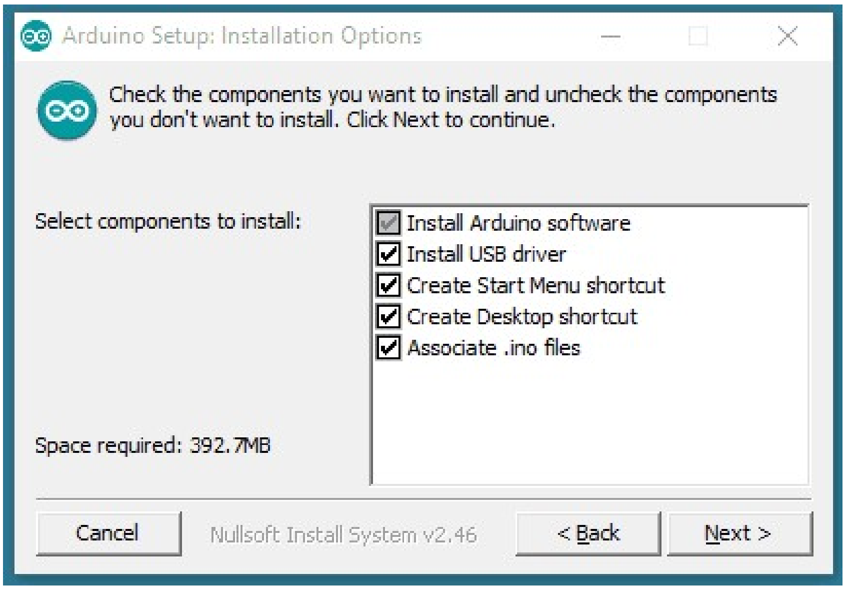
Download the Arduino Software (IDE)

Get the latest version from

https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE.

You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.


Arduino Setup: Installation Options
Cancel
Nullsoft Install System v2.46
< Back
Next >

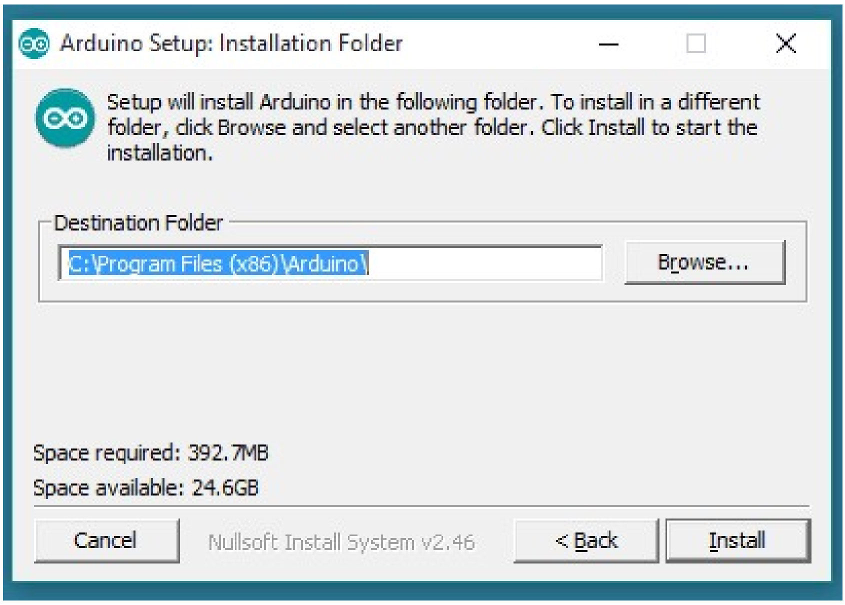
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Install Arduino software
☑ Install USB driver
☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Associate .ino files

Space required: 392.7MB

Choose the components to install.



Arduino Setup: Installation Folder

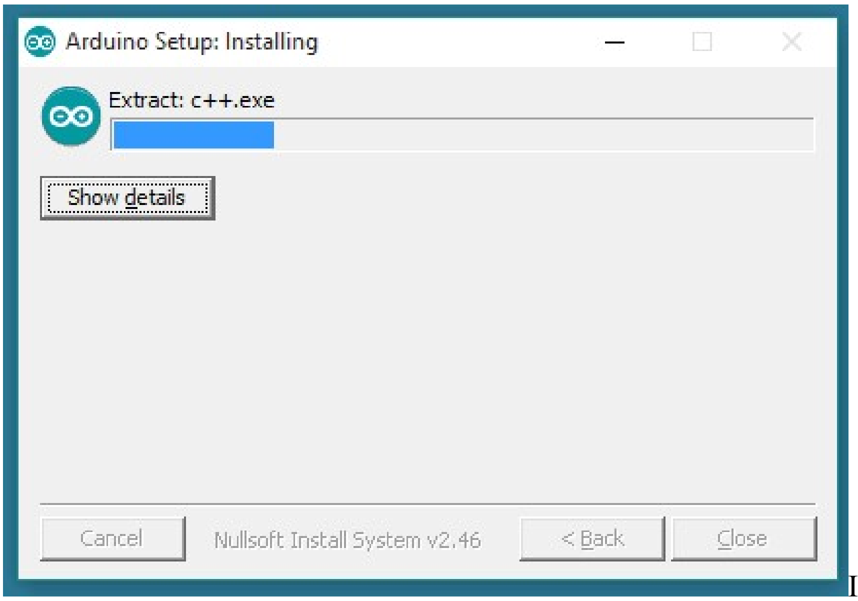
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino\

Space required: 392.7MB

Space available: 24.6GB

Choose the installation directory.



Arduino Setup: Installing

Extract: c++.exe

Installation in progress.

The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Steps to upload the code to the project

The steps are listed below:

- Open the Arduino IDE.
- Select the type of board from Tools -> Board -> Arduino UNO.
- Select the port from Tools -> Port -> COM..
- Upload the sketch to the connection diagram.

1. Make your Arduino board blink an LED.

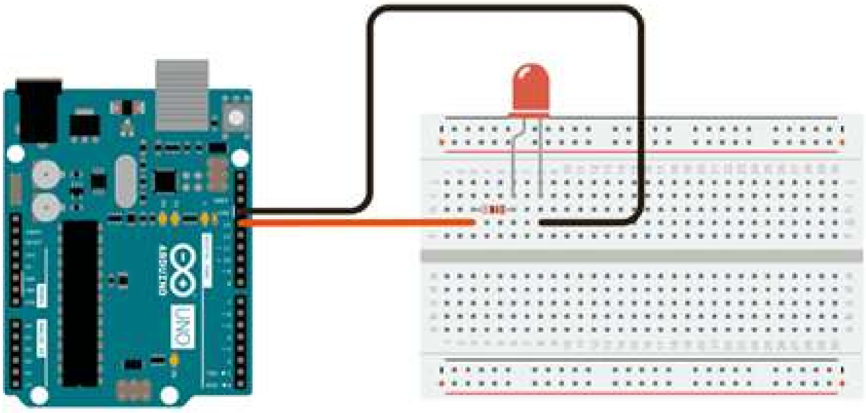
Aim:
To make a LED Blink operation using Arduino

Procedure:

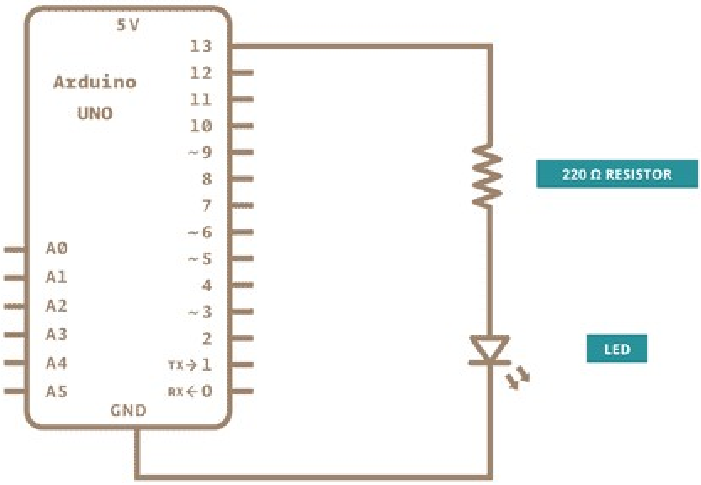
i. Start the Arduino IDE and enter the code
ii. Declare the variable for input pin
iii. Initialize digital pin 13 as an output in void setup function
iv. The loop function runs over and over again forever
v. Turn the LED on by making the voltage HIGH
vi. Turn the LED on by making the voltage LOW

Circuit Diagram:

The value of the resistor in series with the LED may be of a different value than 220 ohm; the LED will light up also with values up to 1K ohm.



Schematic Diagram:



Program

```
int led = 13;
void setup() {
        pinMode(led, OUTPUT);
}
void loop() {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
}
```

2. Create an application to find Motion detection using Arduino board.

Aim:

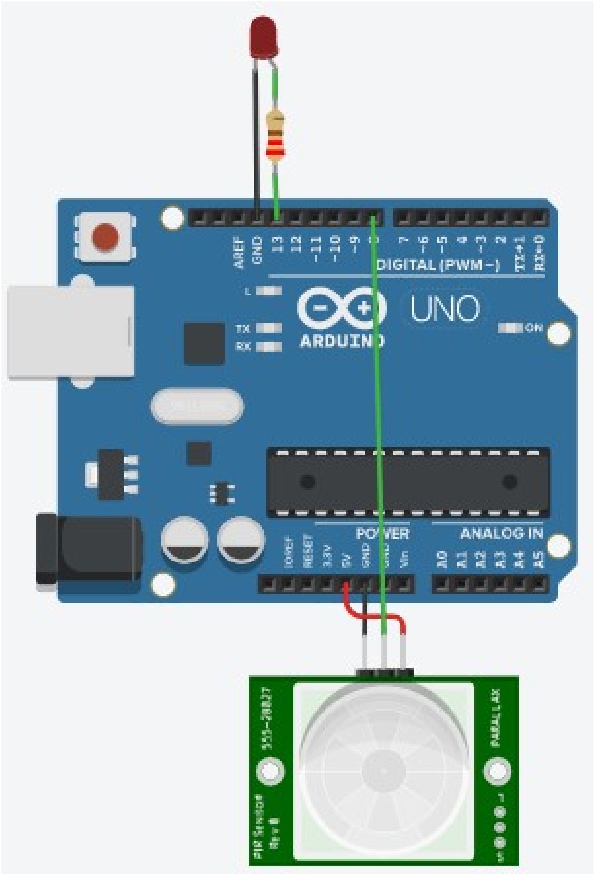
To find Motion detection using Arduino board.

Procedure:

i. Start the Arduino IDE and enter the code
ii. Declare the PIR sensor output variable
iii. Define the baud rate for serial transmission
iv. The loop function runs over and over again forever
v. Based on the sensor output show the movement.
vi. If the PIR sensor value is 1, then print the statement of movement detection
vii. If the PIR sensor value is 0, then print the statement of no movement detection.

Circuit Diagram:

We will show the connection using the Simulator so that the connections become clearer and more precise. We can make the same connection using the hardware devices.



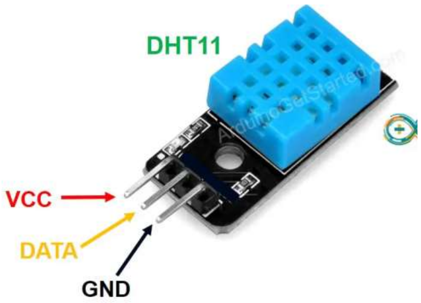
Program:

```
int LEDpin = 13;   // LED pin
int PIRpin = 8;    // The pin of Arduino connected to the PIR output
int PIRvalue = 0;  // It specifies the status of PIR sensor

void setup() {
 pinMode(LEDpin, OUTPUT);
 pinMode(PIRpin, INPUT);
  // the output from the sensor is considered as input for Arduino
 Serial.begin(9600);
}

void loop()
```

```
{
 PIRvalue = digitalRead(PIRpin);
 if (PIRvalue == HIGH)
 {
   digitalWrite(LEDpin, HIGH);
   // turn ON LED if the motion is detected
   Serial.println("hello, I found you...heyyy..");
 }
 else
 {
   digitalWrite(LEDpin, LOW);
   // LED will turn OFF if we have no motion
   Serial.println("I cannot find you");
  delay(1000);
}
}
```

3. Create an application to measure Temperature and Humidity using arduino board.

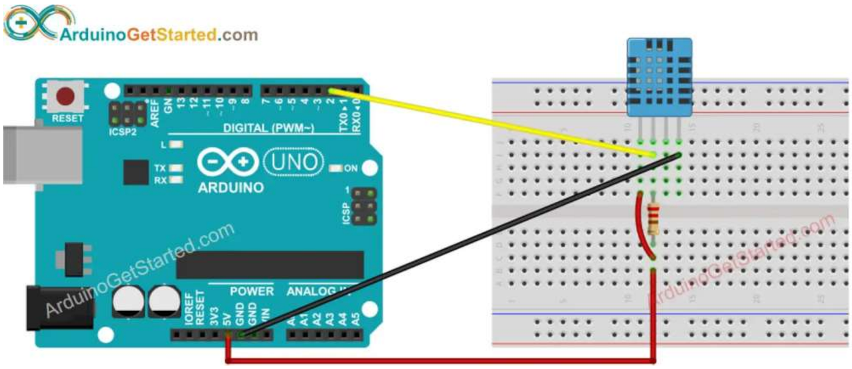
Aim:
To measure the temperature and humidity using Arduino board with DHT 11 Sensor.



Procedure:

i. Start the Arduino IDE and enter the code
ii. Declare the pin for temperature and humidity
iii. Define the baud rate for serial transmission
iv. The loop function runs over and over again forever
v. Based on the sensor output show the temperature and humidity
vi. Display the sensor value in serial monitor.

Circuit Diagram:

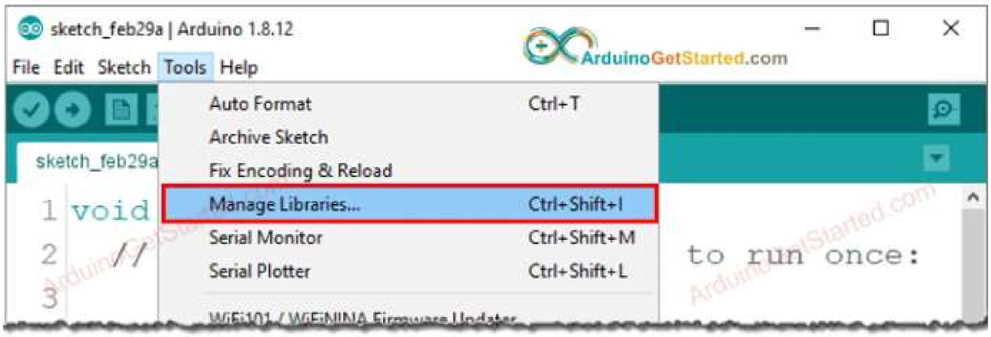


Program:

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin(); // initialize the sensor
}

void loop() {
  // wait a few seconds between measurements.
  delay(2000);

  // read humidity
  float humi  = dht.readHumidity();
  // read temperature as Celsius
  float tempC = dht.readTemperature();
  // read temperature as Fahrenheit
  float tempF = dht.readTemperature(true);

  // check if any reads failed
  if (isnan(humi) || isnan(tempC) || isnan(tempF)) {
    Serial.println("Failed to read from DHT sensor!");
  } else {
    Serial.print("Humidity: ");
    Serial.print(humi);
    Serial.print("%");

    Serial.print("  |  ");

    Serial.print("Temperature: ");
    Serial.print(tempC);
    Serial.print("°C ~ ");
    Serial.print(tempF);
    Serial.println("°F");
  }
}
```

Quick Steps

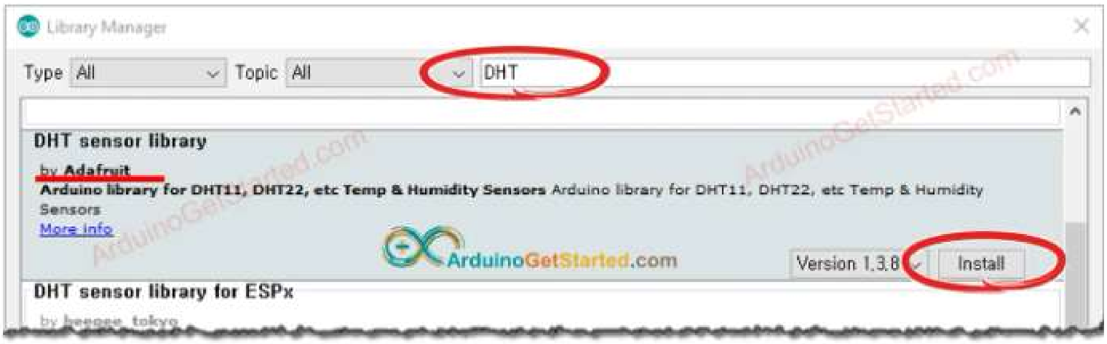
- Connect Arduino to PC via USB cable
- Open Arduino IDE, select the right board and port
- On Arduino IDE, Go to **Tools** » **Manage Libraries**



sketch_feb29a | Arduino 1.8.12
ArduinoGetStarted.com
File Edit Sketch Tools Help
sketch_feb29a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater

```
void
  //
                to run once:
```

- Search ***"DHT"***, then find the DHT sensor library by Adafruit
- Click **Install** button to install the library.



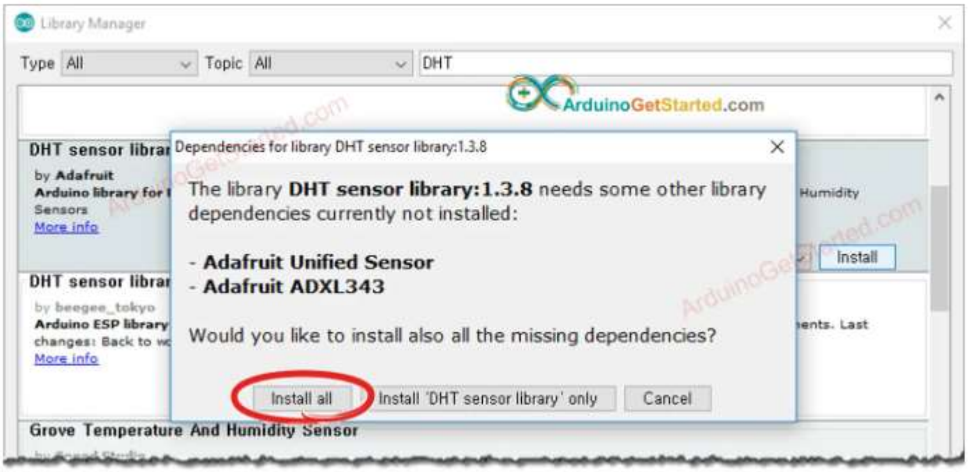
Type All Topic All DHT

DHT sensor library
by **Adafruit**
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors

Version 1.3.8

DHT sensor library for ESPx
by beegee_tokyo

- You will be asked for intalling some other library dependencies
- Click **Install All** button to install all library dependencies.

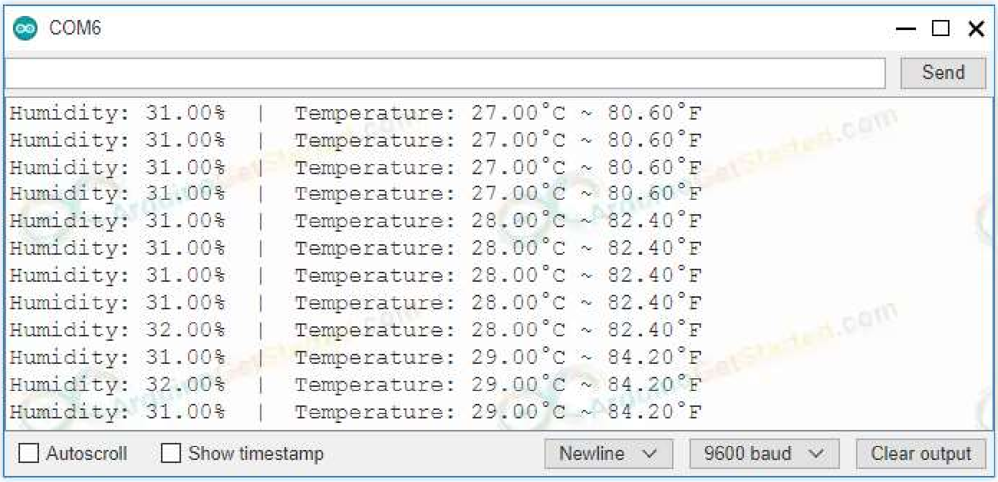


The library **DHT sensor library:1.3.8** needs some other library dependencies currently not installed:

- **Adafruit Unified Sensor**
- **Adafruit ADXL343**

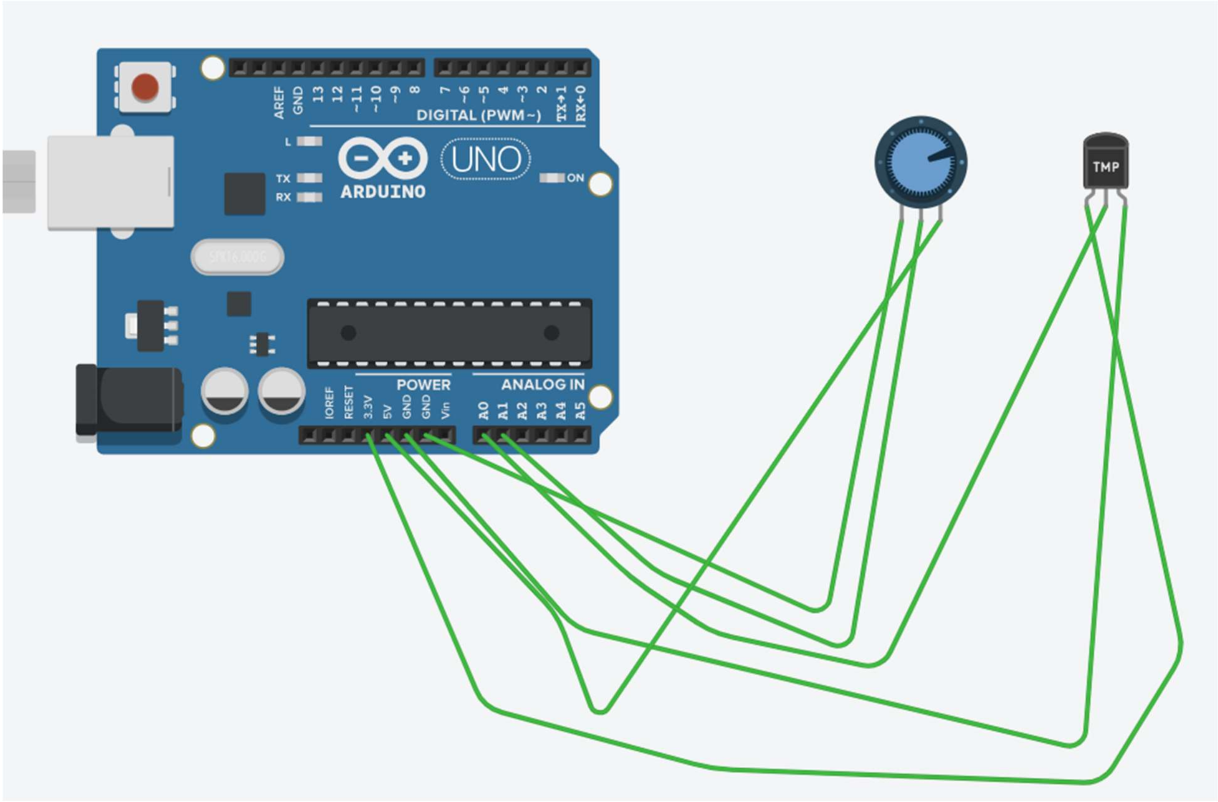
Would you like to install also all the missing dependencies?

- Copy the above code corresponding to the sensor you have and open with Arduino IDE
- Click **Upload** button on Arduino IDE to upload code to Arduino
- Make enviroment around sensor hotter or colder
- See the result on Serial Monitor.



```
Humidity: 31.00%  |  Temperature: 27.00°C ~ 80.60°F
Humidity: 31.00%  |  Temperature: 27.00°C ~ 80.60°F
Humidity: 31.00%  |  Temperature: 27.00°C ~ 80.60°F
Humidity: 31.00%  |  Temperature: 27.00°C ~ 80.60°F
Humidity: 31.00%  |  Temperature: 28.00°C ~ 82.40°F
Humidity: 31.00%  |  Temperature: 28.00°C ~ 82.40°F
Humidity: 31.00%  |  Temperature: 28.00°C ~ 82.40°F
Humidity: 31.00%  |  Temperature: 28.00°C ~ 82.40°F
Humidity: 32.00%  |  Temperature: 28.00°C ~ 82.40°F
Humidity: 31.00%  |  Temperature: 29.00°C ~ 84.20°F
Humidity: 32.00%  |  Temperature: 29.00°C ~ 84.20°F
Humidity: 31.00%  |  Temperature: 29.00°C ~ 84.20°F
```

In TinkerCad



```
const int analogIn = A0;
int humiditysensorOutput = 0;
// Defining Variables
int RawValue= 0;
double Voltage = 0;
double tempC = 0;
double tempF = 0;

void setup(){
  Serial.begin(9600);
  pinMode(A1, INPUT);
}

void loop(){

  RawValue = analogRead(analogIn);
  Voltage = (RawValue / 1023.0) * 5000; // 5000 to get millivots.
  tempC = (Voltage-500) * 0.1; // 500 is the offset
  tempF = (tempC * 1.8) + 32; // convert to F
  Serial.print("Raw Value = " );
  Serial.print(RawValue);
  Serial.print("\t milli volts = ");
  Serial.print(Voltage,0); //
  Serial.print("\t Temperature in C = ");
  Serial.print(tempC,1);
  Serial.print("\t Temperature in F = ");
  Serial.println(tempF,1);
  humiditysensorOutput = analogRead(A1);
  Serial.print("Humidity: "); // Printing out Humidity Percentage
  Serial.print(map(humiditysensorOutput, 0, 1023, 10, 70));
  Serial.println("%");
  delay(5000);  //iterate every 5 seconds

}
```

4. Demonstrate Ultrasonic distance sensor in Arduino board.

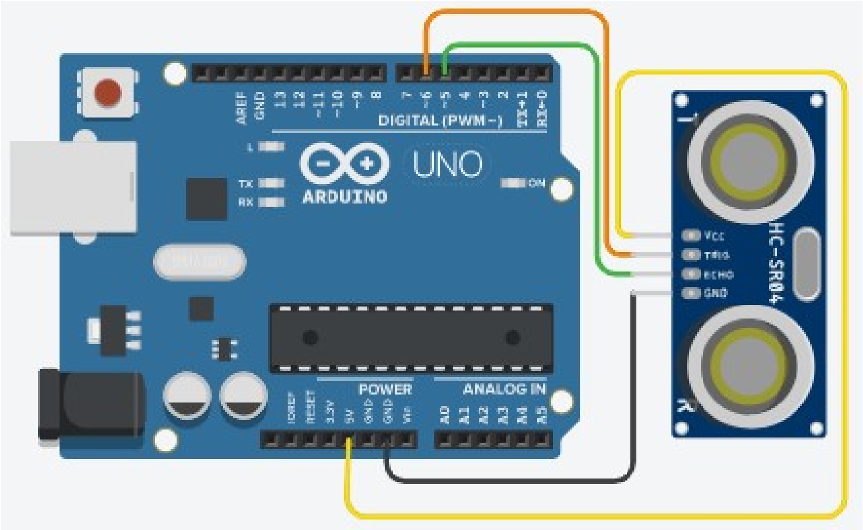
Aim:

To find object distance detection using Arduino board.

Procedure:

i. Start the Arduino IDE and enter the code
ii. Declare the pin for echo and trigger
iii. Define the baud rate for serial transmission
iv. The loop function runs over and over again forever
v. Based on the sensor output show the distance between the object and sensor
vi. Display the distance value in serial monitor.

Circuit Diagram:



Program:

```
#define ECHOpin 5 // it defines the ECHO pin of the sensor to pin 5 of Arduino
#define TRIGpin 6   // we have defined the variable
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

void setup()
{
  pinMode(TRIGpin, OUTPUT); // It sets the ECHO pin as OUTPUT
  pinMode(ECHOpin, INPUT); // It sets the TRIG pin as INPUT
  Serial.begin(9600); // // Serial Communication at the rate of 9600 bps
  Serial.println("Test of the Ultrasonic Sensor HCSR04"); // It will appear on Serial Monitor
  Serial.println("with the Arduino UNO R3 board");
}

void loop()
{
  // It first sets the TRIG pin at LOW for 2 microseconds
  digitalWrite(TRIGpin, LOW);
  delayMicroseconds(4);
  // It now sets TRIG pin at HIGH for 15 microseconds
  digitalWrite(TRIGpin, HIGH);
  delayMicroseconds(15);
  digitalWrite(TRIGpin, LOW);
  // It will read the ECHO pin and will return the time
  duration = pulseIn(ECHOpin, HIGH);
  // distance formula
  distance = duration*(0.034/2); // (speed in microseconds)
  // Speed of sound wave (340 m/s)divided by 2 (forward and backward bounce)
  // To display the distance on Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
 Serial.println(" cm"); //specified unit of distance
```

5. Create an application to learn to test the output & input of GPIO pins on Raspberry pi.

Aim:

To create application to test the GPIO Pins using LED on Raspberry pi.

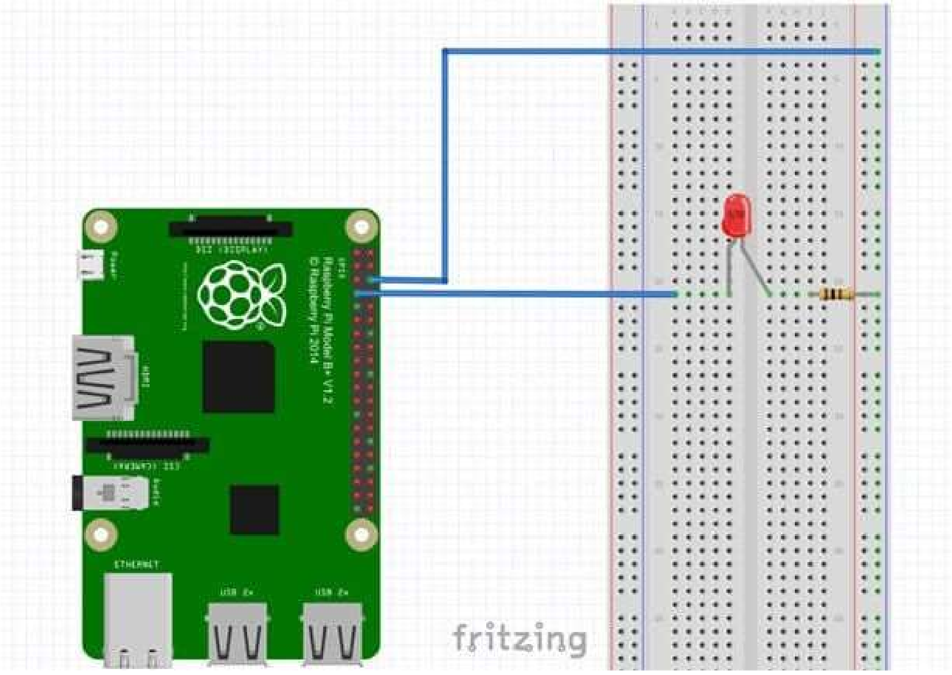
Procedure:

i. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows
https://pimylifeup.com/raspberry-pi-gpio/

ii. Open the python IDE and enter the code

iii. Declare the GPIO pin for LED
iv. Based on the sensor output show the LED Blink.
v. Display the output on the Pi Board.

Circuit Diagram:



Program:

```
#import the GPIO and time package

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(7, GPIO.OUT)

# loop through 50 times, on/off for 1 second

for i in range(50):

    GPIO.output(7,True)
```

```
        time.sleep(1)
        GPIO.output(7,False)
        time.sleep(1)
    GPIO.cleanup()
```

Run the Code

```
    sudo python led_blink.py
```

6. Develop an application to set up a Apache Tomcat web server on a Raspberry Pi.

How to install Apache tomcat on raspberry pi

Please follow the following steps to install apache tomcat on raspberry pi.

1)First open the raspberry pi terminal.

2)Execute the following command to update existing packages.

```
sudo apt-get update
```

3)In order to install apache tomcat on raspberry, you need a JDK. So please execute the following command to install JDK on raspberry pi.

```
sudo apt-get install default-jdk
```

4)Now execute the following command to make tomcat folder.

```
mkdir tomcat
```

5)Execute the following command to go in tomcat folder.

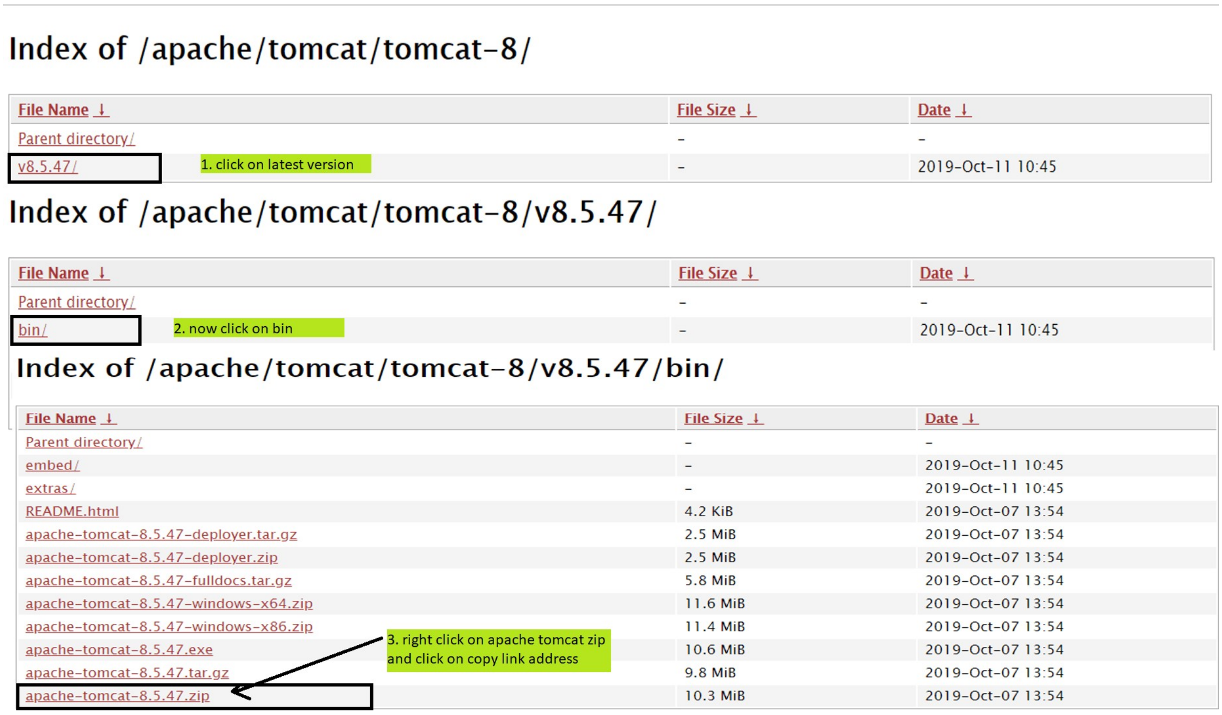
```
cd tomcat
```

6)Execute the following command to download apache tomcat.

```
sudo wget http://mirrors.estointernet.in/apache/tomcat/tomcat-8/v8.5.47/bin/apache-tomcat-8.5.47.zip
```

IMPORTANT NOTE: To know available apache tomcat version please open the following URL in any browser and follow following steps show image.

http://mirrors.estointernet.in/apache/tomcat/tomcat-8/



Index of /apache/tomcat/tomcat-8/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
v8.5.47/	-	2019-Oct-11 10:45

1. click on latest version

Index of /apache/tomcat/tomcat-8/v8.5.47/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
bin/	-	2019-Oct-11 10:45

2. now click on bin

Index of /apache/tomcat/tomcat-8/v8.5.47/bin/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
embed/	-	2019-Oct-11 10:45
extras/	-	2019-Oct-11 10:45
README.html	4.2 KiB	2019-Oct-07 13:54
apache-tomcat-8.5.47-deployer.tar.gz	2.5 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47-deployer.zip	2.5 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47-fulldocs.tar.gz	5.8 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47-windows-x64.zip	11.6 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47-windows-x86.zip	11.4 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47.exe	10.6 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47.tar.gz	9.8 MiB	2019-Oct-07 13:54
apache-tomcat-8.5.47.zip	10.3 MiB	2019-Oct-07 13:54

3. right click on apache tomcat zip and click on copy link address

If there is new version then replace copied path and execute the following command.

```
sudo wget paste copied path here
```

7) Downloaded apache tomcat is available in zip format. So to unzip it. First, we need to install unzip on raspberry pi. To install unzip on raspberry please execute the following command.

```
sudo apt-get install unzip
```

8) Execute the following command to unzip apache tomcat.

```
sudo unzip apache-tomcat-8.5.47.zip
```

9) Now execute the following command to navigate in the bin folder of apache tomcat.

```
cd apache-tomcat-8.5.47/bin/
```

10) Execute the follwoing command to make all .sh files as executeble.

```
sudo chmod 700 *.sh
```

11) Execute the following command to start apache tomcat

```
sudo ./startup.sh
```

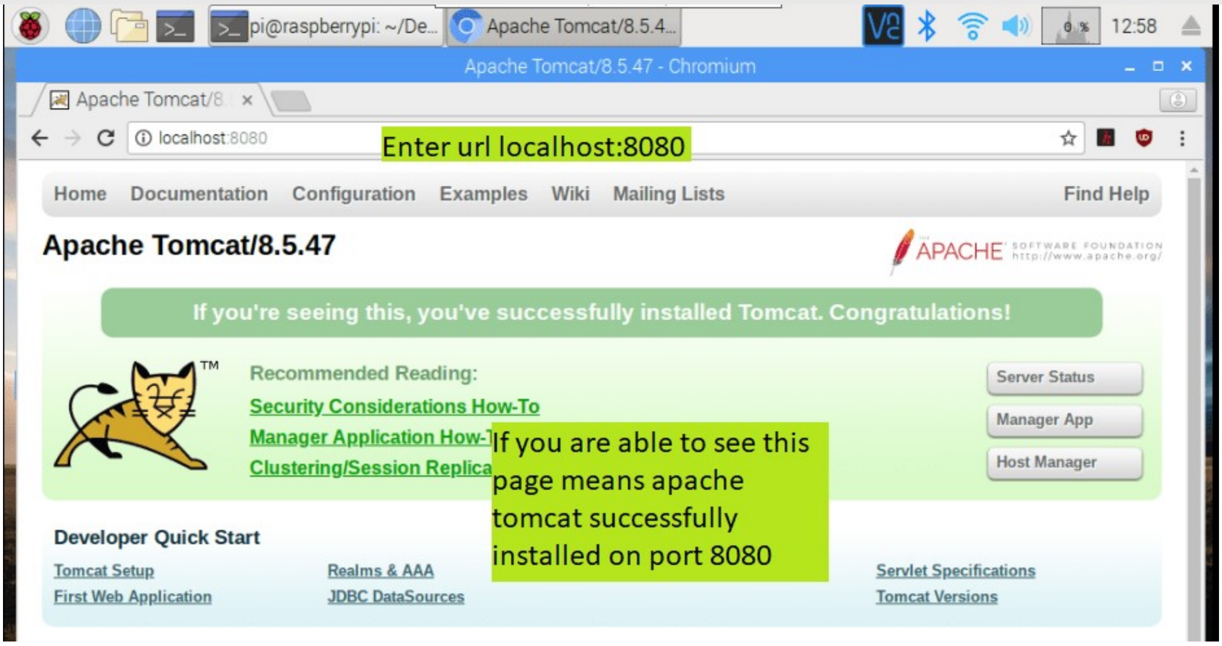
12) Execute the following command to stop apache tomcat

```
sudo ./shutdown.sh
```

13) Now please open any browser on Raspberry pi.

14) Enter the following URL in browser and if you are able to see apache tomcat default page means apache tomcat is successfully installed on raspberry pi.

```
localhost:8080
```



Apache Tomcat/8.5.47 - Chromium

localhost:8080

Enter url localhost:8080

Apache Tomcat/8.5.47

If you're seeing this, you've successfully installed Tomcat. Congratulations!

If you are able to see this page means apache tomcat successfully installed on port 8080

How to Change apache port on raspberry pi

Please follow the following steps to modify the apache tomcat port on raspberry pi. Below we will see how to install apache tomcat on port 80 on raspberry pi.
1)Please open the raspberry pi terminal

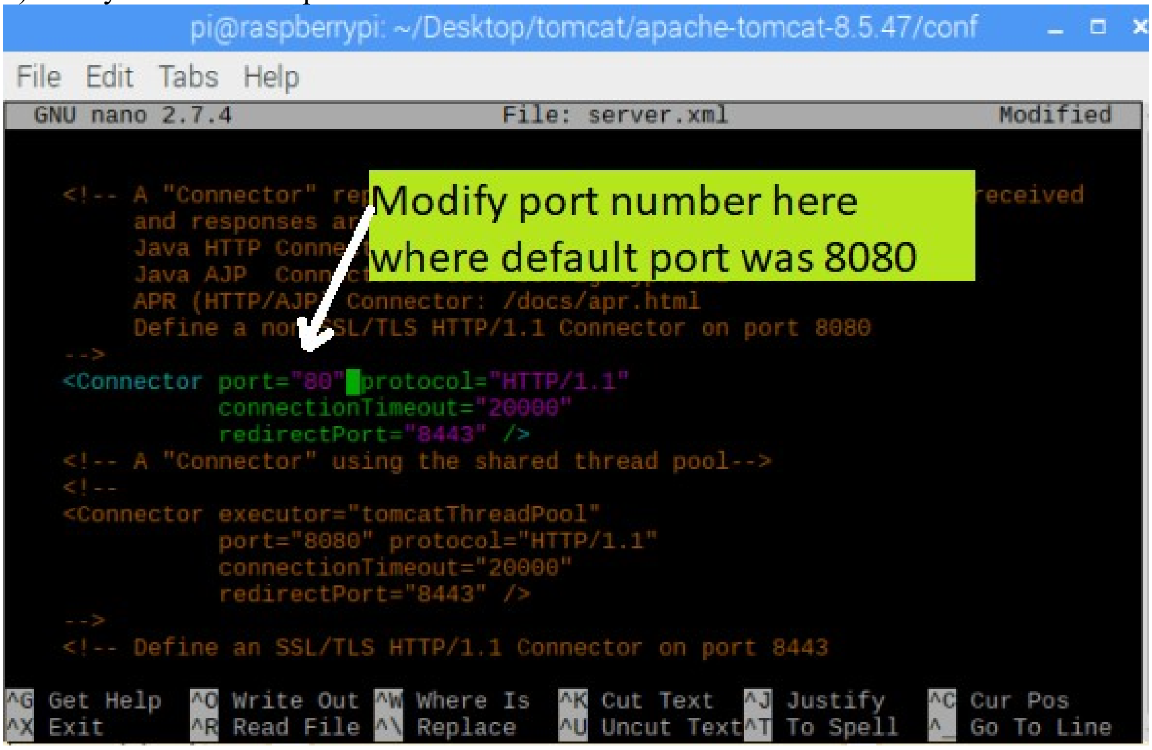
2)Please navigate to conf folder of apache-tomcat-8.5.47

```
cd tomcat/apache-tomcat-8.5.47/conf/
```

3)Now execute the following command to edit server.xml file on raspberry pi

```
sudo nano server.xml
```

4)modify the connector port to 80 which is shown below



Modify port number here where default port was 8080

```
  GNU nano 2.7.4                 File: server.xml                      Modified

    <!-- A "Connector" re                                      received
         and responses ar
         Java HTTP Conne
         Java AJP  Conn
         APR (HTTP/AJP) Connector: /docs/apr.html
         Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

5)Now press the Ctrl+ X and please press y and hit enter to save the changes.

6)Now execute the following command to go in bin folder of apache tomcat.

```
cd ..
cd bin
```

7)Now please execute the following command to stop the apache tomcat.

```
sudo ./shutdown.sh
```

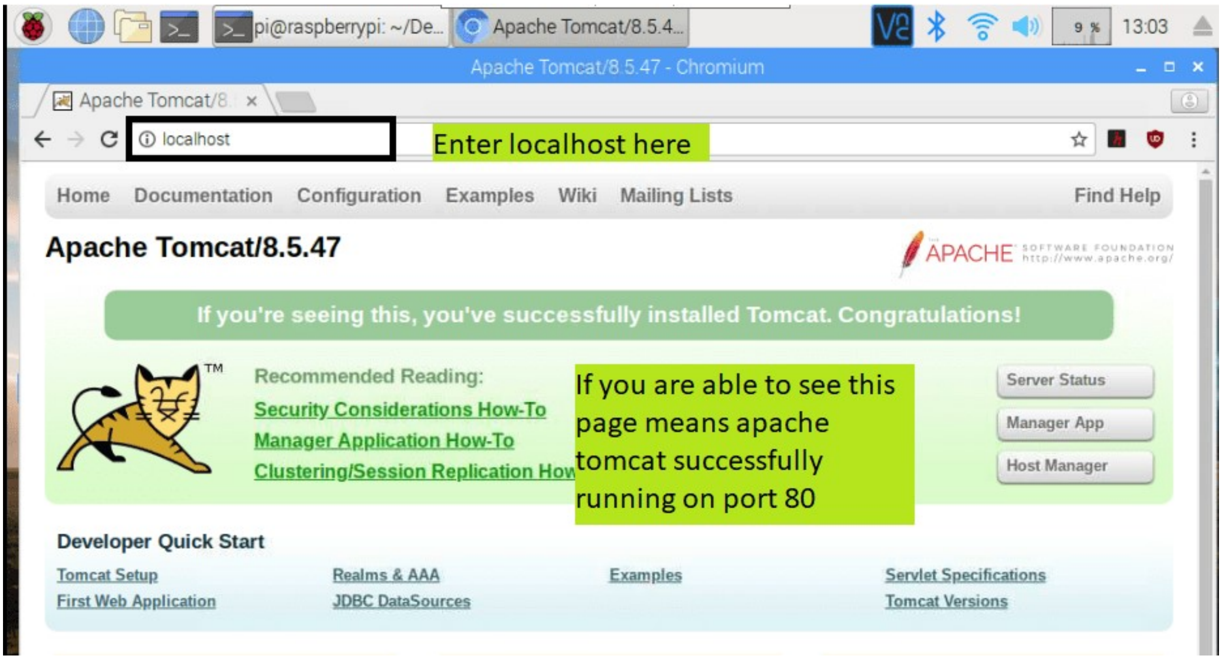
8)Now please execute the following command to start the apache tomcat.

```
sudo ./startup.sh
```

9)Now please open any browser on raspberry pi

10)Enter the following URL in browser and if you are able to see apache tomcat default page means apache tomcat is successfully installed on raspberry pi.

```
localhost
```



Enter localhost here

Apache Tomcat/8.5.47

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How

If you are able to see this page means apache tomcat successfully running on port 80

Developer Quick Start

Tomcat Setup | Realms & AAA | Examples | Servlet Specifications

First Web Application | JDBC DataSources | Tomcat Versions

7. **Design an prototype which measures the amount the gas present inside the working environment using gas sensor in Raspberry pi.**

Aim:

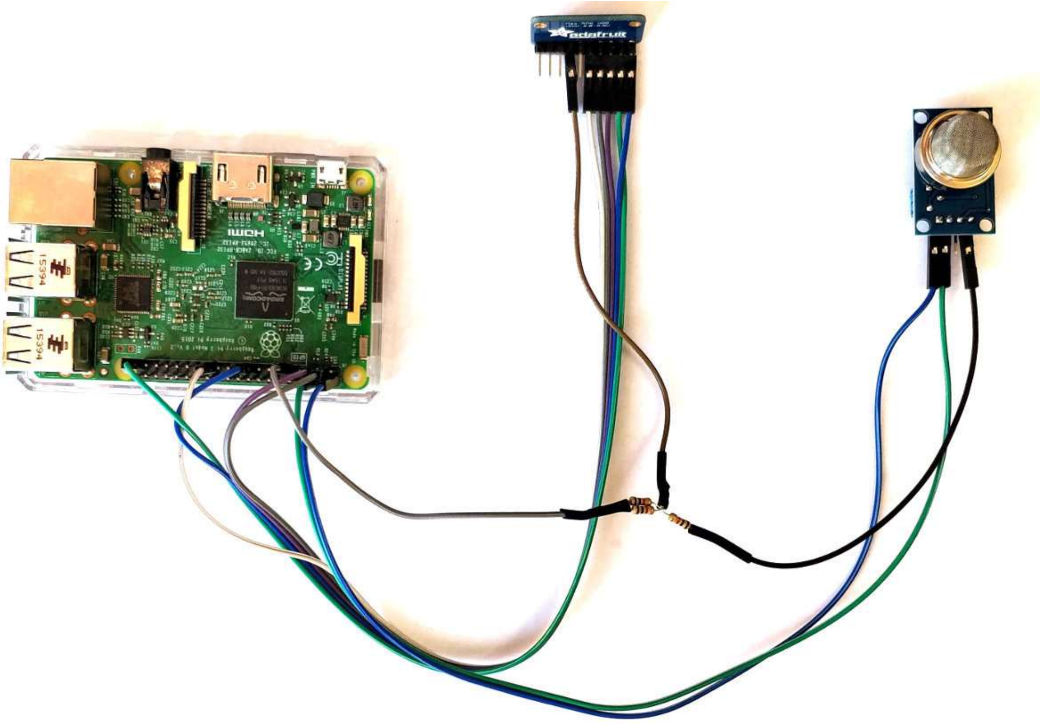
To design a prototype to measure the amount of gas using gas sensor in Raspberry pi

Procedure:

i. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows
https://pimylifeup.com/raspberry-pi-gpio/

ii. Open the python IDE and enter the code
iii. Declare the GPIO pin for gas sensor
iv. Based on the sensor output show the amount of gas present inside the room.
v. Display the output on the Pi Board.

Circuit Diagram:



Program:

```
#!/usr/bin/python
#encoding:utf-8
import time
import board
import busio
import adafruit_ads1x15.ads1015 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)
# Create the ADC object using the I2C bus
ads = ADS.ADS1015(i2c)
ads.gain = 8
# Create single-ended input on channel 0
chan = AnalogIn(ads, ADS.P0)
```

```
# Create differential input between channel 0 and 1
#chan = AnalogIn(ads, ADS.P0, ADS.P1)
print("{:>5}\t{:>5}".format('raw', 'v'))

while True:
    print("{:>5}\t{:>5.5f}".format(chan.value, chan.voltage))
    time.sleep(0.5)
```

8. Build the application to monitor the environment using temperature sensor in Raspberry pi board.

Aim:

To build the application to monitor the environment using temperature sensor of DS18B20 in Raspberry pi board.

Procedure:

i. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows
https://pimylifeup.com/raspberry-pi-gpio/

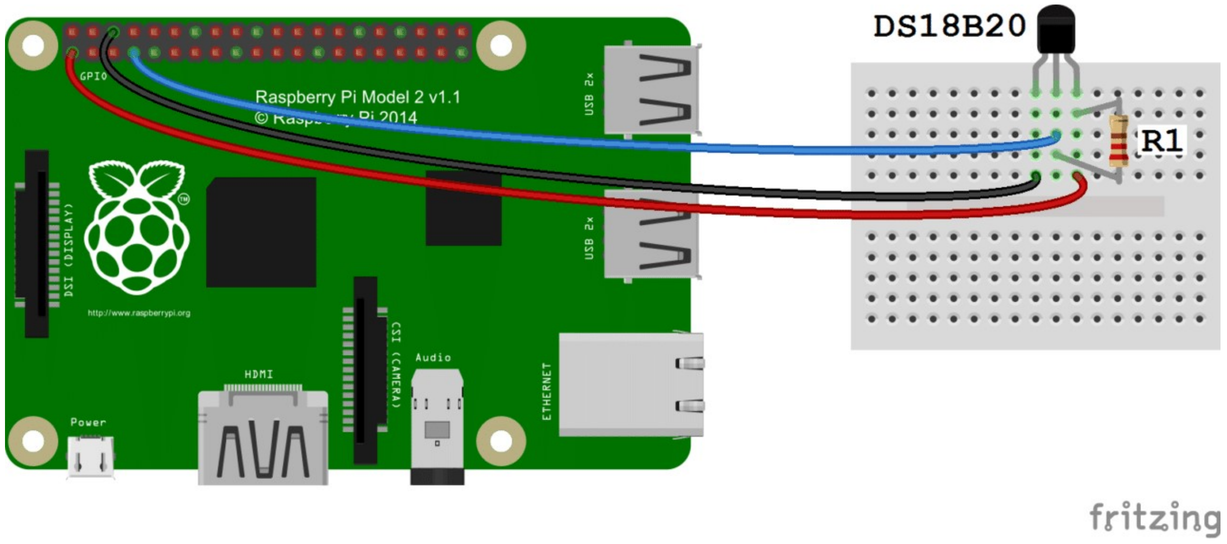
ii. Open the python IDE and enter the code

iii. Declare the GPIO pin for temperature sensor

iv. Based on the sensor output show the temperature sensor

v. Display the output on the Pi Board.

Circuit:



Program:

```
import os
import glob
import time
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines
def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f
while True:
    print(read_temp())
    time.sleep(1)
```

Output:

Save the File with

Filename.py

Run the code with

Sudo filename.py