PART A
CLOUD COMPUTING

EX: NO : 1 **PROGRAM ON SaaS TO CREATE WORD DOCUMENT**

To implement program on SaaS to Create an word document of your class time table and store locally and on cloud with doc and pdf format

AIM:

To implement program on SaaS to Create an word document of your class time table and store locally and on cloud with doc and pdf format.(use www.zoho.com and docs.google.com)

PROCEDURE:

1.With Google Docs, you can create and edit text documents right in your web browser—no special software is required. Even better, multiple people can work at the same time, you can see people's changes as they make them, and every change is saved automatically.

2.To start, you need a document to work with. In this section,

you learn how to:

- Create a new document
- Import and convert old documents to Docs

Create a new document

(i)You can create a new document right in Docs or in Google Drive. In Docs,

click Create new document.

(ii)In Drive, click New > Google Docs > Blank document or From a template.

Import and convert old documents to Docs

(iii)If you have existing text documents, such as Microsoft Word or Adobe PDF files, you can import and convert them to Docs.

- Go to Drive.
- Click New > File Upload and choose a text document from your computer. Supported files include .doc, .docx, .dot, .html, plain text (.txt), .odt, and .rtf.

- Right-click the file you want to convert and select Open with > Google Docs.Converting your document from another program creates a copy of your original file in Docs format. Youcan then edit it in your browser like any other document.

Create Class timetable

Share documents

1. Open the file you want to share.
2. Click share.
3. Enter the email addresses or Google Groups you want to share with.

Note: If you can't add people outside your company, see your G Suite administrator.

4. Choose what kind of access you want to grant people:
   - **Can edit**—Collaborators can add and edit content as well as add comments.
   - **Can comment**—Collaborators can add comments, but not edit content.
   - **Can view**—People can view the file, but not edit or add comments.

   Click Send.

Everyone you shared the document with receives an email with a link to the document.

RESULT:

Thus the word document is created, stored and accessed using cloud.

EX NO:2 PROGRAM ON SAAS TO CREATE SPREADSHEET

To implement program on SaaS to Create a spread sheet to generate a mark sheet for student progress report.

AIM:

To implement program on SaaS to Create a spread sheet to generate a mark sheet for student progress report.

PROCEDURE:

If you're accustomed to creating your spreadsheets using an office suite or software likeMicrosoft Excel, you won't have any issue in creating a Google Spreadsheet. Google Spreadsheet works the same as Excel, and you can do most of the important spreadsheet tasks with it. You can

use Google Spreadsheet directly from your web browser or from its mobile app.

1. Sign into Google Sheets. Visit docs.google.com/spreadsheets and sign in with your Google or Gmail account. Your Gmail account gives you free access to Google Sheets.

2. View your existing sheets. Upon logging in, you will be brought to the main directory. If you already have existing spreadsheets, you can see and access them from here.

3. Create a new spreadsheet. Click the large red circle with a plus sign on the lower right corner. A new window or tab will be opened with the web-based spreadsheet.

4. Name the spreadsheet. “Untitled spreadsheet” appears on the top left corner. This is the current name of the spreadsheet. Click on it, and a small window will appear. Type in the name of the spreadsheet here, and click the “OK” button. You will see the name immediately change.

RESULT:

Thus the Program on saas is implemented by creating spreadsheet and details are entered successfully.

Ex.No.3: BLOGSPOT CREATION

To implement web services by create your BlogSpot and Collaborating via Wikis

AIM :

To implement web services by create your BlogSpot and Collaborating via Wikis

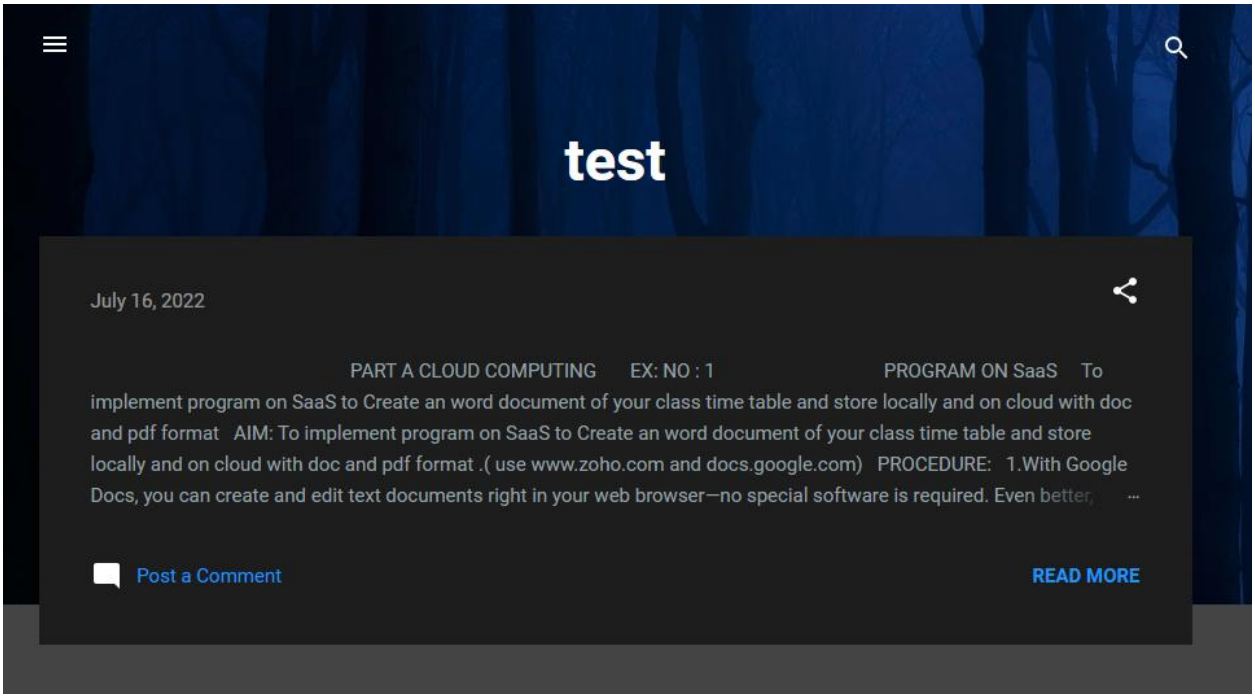
PROCEDURE:

CREATE A BLOG:

1.Search in google as “ blogger”, there are many websites to create blogs.Let us take www.blogger.com.
2.Click on create blog . Sign in to your google account
3.Choose a name for your blog and click on next.
4.Create the blog name click on next.
5.Type the display name and click on Finish.
6.click on create post and update your post in the displayed file and publish.
7.You can apply theme and create many post and labels in your blog and customize it.

8.Finally click on view blog from the menu and you can view the blog which you have created.
9.In the blog site ,you can share on your own blog in any social medias and wikkis.

OUTPUT:



RESULT:

Thus the log is created and collaborated successfully.

EX NO: 4 GOOGLE APP ENGINE

To implement on PaaS to Install Google App Engine, create a program to validate user; create a database login(username, password)in mysql and deploy to cloud

PROCEDURE:

- Create a Compute Engine instance

- Install MySQL
- Connect to MySQL

Create a Compute Engine instance

Create a new project in the Google Cloud console. You can use an existing project, but creating a new project makes cleanup easier.

You can complete all of the steps in this document using the Google Cloud console, but if you prefer to use the gcloud CLI, follow these steps to enable the Compute Engine API and install the Google Cloud CLI.

- Use the Google Cloud console to enable the Compute Engine API.
- Install the gcloud CLI.
- Configure your workspace to make commands less verbose. Substitute your project's values for *PROJECT_ID* and *ZONE* in the following commands. For the full list of zones, see Available regions & zones.

  gcloud config set project *PROJECT_ID*
  gcloud config set compute/zone *ZONE*

Create a Compute Engine instance for MySQL and establish an SSH connection to the newly created instance. The default operating system is Debian version 10. If you prefer to use a different operating system for this tutorial, you can choose from the options described on the public images page in the Compute Engine documentation.

To create a Compute Engine instance in the Google Cloud console:

1. Open the Google Cloud console.
2. Select your newly created project and click **Continue**.
3. Click **Create instance** (**New instance** if you have existing instances). Name the instance **mysql-test**.
4. To specify an operating system other than the default value, in the **Boot disk** section, click **Change** to configure the properties for the boot disk. In the **Public images** tab, select an operating system and then click **Save**.
5. Click **Create**.

To establish an SSH connection:

1. On the **VM instances** page, find your new VM instance in the list.
2. In the **Connect** column, click **SSH**. The SSH terminal opens in a browser window.

Install MySQL

The following steps describe how to install MySQL on your Compute Engine instance.

Versions 10 and later of Debian contain MariaDB instead of MySQL as part of its package management system. MariaDB maintains compatilibity with the MySQL protocol, but has an independently evolving feature set. For more details, see MariaDB vs. MySQL.

To install MySQL, download the release package and manually install using the dpkg commmand.

1. Install the wget dependency.

```
sudo apt-get install -y wget
```

2. Download the MySQL Community Server release package.

```
export DEB_FILE=mysql-apt-config_0.8.20-1_all.deb
cd /tmp
curl -L --output ${DEB_FILE} \
https://dev.mysql.com/get/${DEB_FILE}
```

3. Verify the integrity of the release package file.

```
cat > ${DEB_FILE}.md5 << EOL
799bb0aefb93d30564fa47fc5d089aeb ${DEB_FILE}
EOL
md5sum --check ${DEB_FILE}.md5
```

4. The authenticity and integrity of the file are verified if you see the following output.

```
mysql-apt-config_0.8.20-1_all.deb: OK
```

5.After you have verified the file, add the MySQL package to the local package repository.

```
sudo dpkg -i ${DEB_FILE}
```

6.With the top **MySQL Server & Cluster** menu option selected, press **Return** and then use the arrow keys to choose a server version.

7.This guide expects you to choose either MySQL 8.0 or 5.7. Press **Return** on your keyboard after you have selected the version.

8.When you are satisfied with the options selected in the configuration menu, use the arrow keys to select Ok in the menu and press **Return** on your keyboard.

9.Update the package cache.

sudo apt-get update

10.Install MySQL. The installation process starts the MySQL service for you.

```
sudo apt-get -y install mysql-community-server
```

11.You are prompted to provide some details for the installation such as the root password.

Connect to MySQL

1. Connect to MySQL using the MySQL client.

sudo mysql -u root -p

When you connect to MySQL, the prompt changes to mysql>.

You can then run MySQL commands. For example, the following command shows the threads running, including the current connection.

```
Mysql> SHOW processlist;
+----+-----------------+-----------+------+---------+------+-----------------
------+------------------+
| Id | User            | Host      | db   | Command | Time | State
| Info             |
+----+-----------------+-----------+------+---------+------+-----------------
------+------------------+
|  5 | event_scheduler | localhost | NULL | Daemon  | 1889 | Waiting on empty
queue | NULL             |
| 14 | root            | localhost | NULL | Query   |    0 | init
| show processlist |
+----+-----------------+-----------+------+---------+------+-----------------
------+------------------+
2 rows in set (0.00 sec)
```

You can use the following command to generate a list of users.

```
Mysql> SELECT User, Host, authentication_string FROM mysql.user;

+------------------+-----------+-------------------------------------------
-------------------------+
| User             | Host      | authentication_string
|
+------------------+-----------+-------------------------------------------
-------------------------+
| mysql.infoschema | localhost |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.session    | localhost |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| mysql.sys        | localhost |
$A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| root             | localhost |
$A$005$,BS{G+*#cVYxb6x40q0aFS5dp2/Kz6u2vennR5qe0eBKVA/6VW5B            |
```

```
+-----------------+----------+-----------------------------------------
-------------------------+
4 rows in set (0.00 sec)
```

When you are done running commands, use the `exit` command to quit out of the MySQL client, and then use `exit` again to sign out of the Compute Engine instance.

Mysql>exit.

RESULT:

Thus mysql is installed successfully in google compute engine using paas serives.

EX NO: 5 LINUX INSTALLATION USING VIRTUAL BOX

Install Virtual box / VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

AIM :

To install oracle virtual box and linux OS in the virtual box on the top of windows OS.

PROCEDURE:

1.Download the oracle virtual box from google and install in the windows os .

2.Open oracle virtual box and click on new.

3. Write your system Name for example Linux Mint 15.

4. Select Type: Linux

5. Select Version: Ubuntu.

6.select the RAM size.

7.In hard disk setting , Choose **Create a virtual hard drive now**, to make a virtual disk space. Then select the **VDI** to make a backup. Choose **Dynamically allocated,** Now select the amount of hard drive size.

8.Choose Linux ISO file, Click **Start** to launch systemChoose your system iso file from your computer.for example my system iso file is (**linuxmint-15-cinnamon-dvd-32bit.iso**).

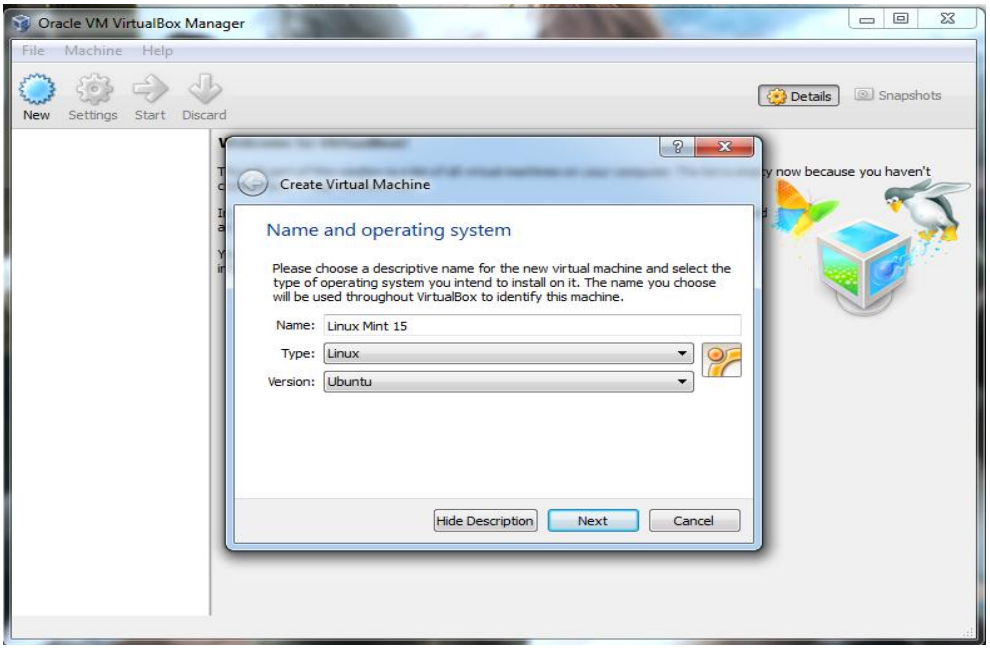
9.Install linux and make account. And select **Erase disk and install Linux Mint.**Then press **Install Now**. Now make your account,Then press **Continue.**

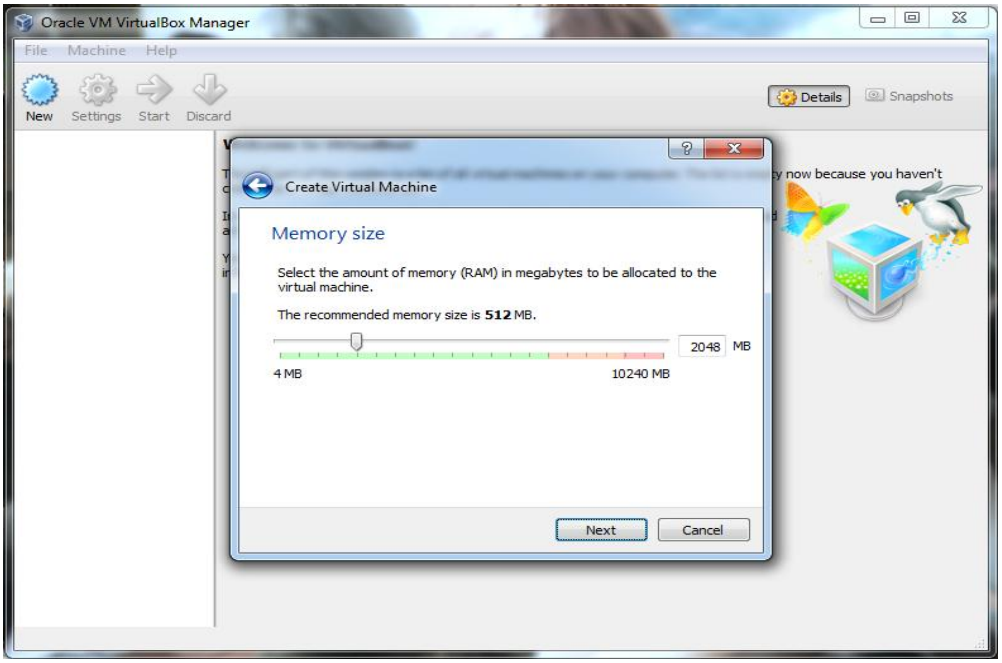
10.Now the linux Os is successfully installed in the oracle virtual box.

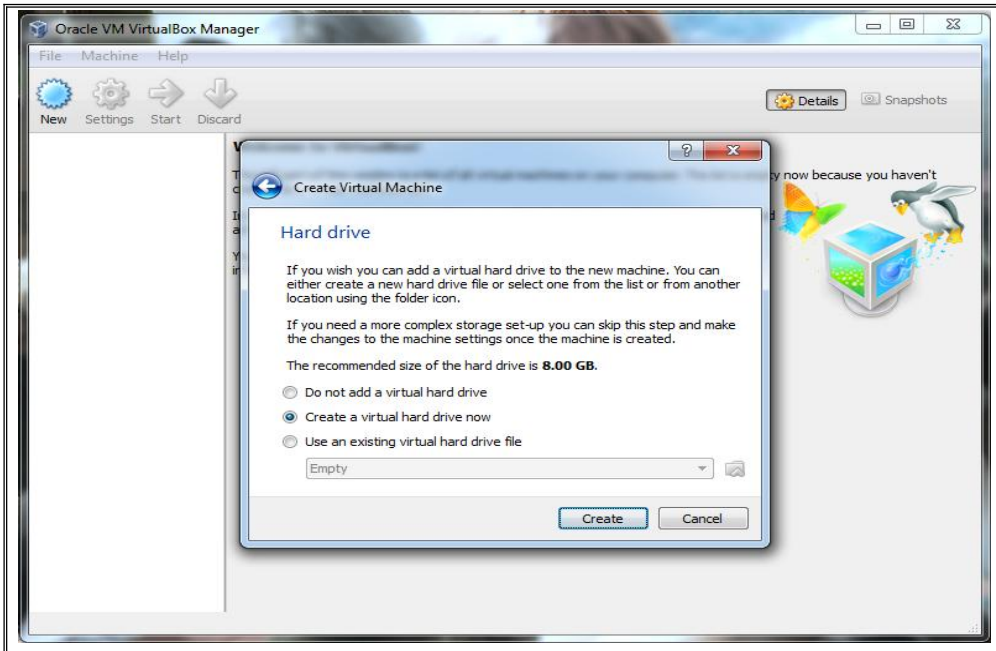
Result :

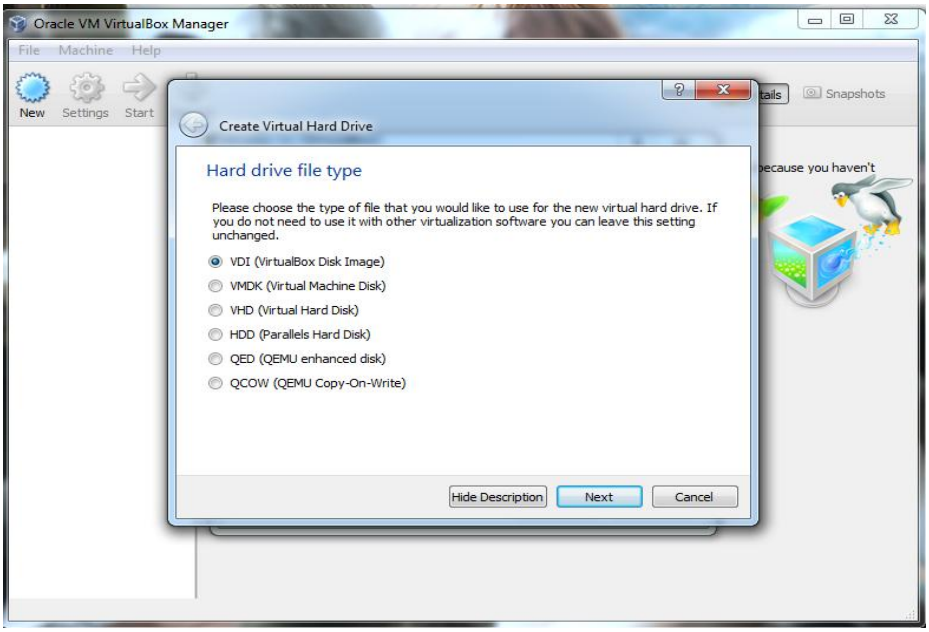
Thus the linux os is installed in the virtual box successfully.

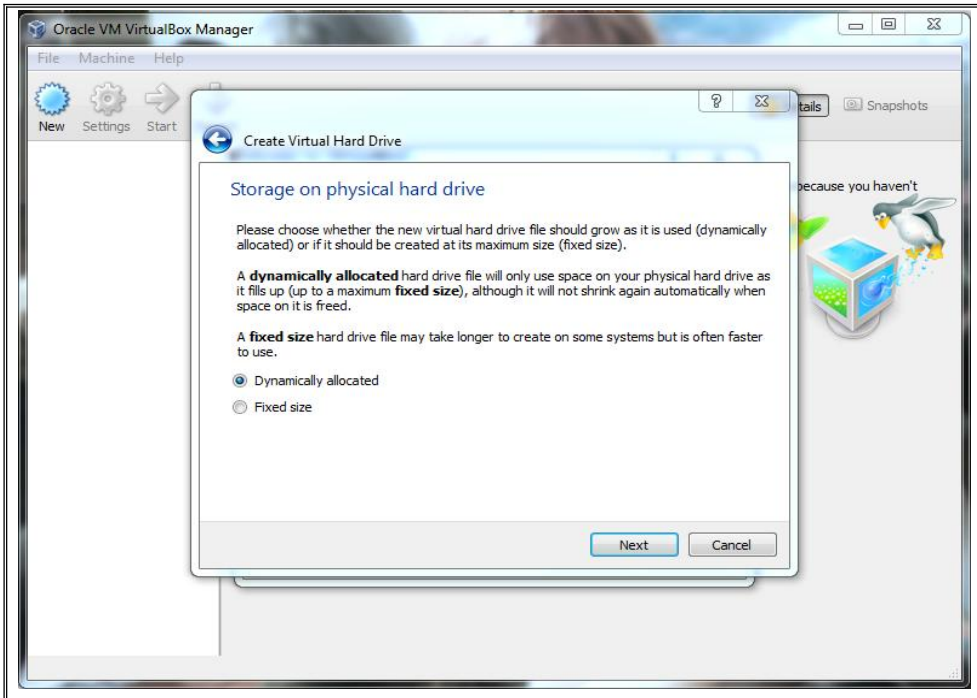
Output:

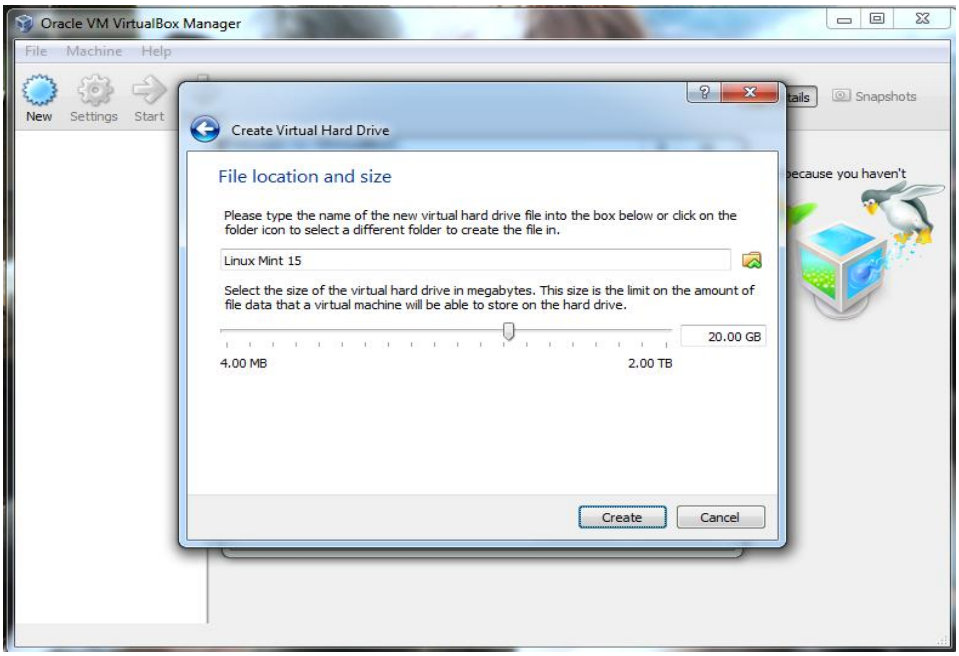

Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Details Snapshots
Create Virtual Machine
Name and operating system
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.
Name: Linux Mint 15
Type: Linux
Version: Ubuntu
Hide Description
Next
Cancel

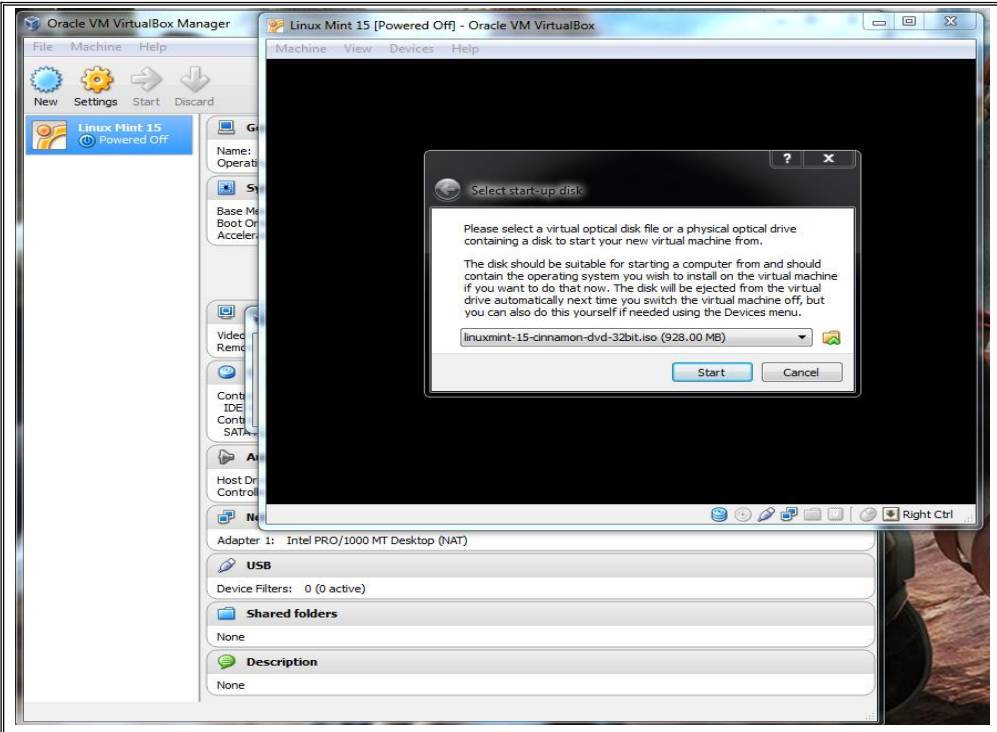

Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Details Snapshots
Create Virtual Machine
Memory size
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.
The recommended memory size is 512 MB.
2048 MB
4 MB
10240 MB
Next
Cancel

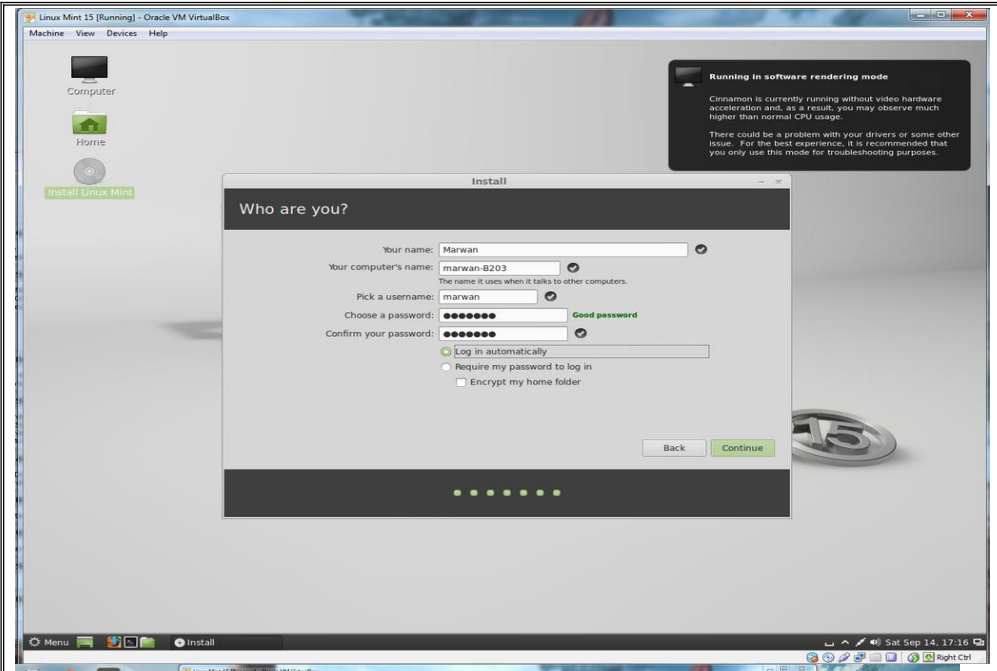

Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Details Snapshots
Create Virtual Machine
Hard drive
If you wish you can add a virtual hard drive to the new machine. You can either create a new hard drive file or select one from the list or from another location using the folder icon.
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.
The recommended size of the hard drive is 8.00 GB.
Do not add a virtual hard drive
Create a virtual hard drive now
Use an existing virtual hard drive file
Empty
Create
Cancel

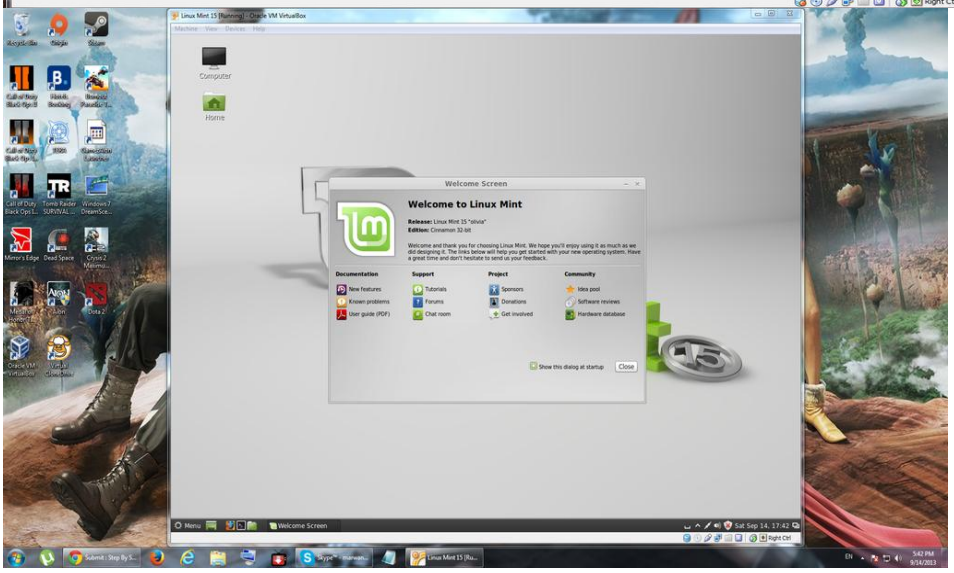

Oracle VM VirtualBox Manager
File Machine Help
New Settings Start
Snapshots
Create Virtual Hard Drive
Hard drive file type
Please choose the type of file that you would like to use for the new virtual hard drive. If you do not need to use it with other virtualization software you can leave this setting unchanged.
VDI (VirtualBox Disk Image)
VMDK (Virtual Machine Disk)
VHD (Virtual Hard Disk)
HDD (Parallels Hard Disk)
QED (QEMU enhanced disk)
QCOW (QEMU Copy-On-Write)
Hide Description
Next
Cancel


Oracle VM VirtualBox Manager
File Machine Help
New Settings Start
Snapshots
Create Virtual Hard Drive
Storage on physical hard drive
Please choose whether the new virtual hard drive file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).
A dynamically allocated hard drive file will only use space on your physical hard drive as it fills up (up to a maximum fixed size), although it will not shrink again automatically when space on it is freed.
A fixed size hard drive file may take longer to create on some systems but is often faster to use.
Dynamically allocated
Fixed size
Next
Cancel


Oracle VM VirtualBox Manager
File Machine Help
New Settings Start
Snapshots
Create Virtual Hard Drive
File location and size
Please type the name of the new virtual hard drive file into the box below or click on the folder icon to select a different folder to create the file in.
Linux Mint 15
Select the size of the virtual hard drive in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard drive.
20.00 GB
4.00 MB
2.00 TB
Create
Cancel


Oracle VM VirtualBox Manager
File Machine Help
New Settings Start Discard
Linux Mint 15
Powered Off
Linux Mint 15 [Powered Off] - Oracle VM VirtualBox
Machine View Devices Help
Select start-up disk
Please select a virtual optical disk file or a physical optical drive containing a disk to start your new virtual machine from.
The disk should be suitable for starting a computer from and should contain the operating system you wish to install on the virtual machine if you want to do that now. The disk will be ejected from the virtual drive automatically next time you switch the virtual machine off, but you can also do this yourself if needed using the Devices menu.
linuxmint-15-cinnamon-dvd-32bit.iso (928.00 MB)
Start
Cancel
Right Ctrl
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
Device Filters: 0 (0 active)
Shared folders
None
Description
None





Ex No : 6 **OPEN STACK INSTALLATION**

Install OpenStack and use it as Infrastructure as a Service and use technology own Cloud.

AIM :

To install the open stack and use it as a infrastructure as a service and Use technology own cloud

PROCEDURE:

Steps to install openstack on Ubuntu 18.04 in Virtual box

==========

1)First install virtual box https://www.virtualbox.org/wiki/Downl...

2)Download Ubuntu 18.04 ISO https://releases.ubuntu.com/18.04/

3)Create Ubuntu 18.04 VM in Virtual box 4 GB RAM + 2 vCPUs Hard disk capacity of Min 10 GB

4)After succesful installation of ubuntu ,start Openstack installation

Execute below steps in terminal:

sudo su apt update -y && apt upgrade -y sudo reboot sudo useradd -s /bin/bash -d /opt/stack -m stack echo "stack ALL=(ALL) NOPASSWD: ALL" |
sudo tee /etc/sudoers.d/stack sudo su - stack If git not installed:

sudo apt install git -y

git clone https://github.com/openstack-dev/devs... -b stable/pike devstack/ cd devstack/ ifconfig

If it did not work:

sudo apt install net-tools

ADMIN_PASSWORD=secret
DATABASE_PASSWORD=\$ADMIN_PASSWORD RABBIT_PASSWORD=\$ADMIN_PASSWORD
SERVICE_PASSWORD=\$ADMIN_PASSWORD
HOST_IP=10.0.2.15 RECLONE=yes EOF ./stack.sh

If stack.sh failed with permission error:

FORCE=yes ./stack.sh

After succesful installation of Openstack access the Horizon dashboard with below URL:

http://10.0.2.15/dashboard

RESULT:

Thus the open stack is installed successfully in the virtual box .

PART B

Ex.no 8: LED BLINK AND LED PATTERN

To implement LED Blink and LED Pattern With Arduino

AIM :

To implement LED blink and LED pattern with arduino.

LIST OF COMPONENTS NEEDED:

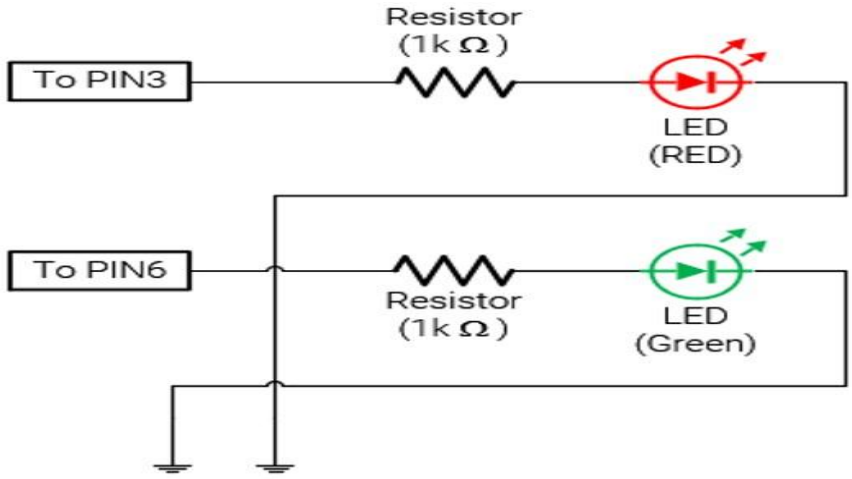
1.ARDUINO UNO
2.BREADBOARD.

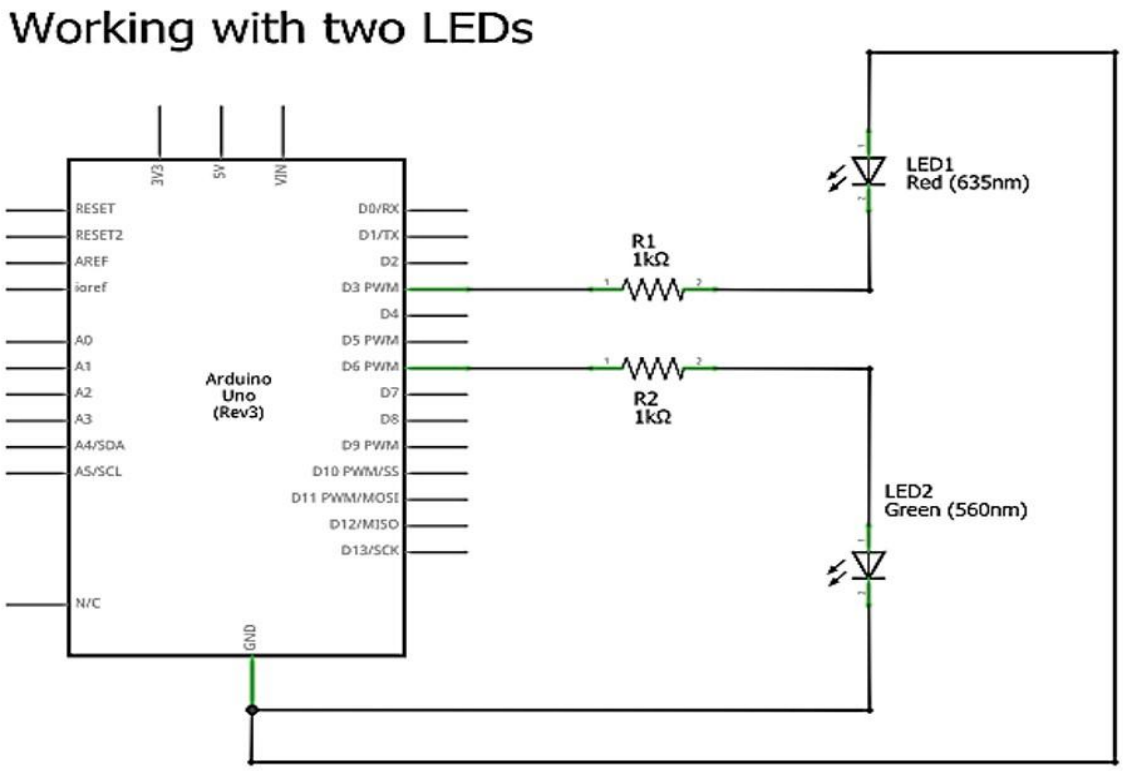
3.5MM LED: RED- 2 NUMBERS
4.RESISTOR 1K OHM
5.JUMPER WIRES-4

APPLICATION NEEDED:

ARDUINO IDE

CIRCUIT DIAGRAM :





ARDUINO CODE:

```
const int LED_red = 3;
const int LED_green = 6;

void setup() {
   pinMode (LED_red, OUTPUT);
   pinMode (LED_red, OUTPUT);
}

void loop() {
   digitalWrite (LED_red, HIGH);
   digitalWrite (LED_green, LOW);
   delay (1000);
   digitalWrite (LED_red, LOW);
   digitalWrite (LED_green, HIGH);
   delay (1000);
}
```

PROCEDURE:

1. Connect LED in the bread board. (For Ex: In led big needle is positive and small needle is negative)
2. Connect jumper wire from RED LED to 3rd pin in arduino uno kit and GREEN LED to 6th pin in arduino uno kit.
3. Connect one end of resistor to red LED negative and other end to ground of arduino .
4. Connect one end of resistor to GREEN LED negative and other end to ground of arduino .
4. Connect UNO board to computer.
5. In computer open arduino software and type the code in the arduino software.
6. Click on tools->arduino board and tools->port->com port.

RESULT:

Thus the LED blink and LED pattern is implemented successfully.

EX NO: 9 **LED BLINK AND LED PATTERN**

To implement LED Pattern with Push Button Control With Arduino

AIM:

To implement LED pattern with push button control with arduino uno kit.

HARDWARE REQUIREMENTS:

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	Push Button	1
4	Resistors(10k ,470 ohm)	1,7
5	5mm LED	7
6	Breadboard	1
7	Connecting wires	1

SOFTWARE REQUIREMENTS:

Arduino uno software

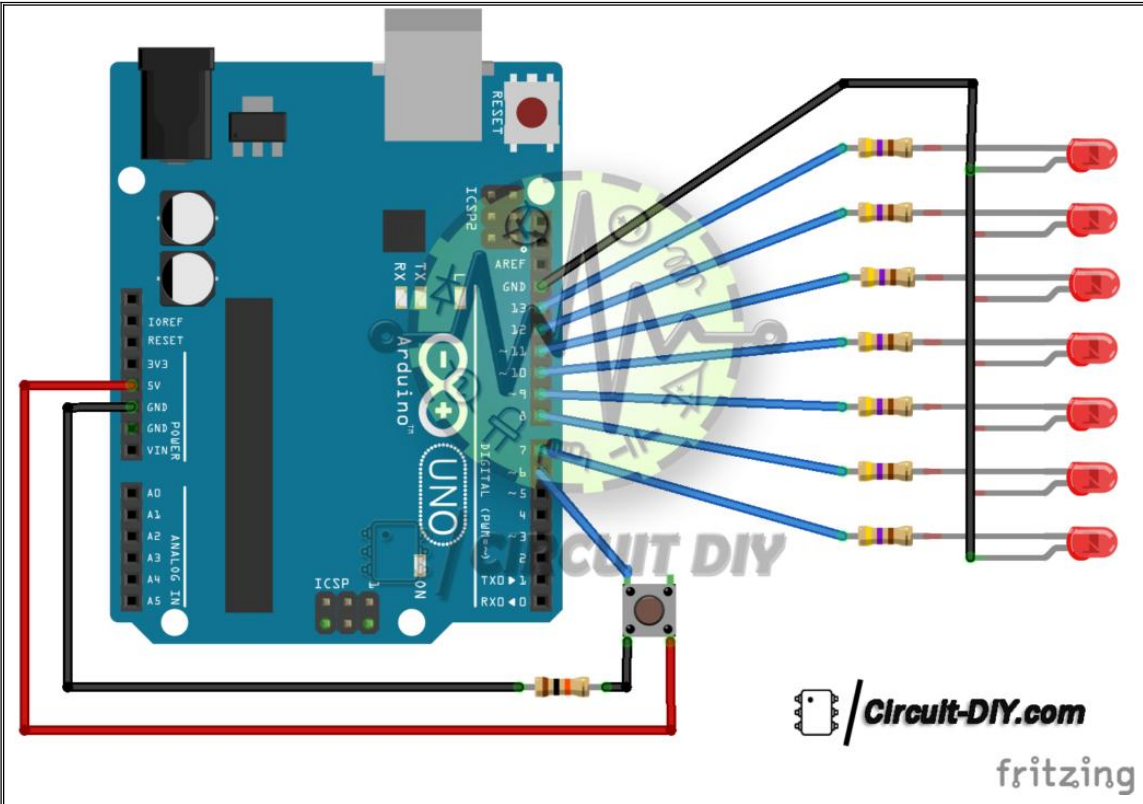
Connection

- **STEP # 1** (Make Push Button Connections)
- Pin1 to 5V of Arduino.
- Resistor 10k B/w Pin2 of Push Button & Ground of Arduino
- Pin2 is also connected to D6 of Arduino
- **STEP # 2** (Make LED Connections)
- Connect All -VE of LED To Ground to Arduino
- **STEP # 3** (Make Resistors Connections)

All Resistor's to +VE of LED and then D7,D8,D9,D10,D11,D12,D13 of Arduino

- **STEP # 4** (Upload Code)

CIRCUIT DIAGRAM :



Code:

```
int L1 = 13;
int L2 = 12;
int L3 = 11;
int L4 = 10;
int L5 = 9;
int L6 = 8;
int L7 = 7;  //7 LED pin

int buttonPin = 6;  //the number of the pushbutton pin

int de=50;  // delay time

int p=0;    // variable for pattem

int buttonState = 0;         // variable for reading the pushbutton status

void setup() {


  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(L3, OUTPUT);
  pinMode(L4, OUTPUT);
  pinMode(L5, OUTPUT);
  pinMode(L6, OUTPUT);
  pinMode(L7, OUTPUT);
```

```
  pinMode(buttonPin, INPUT);

}

void loop()
{
 buttonState = digitalRead(buttonPin);

 if (buttonState == HIGH)

    {
      p++;
      delay(2000);
    }

  if(p==1)
  {
 digitalWrite(L1,1);
 digitalWrite(L2,0);
 digitalWrite(L3,0);
 digitalWrite(L4,0);
 digitalWrite(L5,0);
 digitalWrite(L6,0);
 digitalWrite(L7,0);  //1
  delay(de);

 digitalWrite(L1,0);
 digitalWrite(L2,1);
 digitalWrite(L3,0);
 digitalWrite(L4,0);
 digitalWrite(L5,0);
 digitalWrite(L6,0);
 digitalWrite(L7,0);  //2
  delay(de);

 digitalWrite(L1,0);
 digitalWrite(L2,0);
 digitalWrite(L3,1);
 digitalWrite(L4,0);
 digitalWrite(L5,0);
 digitalWrite(L6,0);
 digitalWrite(L7,0);  //3
  delay(de);

 digitalWrite(L1,0);
 digitalWrite(L2,0);
 digitalWrite(L3,0);
 digitalWrite(L4,1);
 digitalWrite(L5,0);
 digitalWrite(L6,0);
```

```
digitalWrite(L7,0);  //4
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //5
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);  //6
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);  //7
 delay(de);
 }



 if(p==2)
 {
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);  //7
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
```

```
digitalWrite(L7,0);  //6
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //5
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //4
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //3
 delay(de);


 digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //2
 delay(de);

  digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //1
 delay(de);

 }
```

```
   if(p==3)
 {
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //1
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //2
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //3
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //4
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //5
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
```

```
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);  //6
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);  //7
 delay(de);


 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);  //6
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //5
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //4
 delay(de);

 digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //3
 delay(de);
```

```
 digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //2
 delay(de);
 }



   if(p==4)
 {
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);  //1,7
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);  //2,6
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //3,5
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //4
 delay(de);
```

```
 }


    if(p==5)
 {


digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0);  //4
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0);  //3,5
 delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0);  //2,6
 delay(de);

digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1);  //1,7
 delay(de);
 }

 if(p==6)
{
 digitalWrite(L1,1);
 delay(de);
 digitalWrite(L2,1);
 delay(de);
```

```
 digitalWrite(L3,1);
 delay(de);
 digitalWrite(L4,1);
 delay(de);
 digitalWrite(L5,1);
 delay(de);
 digitalWrite(L6,1);
 delay(de);
 digitalWrite(L7,1);  //1,7
 delay(de);
 digitalWrite(L7,0);  //1,7
 delay(de);
 digitalWrite(L6,0);
 delay(de);
 digitalWrite(L5,0);
 delay(de);
 digitalWrite(L4,0);
 delay(de);
 digitalWrite(L3,0);
 delay(de);
 digitalWrite(L2,0);
 delay(de);
 digitalWrite(L1,0);
 delay(de);


}

 if(p==7)
{
 digitalWrite(L1,0);
 digitalWrite(L2,0);
 digitalWrite(L3,0);
 digitalWrite(L4,0);
 digitalWrite(L5,0);
 digitalWrite(L6,0);
 digitalWrite(L7,0);  //1,7
  p=0;
}

}
```

EX NO : 10 LCD DISPLAY WITH ARDUINO

To display “Hello World “ in LCD 16X2 Display With Arduino

AIM :

To display “Hello World “ in LCD 16X2 Display With Arduino

HARDWARE REQUIREMENTS:

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	LCD disply	1
4	I2c module	1
5	Jumper cables(M to F)	4

PROCEDURE:

1.Connect LCD display to I2C module and solder it using the pins.
2.connect I2c module VCC to 5v in arduino
3.Connect I2C module ground pin to Arduino Ground
4.Connect SDA and SCL to A4 and A5 in arduino
5.Fix arduino in the computer.
6.Before running the code you must download two library files namely liquid crystal and wire for running the LCD display using Arduino.
7.Then goto sketch -> include library->add zip library,import two library files which is downloaded.
8.Goto Examples->liquid crystal->select hello worls program.
9.Change the coding as below:
10.After typing the code right click and click auto format. And run the code.
CODE:

```
#include<Wire.h>
#incllude<LiquidCrystal_I2C.h>
Liquid Crystal_I2C lcd(0*27,16,2);
Void setup()
{
lcd.init();
lcd.backlight();
lcd.setcursor(1,0);
lcd.print(“HELLO WORLD”);
lcd.setcursor(1,1)
lcd.print(“CSE DEPT”);
}
```

EX NO : 11 Servo Motor Control with Arduino

To implement the Servo Motor Control with Arduino

AIM :

To implement the servo motor with arduino uno kit.

HARDWARE REQUIREMENTS:

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	Servo motor	1
6	Breadboard	1
7	Connecting wires	1

PROCEDURE:

Servo motor wires:

1.Red color wire is always positive

2.brown color is negative which is ground.

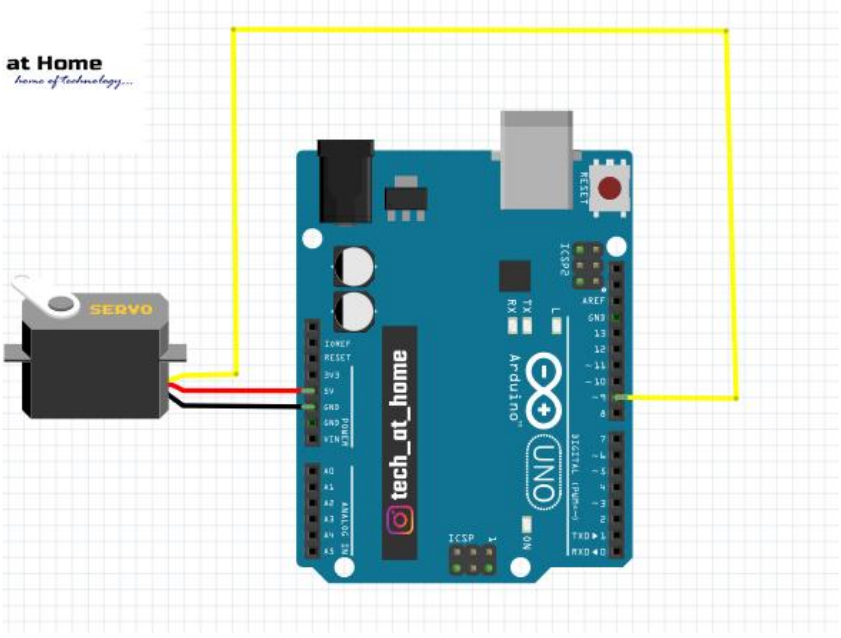
3.orange color is the signal pin which we need to connect to the digital pin of the arduino

4.connect orange color pin to 9th pin in arduino board.

5..import the servo library file into the arduino uno software;

6.Run the code, the servo motor will be shifting from 0degree to 180 degree as given in the code.

CIRCUIT DIAGRAM :



CODE:

```
#include<Servo.h>
Servo s1;
Void setup()
{
S1.attach(9);
```

```
}
Void loop()
{
S1.write(0);
Delay(1000);
S1.write(180);
delay(1000);
s1.write(0);
}
```

RESULT :

Thus the servo motor control is implemented successfully using arduino uno.

EX NO:12: MONITORING THE TEMPERATURE SENSOR

To implement and monitor the LM35 Temperature Sensor and Ultrasonic Distance Measurement With Arduino

AIM:

To implement and monitor the LM35 Temperature Sensor and Ultrasonic Distance Measurement With Arduino

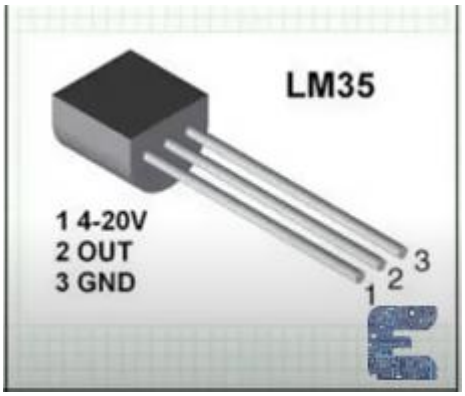
HARDWARE REQUIREMENTS:

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	LCD display	1
4	I2c module	1
5	Jumper cables(M to F)	4
6	Ultrasonic sensor	1
7	LM35 temperature sensor	1

PROCEDURE:

1.Connect LCD display to arduino kit by following steps:
(i).connect I2c module VCC to 5v in arduino
(ii).Connect I2C module ground pin to Arduino Ground
(iii)Connect SDA and SCL to A4 and A5 in arduino

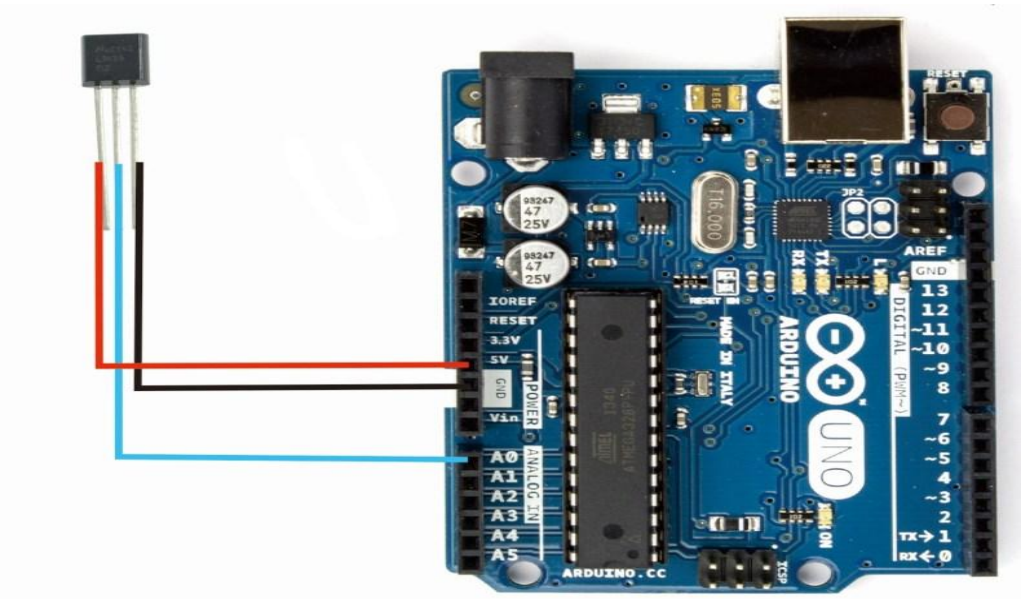
2.Connect Ultrasonic sensor to arduino by following steps:
(i)connect TRIG to pin 2 of arduino
(ii) Connect ECHO to pin 3 of arduino.
3.Connect LM35 temperature sensor with arduino kit by following steps:



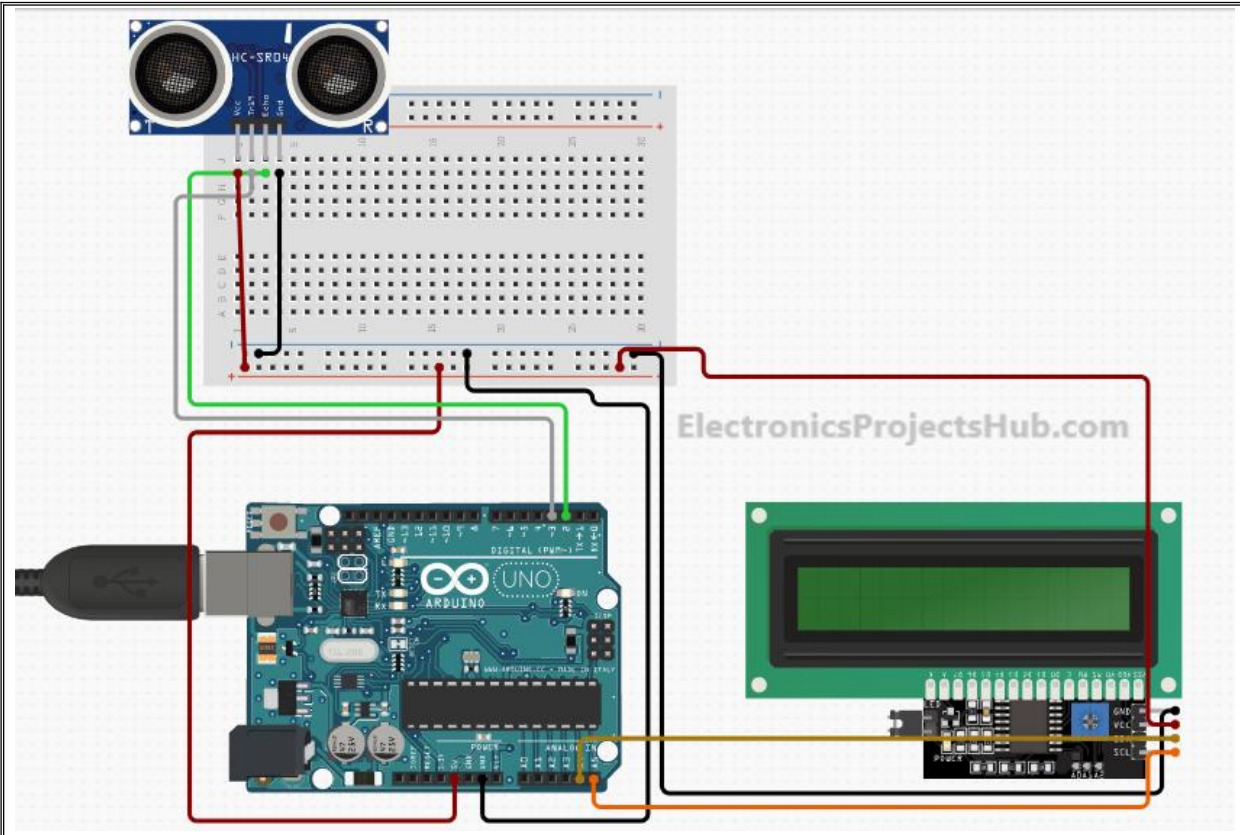
(i) Connect positive(+) pin(1) to 5v of arduino
(ii) connect pin 2 of LM35 to A0 of Arduino
(iii)connect pin 3 to ground of arduino

3.Upload the header files <wire.h> and <liquid crystaldisplay.h>

CIRCUIT DIAGRAM FOR LM35 TO ARDUINO



CIRCUIT DIAGRAM FOR ULTRASONIC SENSOR WITH ARDUINO



CODE:

```
#include<wire.h>
#incllude<LiquidCrystal_I2C.h>
Liquid Crystal_I2C lcd(0*3F,16,2);
Const int trigPin=2;
Const int echoPin=3;
long duration ;
int dist;

Void setup()
{

Serial.begin(9600);
pinMode(trigPin,OUTPUT);
pinMode(echoPin,INPUT);
Serial.println(“Ultrasonic working Principle”);
lcd.begin();
lcd.backlight();
lcd.clear();
lcd.setcursor(4,0);
lcd.print(WELCOME”);
lcd.setcursor(0,1);
lcd.print(“LET US SEE OUTPUT”);
delay(1000);
}
Void loop()
{
digitalWrite(trigPin,LOW);
```

```
delayMicroseconds(2);
digitalWrite(trigPin,HIGH);
delayMicroseconds(2);
digitalWrite(trigPin,LOW);
duration=pulseIn(echoPin,HIGH);
dist=(duration/2)*0.0340;
Serial.print(“Distance: “);
Serial.print(dist);
Serial.print(“  cm “);
lcd.clear();
lcd.setcursor(0,0);
lcd.print(“OUTPUT”);
lcd.setcursor(0,1);
lcd.print(“Distance: “);
lcd.print(dist);
lcd.print(“   cm”);
delay(500);

// DISPLAY TEMPERATURE//
lcd.clear();
  int reading = analogRead(sensorPin);

  float voltage = reading * (5.0 / 1024.0);

  float temperatureC = voltage * 100;

  Serial.print("Temperature: ");

   lcd.setCursor(0,0);
  lcd.print("Temperature:");
  lcd.setCursor(0,1);
  lcd.print(temperatureC);
  lcd.setCursor(6,1);
  lcd.print("C");


  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  lcd.setCursor(8,1);
  lcd.print(temperatureF);
  lcd.setCursor(14,1);
  lcd.print("F");

  delay(1000);

  }
```

RESULT:

Thus the LM35 Temperature Sensor and Ultrasonic Distance Measurement Is used successfully using arduino kit.

EX NO :13 IR SENSOR ANALOG INPUT USING ARDUINO

To implement the IR Sensor Analog Input With Arduino

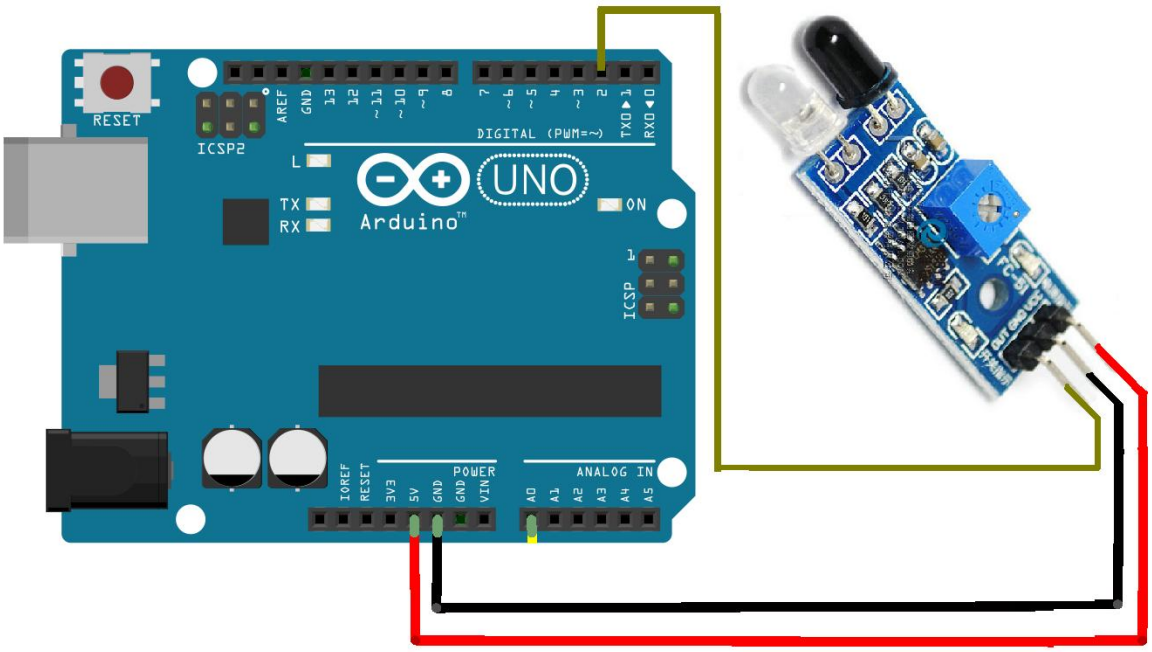
AIM:

To implement the IR Sensor Analog Input With Arduino

HARDWARE REQUIREMENTS:

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	Jumper cables(M to F)	4
4	IR sensor	1
5	LED	1

CIRCUIT DIAGRAM:



PROCEDURE:

1.Connect IR sensor with the following steps:

(i)connect out pin to 7 th pin of arduino
(ii)connect VCC to 5v of arduino
(iii) connect GRND to GRND of arduino

2.Connect LED to detect the sensor

(i)connect 9 th pin to LED positive and negative to grnd.
3.Upload the code and sensor will work with led light detection.
4.We can use buzzor for detection the sound.

CODE:

```
Void setup()
{
pinMode(7,INPUT);
pinMode(9,OUTPUT);
}
Void loop()
{
If (digitalRead(7)== HIGH)
```

```
{
        digitalWrite(9,HIGH);
}
else
{
        digitalWrite(9,LOW);
}
}
```

RESULT:

Thus the IR Sensor Analog Input is implemented With Arduino successfully.

EX NO:14 TEMPERATURE SENSOR MONITORING

Using ThinkSpeak Cloud Reading Temperature Sensor Monitoring with NodeMCU /Raspberry Pi

AIM :

To use node MCU for reading temperature sensor and monitor the temperature.

HARDWARE REQUIREMENTS :

S.NO	COMPONENTS	QUANTITY
1	Arduino Uno R3	1
2	USB Cable A/B	1
3	Jumper cables(M to F)	4
4	NODE MCU	1
5	WIFI ADAPTER	1

Procedure :

1. To create the think speak account
   - Sign up to THINK SPEAK account ,by clicking on create new account
   - Enter the details e-mail id,name and click on continue,
   - You will be receiving the confirmation mail link and click on the link and then continue to sign in the thinkspeak account.
2. Click on create channel and enter the field details.Then channel will be created.
3. Click on API keys and there will be 2 keys write API and Read API keys,write API keys will store the sensor values and write in think speak and when you give the think speak key value to arduino board then it is called write API keys.
4. Download DHT library and update into the arduino IDE

5. Upload below code in the arduino uno
6. Run the code and automatically the temperature will be increased upto the current temperature in the think speak cloud interface.

/Code to connect Arduino with ThingSpeak using Wifi Module (Store Humidity Value)

```
#include "dht.h"
#define dht_apin A0
dht DHT;

#include <SoftwareSerial.h>      //Software Serial library
SoftwareSerial espSerial(2, 3);   //Pin 2 and 3 act as RX and TX. Connect them to TX and RX of ESP8266
#define DEBUG true
String mySSID = "GS";       // WiFi SSID
String myPWD = "ptleecnpt"; // WiFi Password
String myAPI = "IFNXUVCVENMT25D8";   // API Key
String myHOST = "api.thingspeak.com";
String myPORT = "80";
String myFIELD = "field1";
int sendVal;



void setup()
{
 Serial.begin(9600);
 Serial.println("DHT11 Humidity & temperature Sensor\n\n");
 delay(1000);
 espSerial.begin(115200);
 espData("AT+RST", 1000, DEBUG);                      //Reset the ESP8266 module
 espData("AT+CWMODE=1", 1000, DEBUG);                 //Set the ESP mode as station mode
 espData("AT+CWJAP=\"" + mySSID + "\",\"" + myPWD + "\"", 1000, DEBUG);
//Connect to WiFi network
 delay(1000);
}

void loop()
```

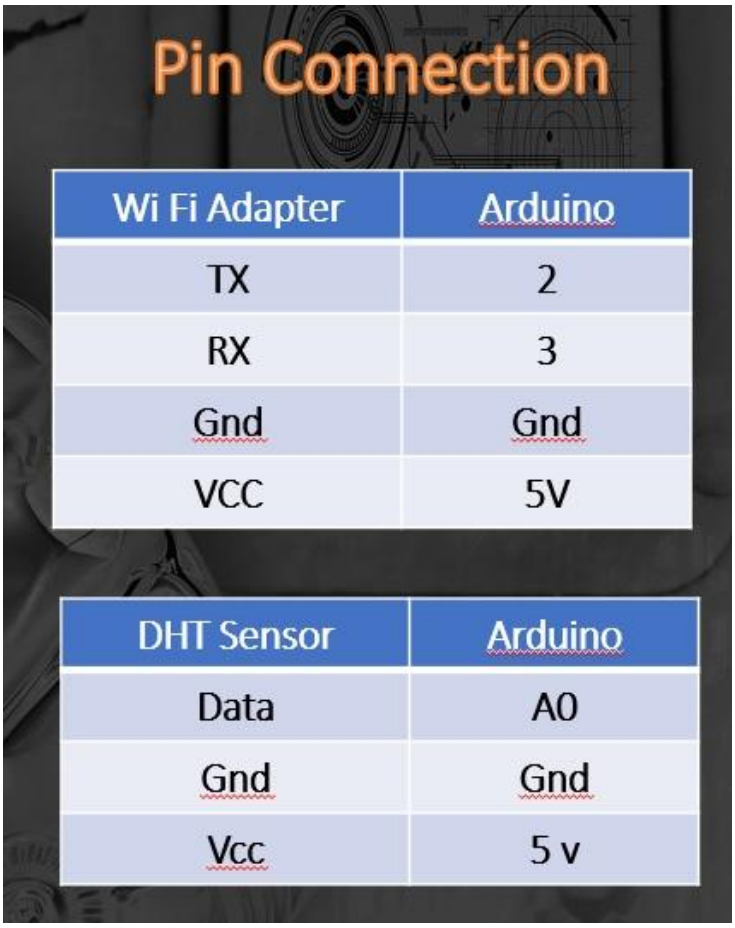
```
{

 DHT.read11(dht_apin);
 String sendData = "GET /update?api_key=" + myAPI + "&" + myFIELD + "=" +
String(DHT.humidity);
 espData("AT+CIPMUX=1", 1000, DEBUG);       //Allow multiple connections
 espData("AT+CIPSTART=0,\"TCP\",\"" + myHOST + "\"," + myPORT, 1000,
DEBUG);
 espData("AT+CIPSEND=0," + String(sendData.length() + 4), 1000, DEBUG);
 espSerial.find(">");
 espSerial.println(sendData);
 Serial.print("Value to be sent: ");
 Serial.println(DHT.humidity);
 espData("AT+CIPCLOSE=0", 1000, DEBUG);
 delay(10000);
}

String espData(String command, const int timeout, boolean debug)
{
 Serial.print("AT Command ==> ");
 Serial.print(command);
 Serial.println("     ");

 String response = "";
 espSerial.println(command);
 long int time = millis();
 while ( (time + timeout) > millis())
 {
   while (espSerial.available())
   {
     char c = espSerial.read();
     response += c;
   }
 }
 if (debug)
 {
 }
```

```
  return response;
}
```



RESULT:

Thus the thinkspeak cloud is connected with arduino uno board and temperature is calculated successfully.