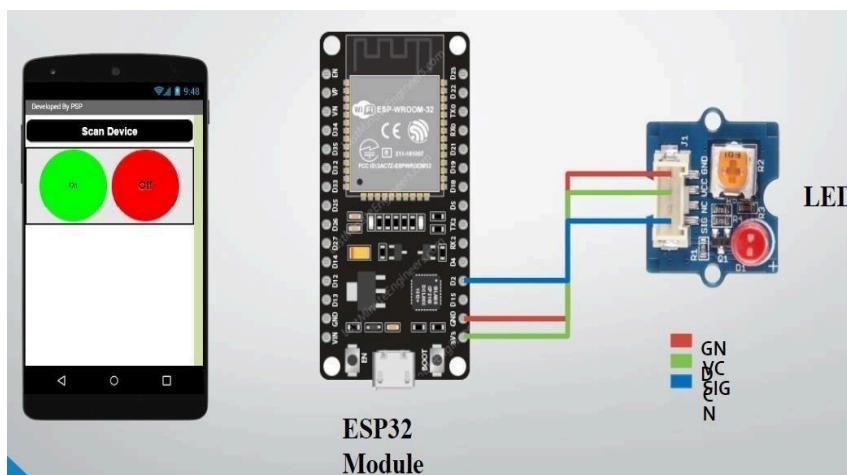
4) Jumper Wires:



Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins.

CONNECTION DIAGRAM:

12.3



CODES:

12.1

```
#include <BluetoothSerial.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it #endif

#if !defined(CONFIG_BT_SPP_ENABLED)
    #error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif
BluetoothSerial SerialBT;

#define BT_DISCOVER_TIME          10000

static bool btScanAsync = true;
static bool btScanSync = true;
void btAdvertisedDeviceFound(BTAdvertisedDevice* pDevice) {
    Serial.printf("Found a device asynchronously: %s\n", pDevice->toString().c_str());
}

void setup() {
    Serial.begin(115200);
    SerialBT.begin("ESP32test"); //Bluetooth device name
    Serial.println("The device started, now you can pair it with bluetooth!");

    if (btScanAsync) {
        Serial.print("Starting discoverAsync...");
        if (SerialBT.discoverAsync(btAdvertisedDeviceFound)) {
            Serial.println("Findings will be reported in \"btAdvertisedDeviceFound\"");
            delay(10000);
        }
        Serial.print("Stopping discoverAsync... ");
        SerialBT.discoverAsyncStop();
    }
}
```

```

Serial.println("stopped");

} else {

Serial.println("Error on discoverAsync f.e. not workin after a \"connect\"");

}

}

if (btScanSync) {

Serial.println("Starting discover...");

BTScanResults *pResults = SerialBT.discover(BT_DISCOVER_TIME); if

(pResults)

pResults->dump(&Serial);

else

Serial.println("Error on BT Scan, no result!");

}

}

void loop() {
delay(100);
}

```

12.2

```

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it #endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.

#endif

BluetoothSerial SerialBT;

void setup() {

```

```

Serial.begin(115200);

SerialBT.begin("ESP32test"); //Bluetooth device name

Serial.println("The device started, now you can pair it with bluetooth!"); }

void loop() { if
(Serial.available()) {

SerialBT.write(Serial.read());

}

if (SerialBT.available()) { BTData=SerialBT.read();

Serial.write(BTData);
}

delay(20);

}

```

12.3

```

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it #endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip. #endif
BluetoothSerial SerialBT;

const int led=2;

byte BTData;

void setup() {
pinMode(led,OUTPUT);
Serial.begin(115200);
SerialBT.begin("ESP32test"); //Bluetooth device name
Serial.println("The device started, now you can pair it with bluetooth!"); }

```

```

void loop() { if
(Serial.available()) {

SerialBT.write(Serial.read());}

}

if (SerialBT.available()) { BTData=SerialBT.read();}

Serial.write(BTData);

}

if(BTData=='1')

{

digitalWrite(led,HIGH);

}

else if(BTData=='0')

{

digitalWrite(led,LOW);

}

delay(20);

}

```

OUTPUTS:

12.1:

```

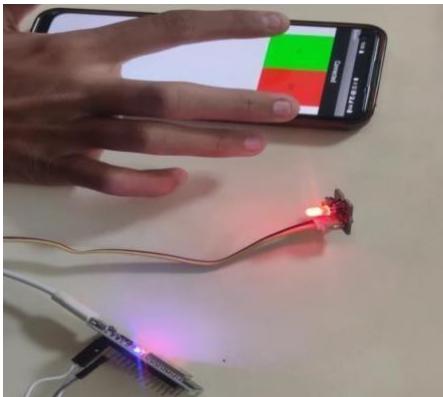
Found a device asynchronously: Name: OPPO F11, Address: a4:fc:9:39:50:25, cod: 58987, rssi: -66
Found a device asynchronously: Name: DENGTOP-HN3AVDE, Address: c0:e4:34:5d:90:0a, cod: 27527, rssi: -93
Found a device asynchronously: Name: Vandit Kalathiya, Address: 22:22:53:cd:54:5d, cod: 58987, rssi: -92
Found a device asynchronously: Name: Vandit Kalathiya, Address: 22:22:53:cd:54:5d, cod: 58987, rssi: -92
Name: realme 9 5G, Address: c4:df:39:5c:1e:ce, cod: 58987, rssi: -91
Found a device asynchronously: Name: , Address: e0:9e:b4:45:08:70, cod: 27527, rssi: -96
Found a device asynchronously: Name: , Address: 7c:6b:9c:2d:0a:f0, cod: 58987, rssi: -87
Found a device asynchronously: Name: ESP32test, Address: ec:62:60:04:38:4a, cod: 18049, rssi: -76
Stopping discoverAsync... stopped
Starting discover...
>> Dump finished - 13
- 1: Name: ESP32test, Address: 0c:b8:15:f3:b9:86, cod: 18049, rssi: -87
- 2: Name: Vandit Kalathiya, Address: 22:22:53:cd:54:5d, cod: 58987, rssi: -75
- 3: Name: realme X2 Pro, Address: 44:46:07:0c:90:28, cod: 58987, rssi: -57
- 4: Name: OnePlus Nord 2T 5G, Address: 49:74:12:1e:91:73:02, cod: 58987, rssi: -72
- 5: Name: Galaxy A32, Address: 60:4a:e9:9b:h2:19, cod: 58987, rssi: -04
- 6: Name: oppo reno 7 5g, Address: 74:ef:4b:eb:h0:46, cod: 58987, rssi: -60
- 7: Name: , Address: 7c:6b:9c:2d:0a:f0, cod: 58987, rssi: -95
- 8: Name: OPPO F11, Address: a4:c9:39:50:25:00, cod: 58987, rssi: -64
- 9: Name: OPPO A73, Address: d0:20:ba:0a:ab:a0, cod: 58987, rssi: -74
- 10: Name: realme 50 5G, Address: cd:df:39:4f:f1:f4, cod: 58987, rssi: -70
- 11: Name: realme 6, Address: d0:20:ba:0a:ab:a0, cod: 58987, rssi: -63
- 12: Name: ESP32test, Address: ec:62:60:04:38:4a, cod: 18049, rssi: -73
- 13: Name: ESP32-G06, Address: ec:62:60:9b:86:f6, cod: 18049, rssi: -91
-- Dump finished --

```

12.2:

```
COM4
=====
hii
Hello
Hello how are you
```

12.3:



Experiment-13

Aim: Learn to utilize Wi-Fi Module on ESP32 and implement codes to

13.1 Scan Wi-Fi Networks

13.2 Set up a simple Wi-Fi web server to blink an LED from the web

13.3 Set up a Wi-Fi access point and provide a web server on it

OBJECTIVES:

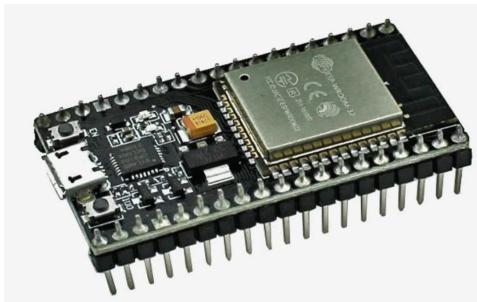
13.1: To Scan for all the Wi-Fi Networks in Range

13.2: To Connect ESP32 with a Wi-Fi Network and use it to control the LED

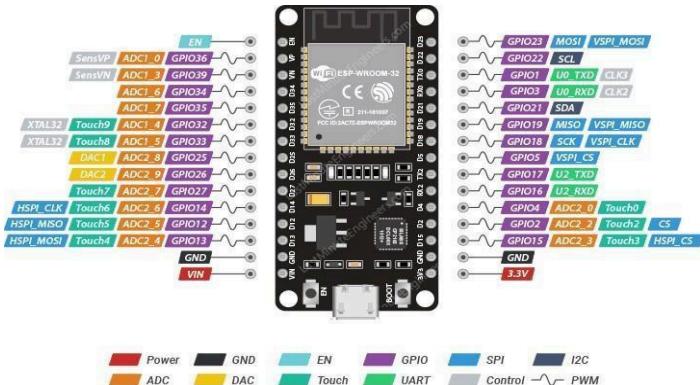
13.3.: To Set-up ESP32 as Web Server and then using Wi-Fi Access Point to control the LED

COMPONENTS:

1) ESP32:



The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities.



2) USB CABLE:



USB cable is used to connect the Computer to ESP32

3) LED:



A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it.

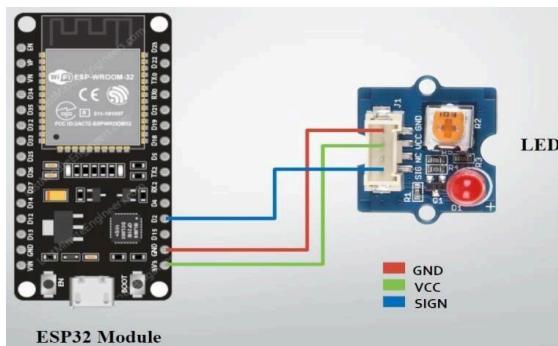
4) Jumper Wires:



Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins.

CONNECTION DIAGRAM:

a) 13.2 and 13.3



CODES:

13.1

```
#include "WiFi.h"

void setup()
{
    Serial.begin(115200);

    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);

    Serial.println("Setup done");
}

void loop(){
    Serial.println("scan start");

    int n = WiFi.scanNetworks();
    Serial.println("scan done");    if
(n == 0) {
        Serial.println("no networks found");
    } else {
        Serial.print(n);
        Serial.println(" networks found");    for
(int i = 0; i < n; ++i) {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));

```

```
    Serial.print(")");
    Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)? ":"*");      delay(10);
}
Serial.println("");  delay(5000);
}
```

13.2

```
#include <WiFi.h>
```

```
const char* ssid    = "SpyDer";
const char* password = "123456789";
WiFiServer server(80);
```

```
void setup()
```

```
{
    Serial.begin(115200);
    pinMode(2, OUTPUT); // set the LED pin mode
```

```
delay(10);
```

```
Serial.println();
```

```
Serial.println();
```

```
Serial.print("Connecting to ");
```

```
Serial.println(ssid);
```

```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {      delay(500);
Serial.print(".");
}
```

```
Serial.println("");
```

```

Serial.println("WiFi connected.");
Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
server.begin();

}

int value = 0;

void loop(){
    WiFiClient client = server.available(); // listen for incoming clients

    if (client) { // if you get a client,
        Serial.println("New Client."); // print a message out the serial port
        String currentLine = ""; // make a String to hold incoming data from the client
        while (client.connected()) { // loop while the client's connected
            if (client.available()) {
                // if there's bytes to read from the client,
                char c = client.read(); // read a byte, then
                Serial.write(c); // print it out the serial monitor
                if (c == '\n') { // if the byte is a newline character
                    if (currentLine.length() == 0) { // if
                        client.println("HTTP/1.1 200 OK");
                        client.println("Content-type:text/html");
                        client.println();
                        client.print("Click <a href=\"/H\">here</a> to turn the LED on pin 2 on.<br>");
                        client.print("Click <a href=\"/L\">here</a> to turn the LED on pin 2 off.<br>");

                        client.println();
                    }
                }
            }
            else { // if you got a newline, then clear currentLine:
                currentLine = "";
            }
            else if (c != '\r') { // if you got anything else but a carriage return character,
                currentLine += c; // add it to the end of the currentLine
            }
        }
        // Check to see if the client request was "GET /H" or "GET /L": if
        (currentLine.endsWith("GET /H")) {
            digitalWrite(2, HIGH); // GET /H turns the LED on
        }
    }
}

```

```
    }

    if (currentLine.endsWith("GET /L")) {      digitalWrite(2, LOW);
// GET /L turns the LED off

    }

}

}

// close the connection:

client.stop();

Serial.println("Client Disconnected.");

}

}
```

13.3

```
#include <WiFi.h>

#include <WiFiClient.h>

#include <WiFiAP.h>

#define LED_BUILTIN 2 // Led Signal Pin

const char *ssid = "Group5";

const char *password = "12345678";

WiFiServer server(80);
```

```
void setup() {

pinMode(LED_BUILTIN, OUTPUT);

Serial.begin(115200);

Serial.println();

Serial.println("Configuring access point...");
```

```
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: "); Serial.println(myIP);
server.begin();

Serial.println("Server started");
}

void loop(){
WiFiClient client = server.available(); // listen for incoming clients
if (client) { // if you get a client,
Serial.println("New Client."); // print a message out the serial port
String currentLine = ""; // make a String to hold incoming data from the client
while (client.connected()) { // loop while the client's connected if (client.available()) {
// if there's bytes to read from the client, char c = client.read(); // read a byte, then
Serial.write(c); // print it out the serial monitor if (c ==
'\n') { // if the byte is a newline character

if (currentLine.length() == 0) {
client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html"); client.println();

client.print("Click <a href=\"/H\">here</a> to turn ON the LED.<br>");
client.print("Click <a href=\"/L\">here</a> to turn OFF the LED.<br>");

client.println();
break;
} else { // if you got a newline, then clear currentLine:
currentLine = "";
}
} else if (c != '\r') { // if you got anything else but a carriage return character,
currentLine += c; // add it to the end of the currentLine
}

if (currentLine.endsWith("GET /H")) { digitalWrite(LED_BUILTIN, HIGH);
// GET /H turns the LED on
}
}
```

```

        }

        if (currentLine.endsWith("GET /L")) {      digitalWrite(LED_BUILTIN, LOW);
// GET /L turns the LED off

    }

}

}

// close the connection:

client.stop();

Serial.println("Client Disconnected.");

}

}

```

OUTPUTS:

13.1:

```

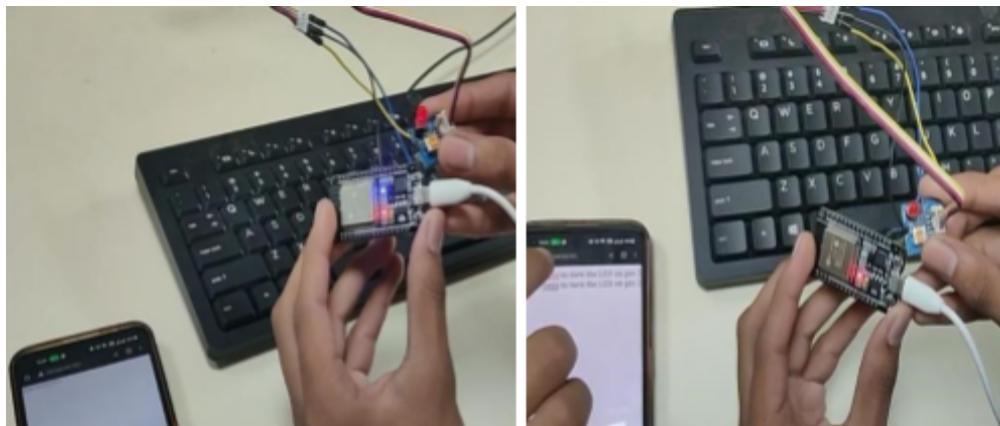
4: FYECLAB2-03 8302 (-74) *
5: OPPO A76 (-75) *
6: WINCCELL (-81) *
7: CHARUSAT Wi-Fi (-83) *
8: windowsucks (-86) *
9: Solar (-87) *
10: Shruti (-87) *
11: Hinal (-88) *
12: JioPrivateNet (-90) *
13: JioNet@CharotarUniversity (-92) *
14: OnePlus (-93) *

scan start
scan done
14 networks found
1: vivo 1920 (-52) *
2: vivo 1901 (-64) *
3: Maru net vaparu em... (-67) *
4: FYECLAB2-03 8302 (-72) *
5: OPPO A76 (-73) *
6: Redmi Note 10 Pro (-76) *
7: oppo reno 7 5G (-81) *
8: WINCell (-87) *
9: CHARUSAT Wi-Fi (-89) *
10: Hinal (-89) *
11: Solar (-92) *
12: Solar (-92) *
13: JioNet@CharotarUniversity (-95) *
14: JioPrivateNet (-96) *

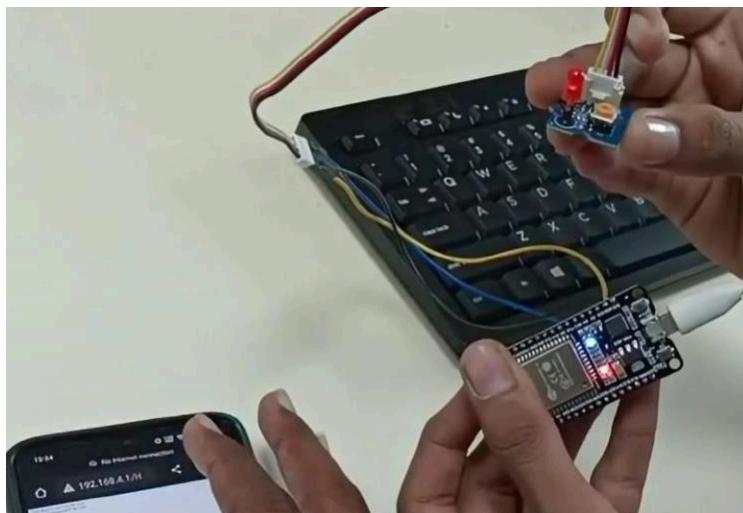
scan start
scan done
10 networks found
1: vivo 1920 (-53) *
2: Redmi Note 10 Pro (-59) *
3: Hinal (-65) *
4: OPPO A76 (-67) *
5: Maru net vaparu em... (-69) *
6: FYECLAB2-03 8302 (-70) *
7: oppo reno 7 5G (-76) *
8: WINCell (-80) *
9: CHARUSAT Wi-Fi (-81) *
10: Hinal (-96) *

```

13.2:



13.3:





C Language Fundamentals

Welcome to our exploration of the fundamental concepts of the C programming language. In this presentation, we'll dive into the core building blocks that form the foundation of C, paving the way for more advanced programming skills.

Introduction to C Programming

History and Applications

C was developed in the 1970s and has become a widely used language for system programming, embedded systems, and more.

Key Features

C is known for its efficiency, portability, and low-level control, making it a popular choice for a variety of projects.

Development Workflow

The C programming workflow involves writing code, compiling it, and then executing the resulting program.

Data Types and Variables

1 Primitive Data Types

C supports a range of data types such as integers, floating-point numbers, characters, and booleans.

2 Variable Declarations

Variables are used to store data, and they must be declared with a specific data type.

3 Memory Allocation

The size and range of each data type determine how much memory is allocated for storing the variable's value.



Operators and Expressions

Arithmetic Operators

C provides a set of arithmetic operators, including addition, subtraction, multiplication, and division, for performing mathematical operations.

Logical Operators

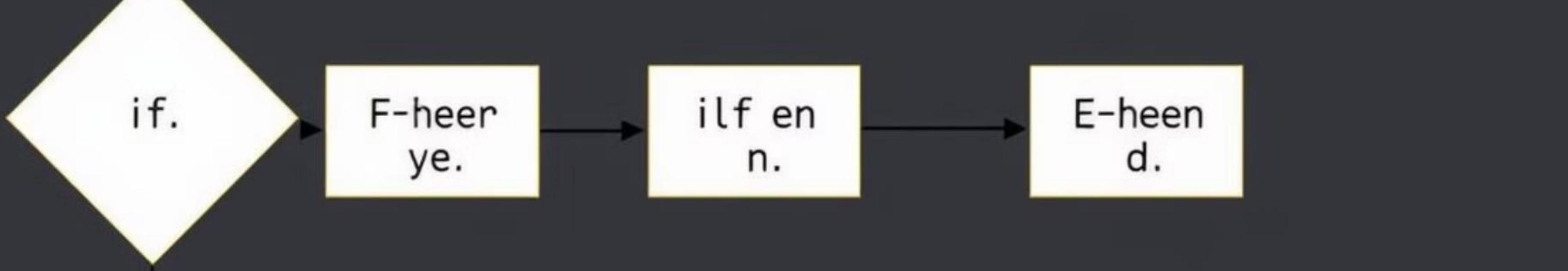
Logical operators, such as AND, OR, and NOT, are used to combine and evaluate boolean expressions.

Expressions and Precedence

Expressions in C can be built using various operators, and the order of evaluation is determined by operator precedence rules.

(+ 2 - 2 + + 3 = 5] xx + 3 + 5] = 5 +
(+ 1 - 1 4 - 3 = 5 [++ 4 + 3 = 1 +
++ [5 - 5 + 1 5 = 15] = + 3 =] = 0 =
(+ 1 + 5 4 + 4 = 5 xx + 1 + 4 = 3 +

(0 + 7 = 1 2 + 3 = 26 + 3 x= 11 = =
+ - + 1 + - 2 + - [x - 7 + 5 = 3]
[+ 1 + + 2 ↳ 10 + 6.- xe. 3.] + 5 = + 1
+ 3 + 4 + + 3] = [1f + 10 + 5 = 3.
+ 4 + 2 + 1 3 = 6.1y] + 3] + 2 = 2.
(+ 3 + 2 +] 1 + = [.56.1] + 3 1 x 1 -
x + + - 7 +] 5 = = + y - 13 = = 3 +
+ (+ + + 6 + - x) + = [xx + 114 = 13.3.
x - + - + + - (1 + x + xx e. 211 = 11 5.
+ + 1 + 1 + 1 4 = 5 16 + 3 = = 1 +
: + 6 + 4 + 1 1 = 1] ‡. 14 6 = = 1.1



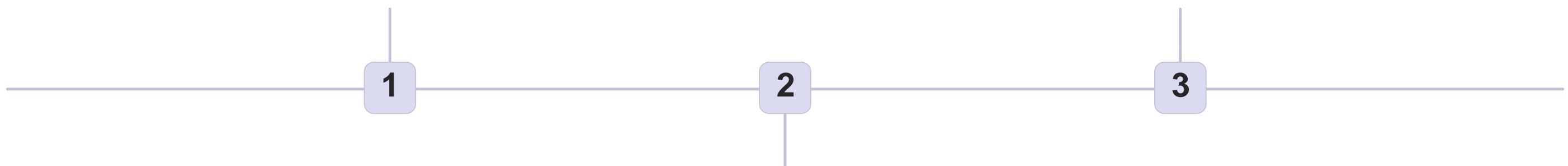
Control Structures

Conditional Statements

C provides conditional statements, such as if-else and switch, to make decisions based on specific conditions.

Branching and Jumping

C also supports branching statements, such as break and continue, to control the flow of execution within loops.



Loops

Loops, including while, do-while, and for, allow you to repeatedly execute a block of code until a certain condition is met.

Functions and Scope



Function Definitions

Functions in C allow you to encapsulate and reuse blocks of code, promoting modularity and code organization.



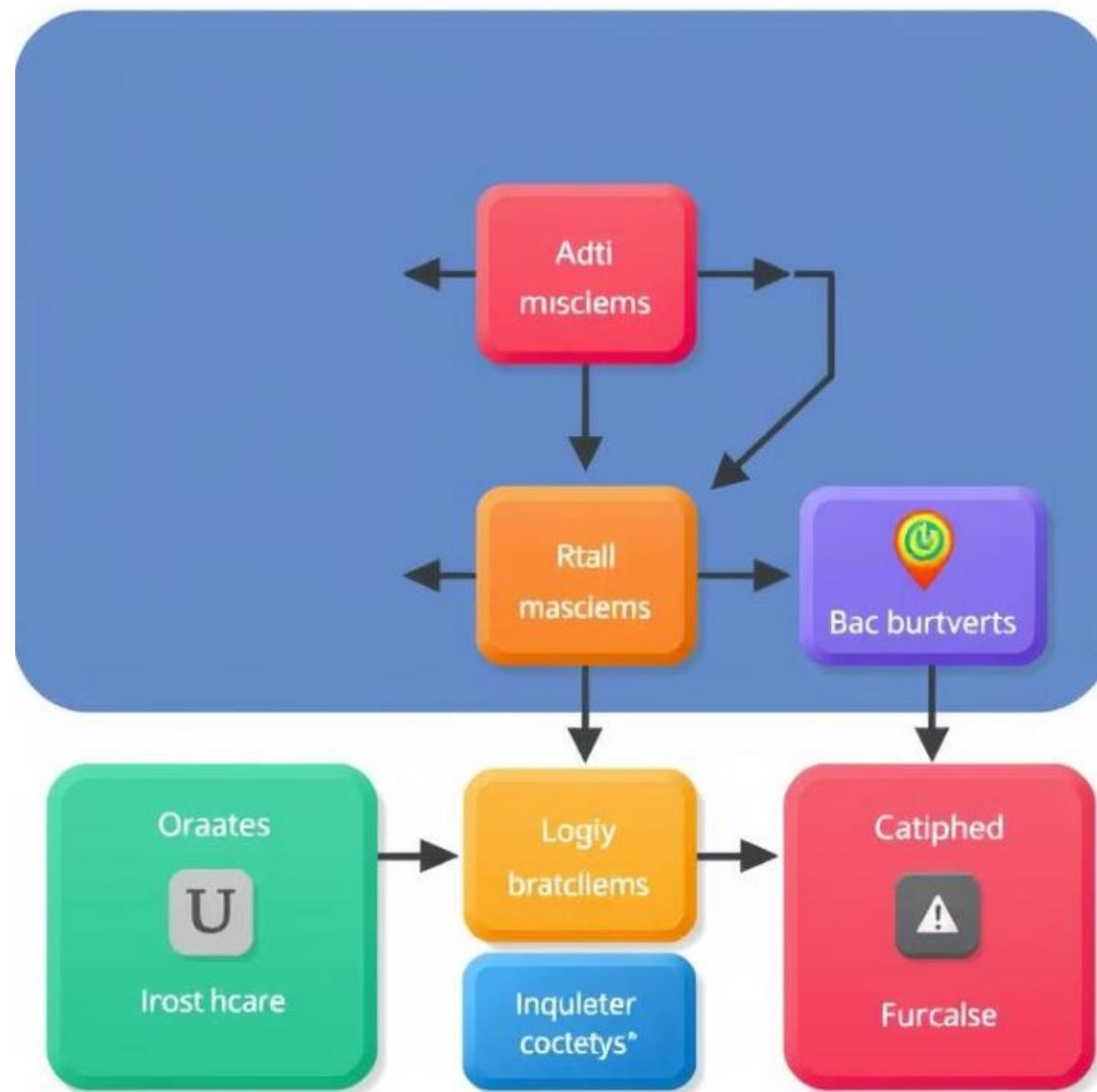
Variable Scope

The scope of variables, whether local or global, determines where they can be accessed and modified within a C program.

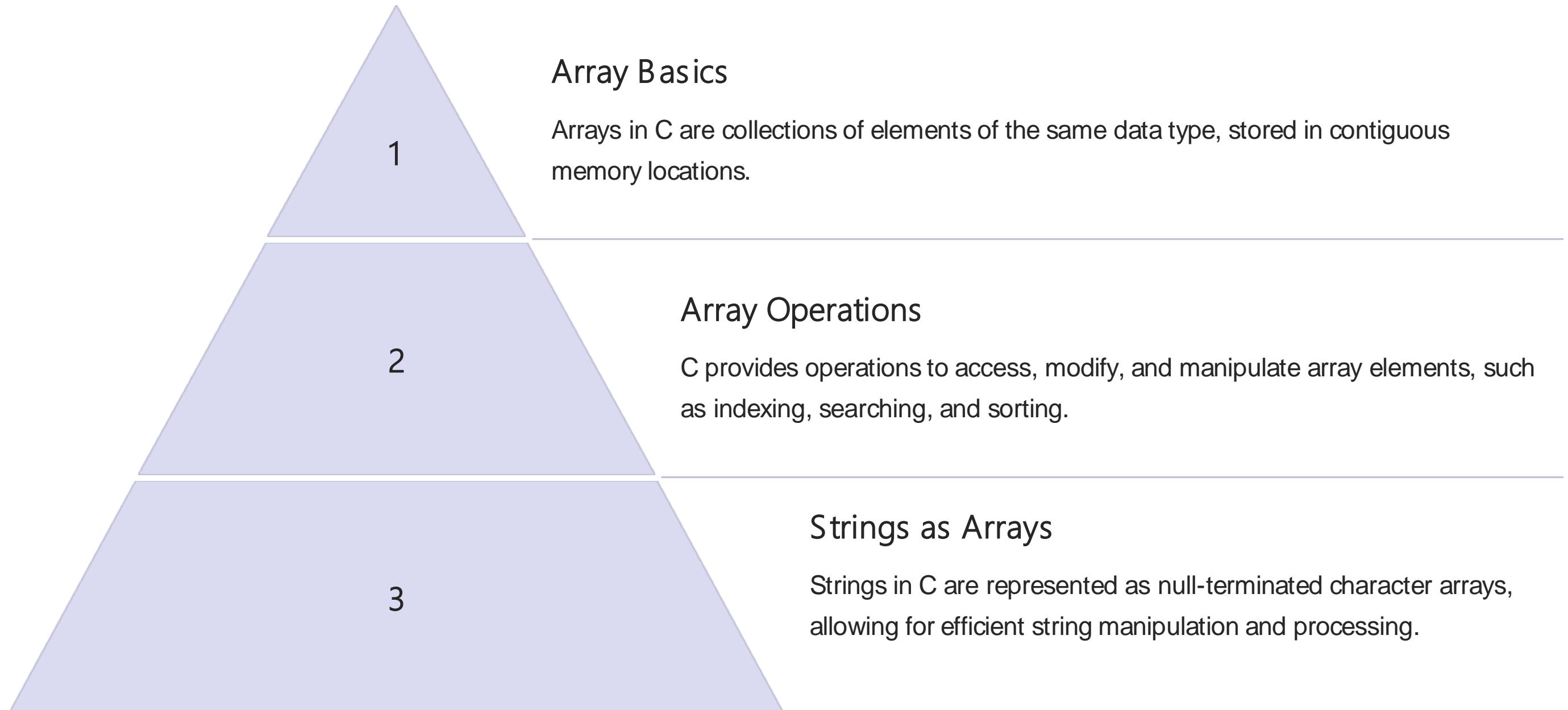


Return Values

Functions can return values, allowing them to provide computed results or communicate with the calling code.



Arrays and Strings



Pointers and Memory Management

Pointer Basics

1

Pointers in C are variables that store memory addresses, allowing direct access and manipulation of data in memory.

Dynamic Memory Allocation

2

C provides functions like malloc and free to dynamically allocate and deallocate memory at runtime.

Memory Leaks

3

Proper memory management is crucial to avoid memory leaks, which can lead to program instability and crashes.