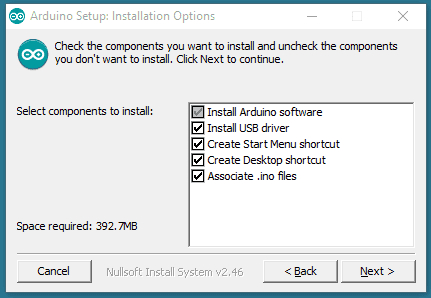
## Download the Arduino Software (IDE)

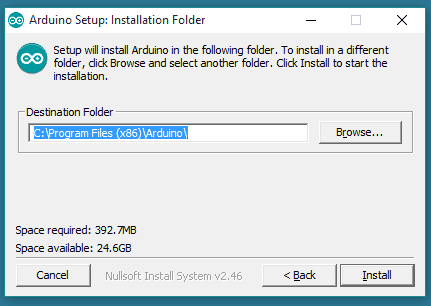
Get the latest version from

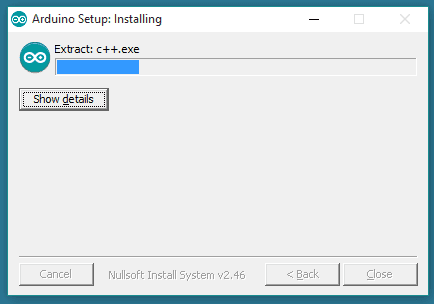
<https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE>.

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](https://arduino.cc/en/Guide/PortableIDE).

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

[](https://docs.arduino.cc/static/33f20406f68c5707052471d78a90a5c6/9cb4e/DRV_Capture1.png)Choose the components to install.

[](https://docs.arduino.cc/static/12311f50263afe3f12349d932fdeb3f5/9cb4e/DRV_Capture2.png)Choose the installation directory.

[](https://docs.arduino.cc/static/02501558f5cba4564376f0bb8adfcf01/ade6e/DRV_Capture3.png)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:**

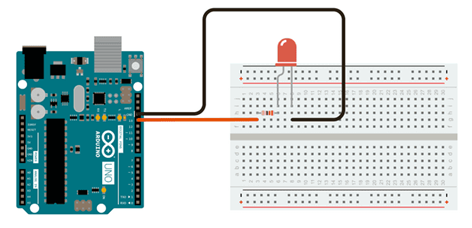
To make a LED Blink operation using Arduino

**Procedure:**

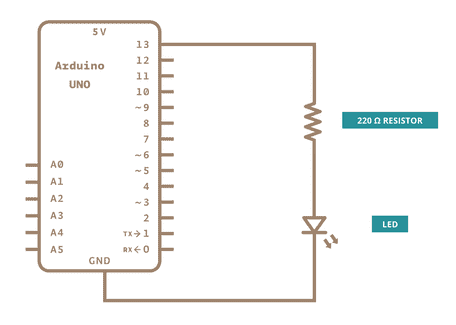
1. Start the Arduino IDE and enter the code
2. Declare the variable for input pin
3. Initialize digital pin 13 as an output in void setup function
4. The loop function runs over and over again forever
5. Turn the LED on by making the voltage HIGH
6. Turn the LED on by making the voltage LOW

**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:**

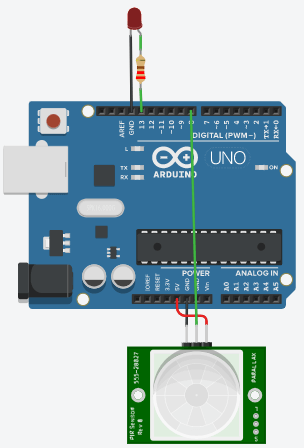
To find Motion detection using Arduino board.

**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the PIR sensor output variable
3. Define the baud rate for serial transmission
4. The loop function runs over and over again forever
5. Based on the sensor output show the movement.
6. If the PIR sensor value is 1, then print the statement of movement detection
7. 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);

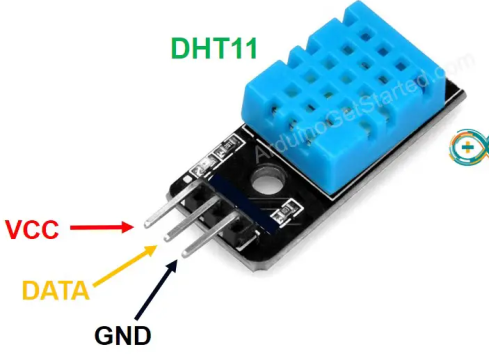
}

}

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

**Aim:**

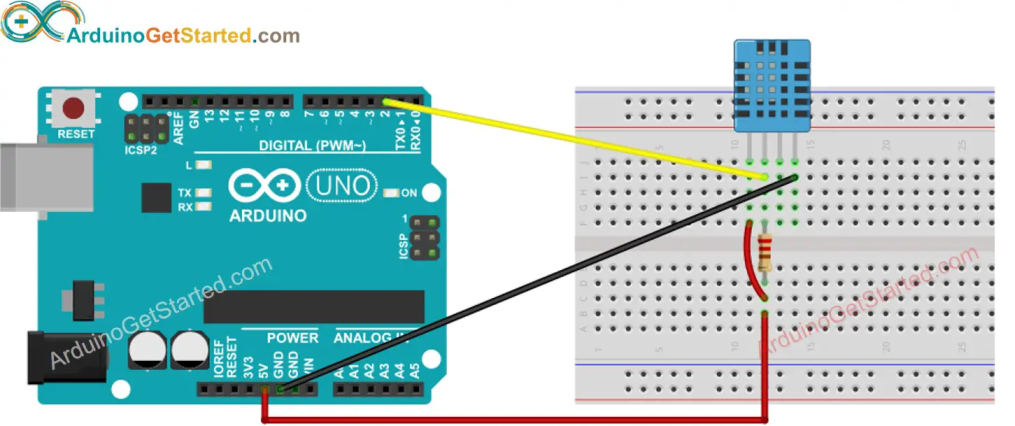
To measure the temperature and humidity using Arduino board with DHT 11 Sensor.



**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the pin for temperature and humidity
3. Define the baud rate for serial transmission
4. The loop function runs over and over again forever
5. Based on the sensor output show the temperature and humidity
6. 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);

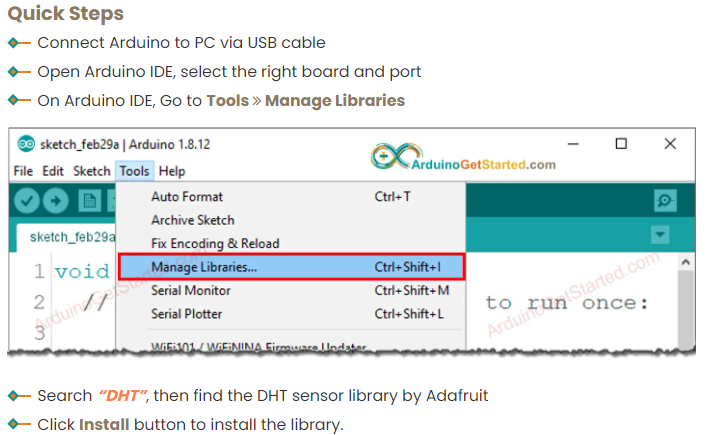
Serial.print("°C ~ ");

Serial.print(tempF);

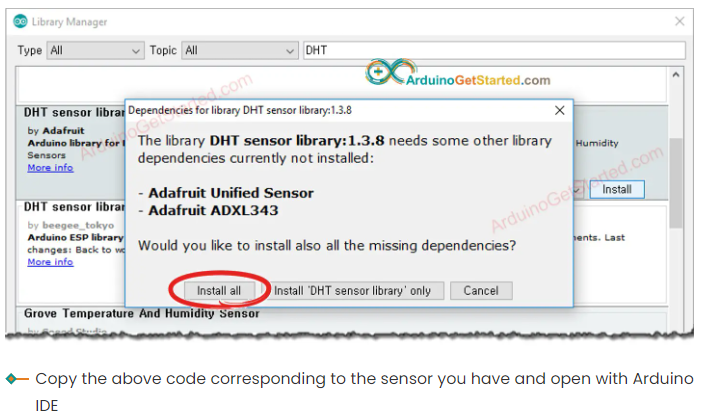
Serial.println("°F");

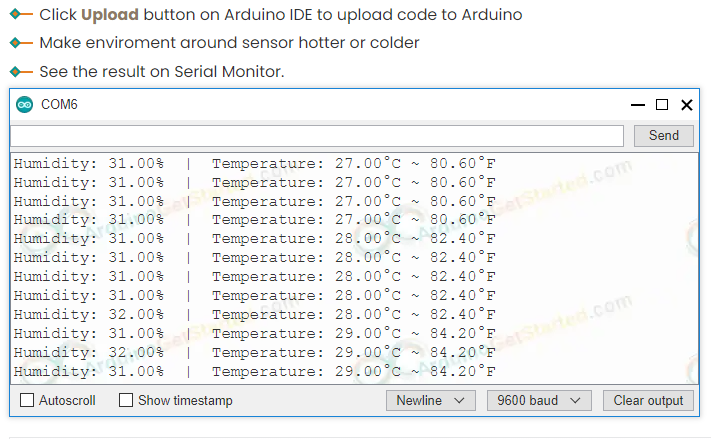
}

}

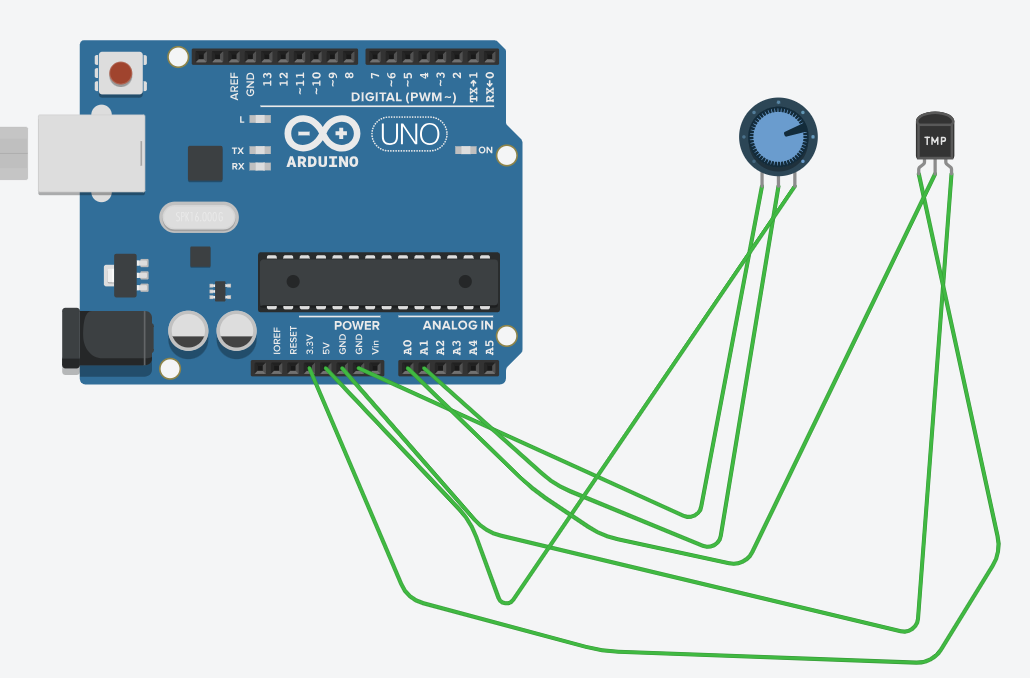








**In TinkerCad**



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.**

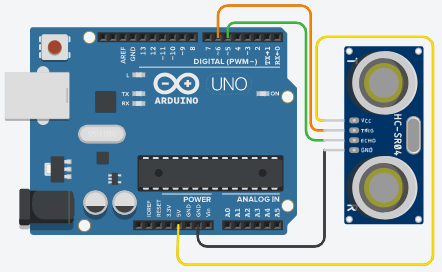
**Aim:**

To find object distance detection using Arduino board.

**Procedure:**

1. Start the Arduino IDE and enter the code
2. Declare the pin for echo and trigger
3. Define the baud rate for serial transmission
4. The loop function runs over and over again forever
5. Based on the sensor output show the distance between the object and sensor
6. 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.

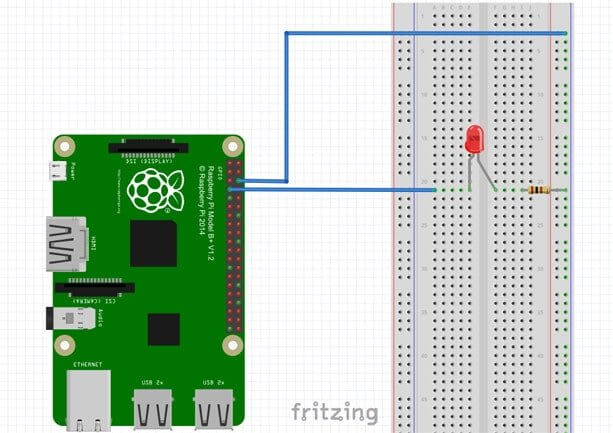
**Procedure:**

1. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows

<https://pimylifeup.com/raspberry-pi-gpio/>

1. Open the python IDE and enter the code
2. Declare the GPIO pin for LED
3. Based on the sensor output show the LED Blink.
4. 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.

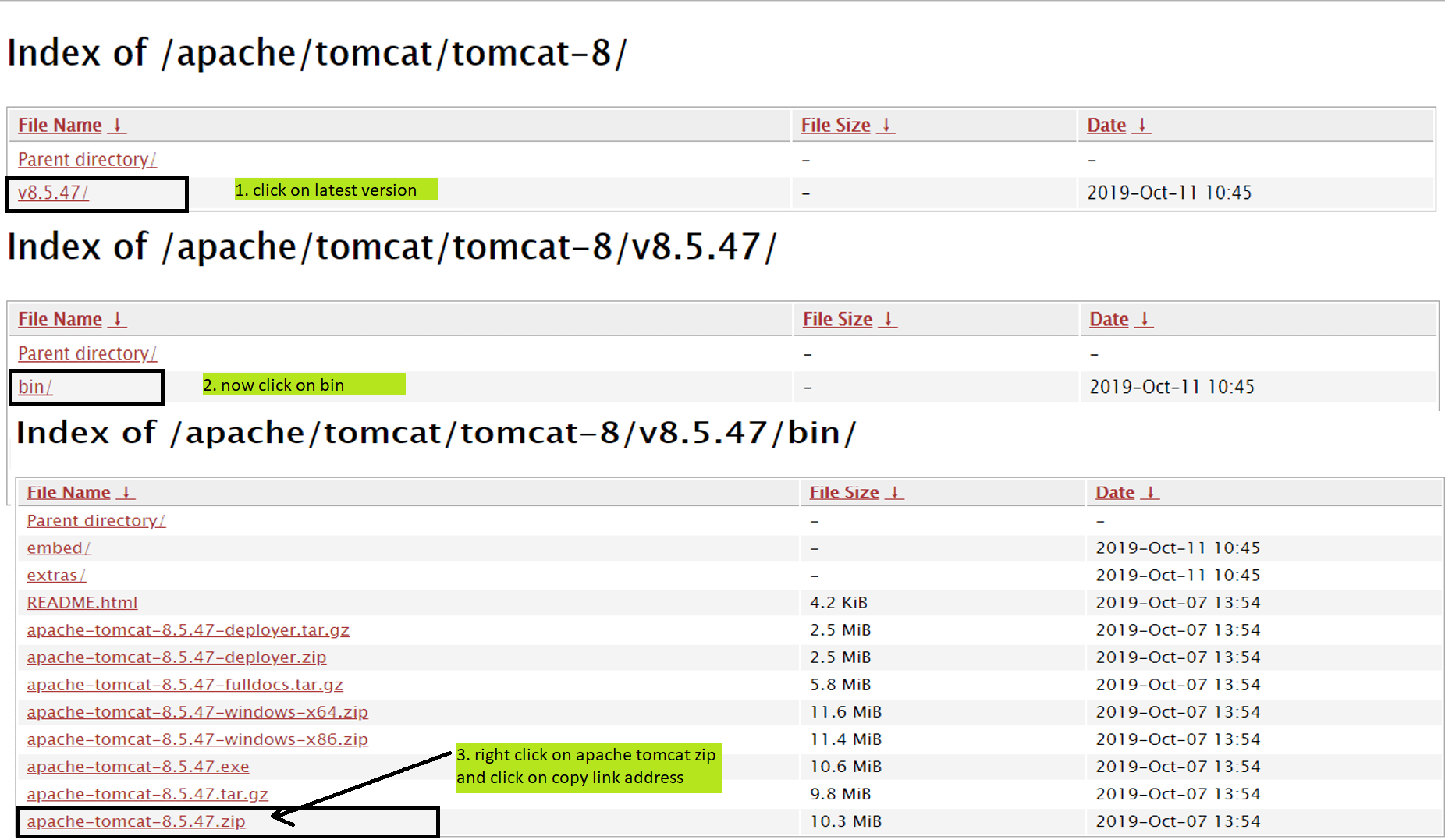
**cd tomcat**

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/**

  
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**

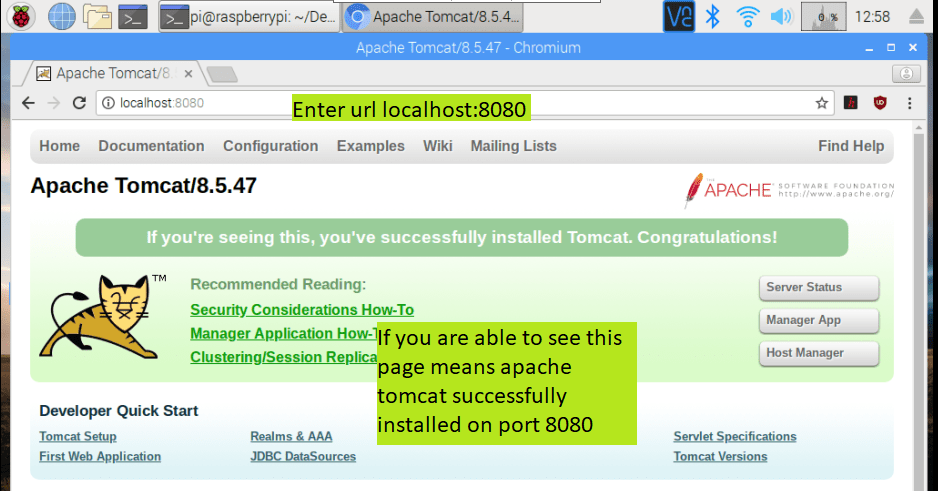
12) Execute the following command to stop apache tomcat

**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**



### **How to Change apache port on raspberry pi**

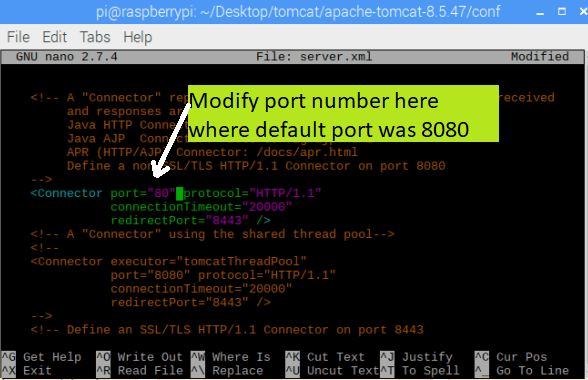
Please follow the following steps to modify the apache tomcat port on raspberry pi. Below we will see how to install apache tomcat on port 80 on raspberry pi.  
1)Please open the raspberry pi terminal

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  
  
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**

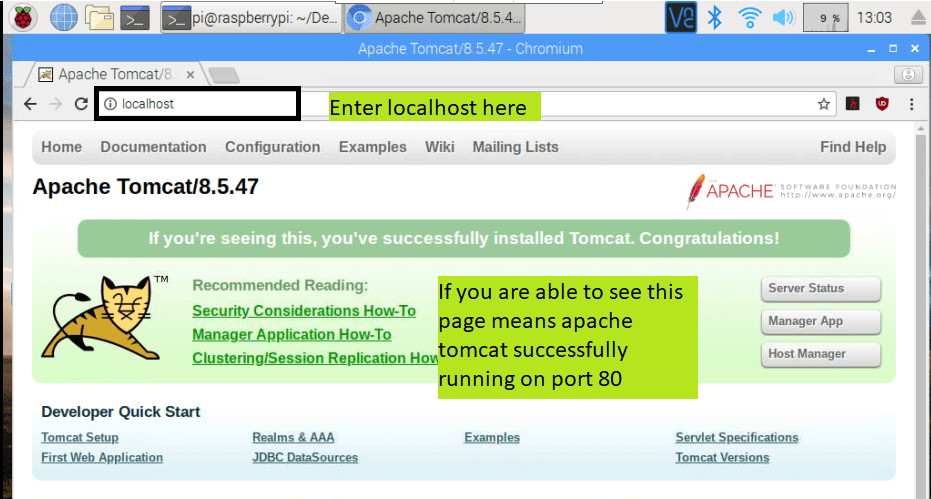
8)Now please execute the following command to start the apache tomcat.

**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**



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

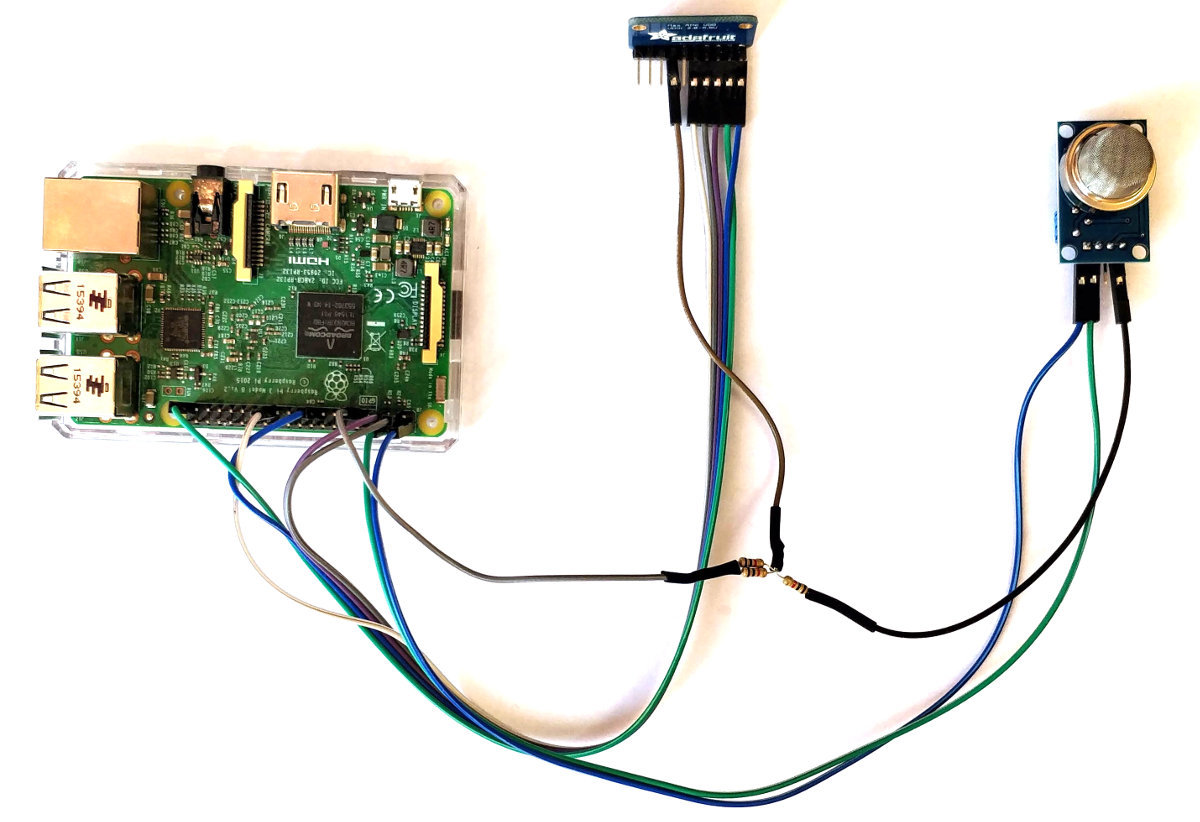
**Procedure:**

1. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows

<https://pimylifeup.com/raspberry-pi-gpio/>

1. Open the python IDE and enter the code
2. Declare the GPIO pin for gas sensor
3. Based on the sensor output show the amount of gas present inside the room.
4. 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.

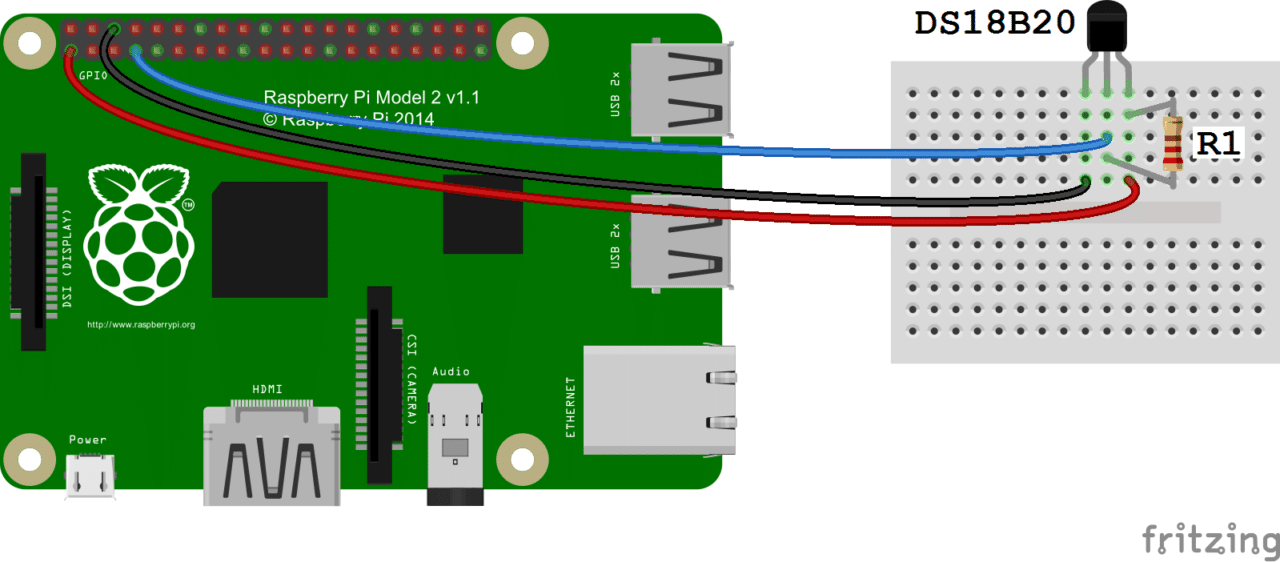
**Procedure:**

1. Install the NOOBS software on the Pi Board and follows the instruction from the link as follows

<https://pimylifeup.com/raspberry-pi-gpio/>

1. Open the python IDE and enter the code
2. Declare the GPIO pin for temperature sensor
3. Based on the sensor output show the temperature sensor
4. 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**