**Experiment 1**

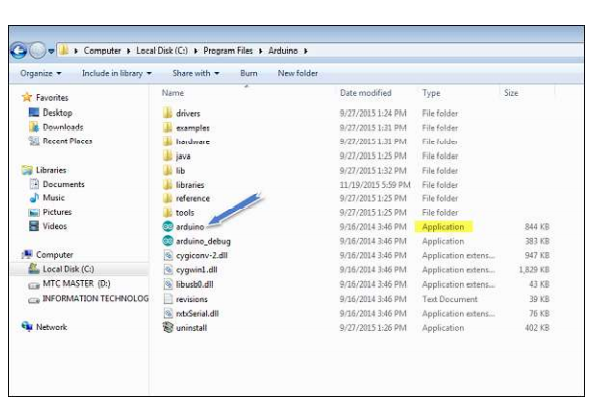
**Aim: Familiarization with Arduino/Raspberry Pi and perform necessary software installation.**

**Arduino Installation**

Step 1: Download Arduino from “https://www.arduino.cc ”

Step 2: Launch Arduino IDE.

Step 3: After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.



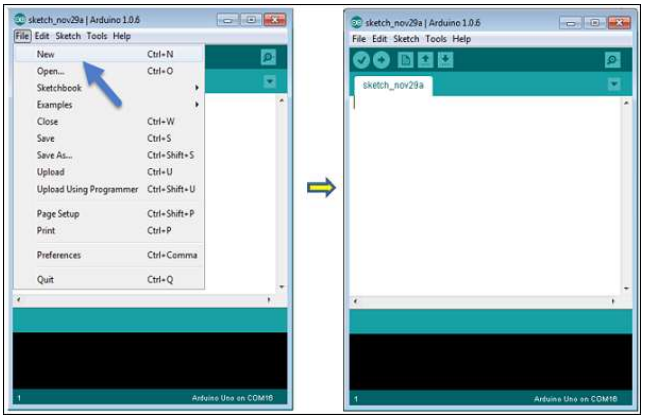
Step 4: Open your first project.

Once the software starts, you have two options −

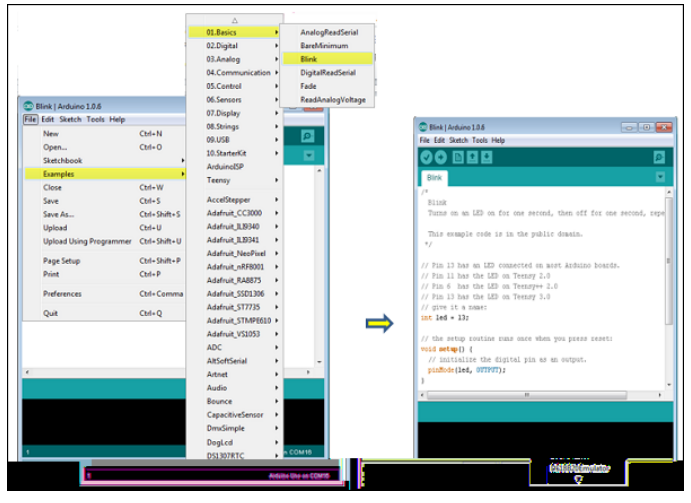
• Create a new project.

• Open an existing project example.

To create a new project, select File → New.



To open an existing project example, select File → Example → Basics → Blink.

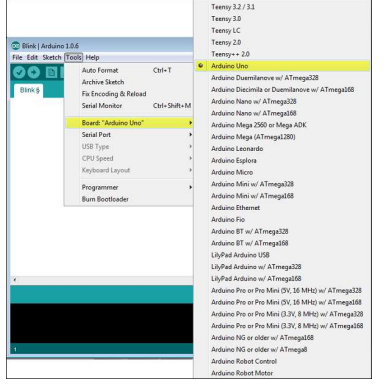


Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 5: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools → Board and select your board.

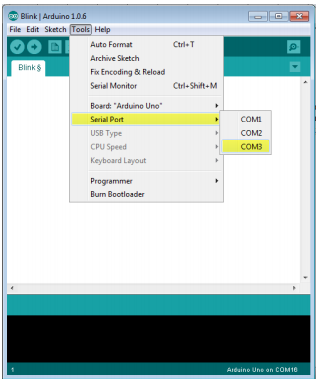
****

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

Step 7 − Select your serial port.

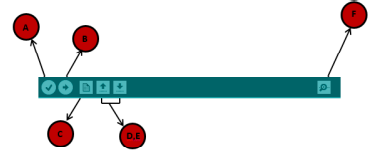
Select the serial device of the Arduino board.

Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

****

Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

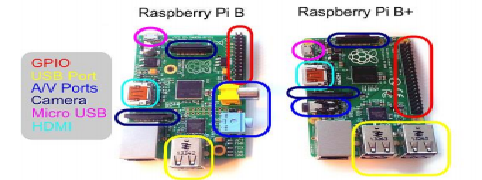
F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

**Raspberry Pi Installation**

**REQUIRED ITEMS**

* A Raspberry Pi (Either a Model B or Model B+)

****

* SD Card

o We recommend an 8GB class 4 SD card.

* Display and connecting cables

o Any HDMI/DVI monitor or TV should work as a display for the Pi.

o For best results, use one with HDMI input, but other connections are available for older devices.

* Keyboard and mouse

o Any standard USB keyboard and mouse will work with your Raspberry Pi.

* Power supply

o Use a 5V micro USB power supply to power your Raspberry Pi. Be careful that whatever power supply you use outputs at least 5V; insufficient power will cause your Pi to behave unexpectedly.

* Internet connection

o To update or download software, we recommend that you connect your Raspberry Pi to the internet either via an Ethernet cable or a WiFi adaptor.

* Sound

o Headphones, earphones or speakers with a 3.5mm jack will work with your Raspberry

**PLUGGING IN YOUR RASPBERRY PI**

1. Begin by slotting your SD card into the SD card slot on the Raspberry Pi, which will only fit one way.

2. Next, plug in your USB keyboard and mouse into the USB slots on the Raspberry Pi. Make sure that your monitor or TV is turned on, and that you have selected the right input (e.g. HDMI 1, DVI, etc).

3. Then connect your HDMI cable from your Raspberry Pi to your monitor or TV.

4. If you intend to connect your Raspberry Pi to the internet, plug in an Ethernet cable into the Ethernet port next to the USB ports; if you do not need an internet connection, skip this step.

5. Finally, when you are happy that you have plugged in all the cables and SD card required, plug in the micro USB power supply. This action will turn on and boot your Raspberry Pi.

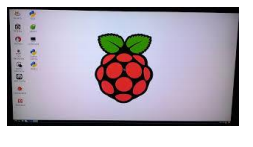
6. If this is the first time your Raspberry Pi SD card have been used, then you will have to select an operating system and configure it.

**LOGGING INTO YOUR RASPBERRY PI**

1. Once your Raspberry Pi has completed the boot process, a login prompt will appear. The default login for Raspbian is username pi with the password raspberry. Note you will not see any writing appear when you type the password. This is a security feature in Linux.

2. After you have successfully logged in, you will see the command line prompt pi@raspberrypi~$.

3. To load the graphical user interface, type startx and press Enter on your keyboard.



**DOWNLOAD AND INSTALL WIRING PI**

WiringPi is maintained under GIT for ease of change tracking, however there is a Plan B if you’re unable to use GIT for whatever reasons (usually your firewall will be blocking you, so do check that first!)

**ONLINE INSTALL**

If you do not have GIT installed, then under any of the Debian releases (e.g. Raspbian), you can install it with:

**sudo apt-get install git-core**

If you get any errors here, make sure your Pi is up to date with the latest versions of Raspbian:

**sudo apt-get update**

**sudo apt-get upgrade**

To obtain WiringPi using **GIT:**

**git clone git://git.drogon.net/wiringPi**

If you have already used the clone operation for the first time, then

**cd wiringPi**

**git pull origin**

Will fetch an updated version then you can re-run the build script below.

To build/install there is a new simplified script:

**cd wiringPi**

**./build**

The new build script will compile and install it all for you – it does use the sudo command at one point, so you may wish to inspect the script before running it.

**TESTING SERIAL PORT IN RASPBERRY PI**

A great way to test out the serial port is to use the minicom program. If you dont have this installed run

**sudo apt-get install minicom**

Connect your PC to the Raspberry Pi serial port using an appropriate serial port adapter and wiring, then open Putty or a similar serial terminal program on PC side. Setup a connection using the serial port at 9600 baud.

Now run up minicom on the Raspberry Pi using

**minicom -b 9600 -o -D /dev/ttyAMA0**

What you type into the minicom terminal screen should appear on the serial PC terminal and vice versa.

**Experiment 2**

**Aim:To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.**

**Requirements:**

1. Standard 5mm LED
2. One Resistor
3. Arduino Uno microcontroller board (ATmega328p)
4. Jumper wires

**Procedure:**

Step 1: Open [www.wokwi.com](http://www.wokwi.com) in browser and select “Arduino Uno” microcontroller.

Step 2: In the Simulation part, select the above list by clicking “+” symbol, which are specified in above requirements.

Step 3: By using jumper wires, Connect Anode (A-pin) of LED to digital pin 3 of Arduino microcontrollerthrough the resistor and Cathode (C-pin) of LED to ground.

Step 4: Write program in “sketch.ino”.

**Program:**

voidsetup() {

  // put your setup code here, to run once:

pinMode(3, OUTPUT);

}

voidloop() {

  // put your main code here, to run repeatedly:

digitalWrite(3,HIGH); //we can also write digitalWrite(3,1);

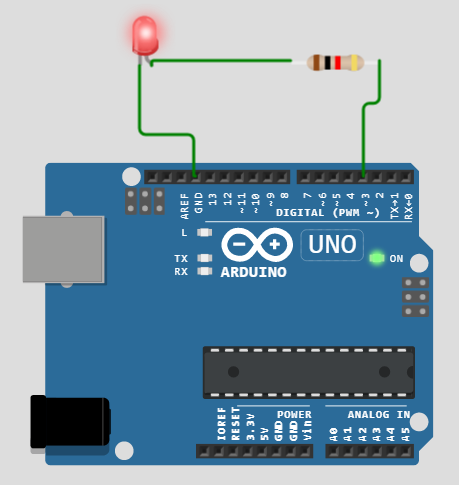
delay(1000);

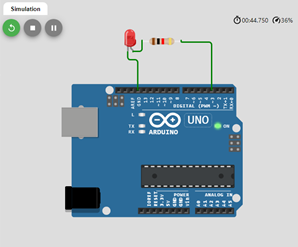
digitalWrite(3,LOW);//we can also write digitalWrite(3,0);

delay(1000);

}

**Output:**LED will turn on for 1 sec after every 2 seconds by interfacing LED with Arduino.





**Experiment 3**

**Aim: To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.**

**Requirements:**

1. Standard 5mm LED
2. Two Resistors
3. Arduino Uno microcontroller board (ATmega328p)
4. Jumper wires
5. Push button

**Procedure:**

Step 1: Open [www.wokwi.com](http://www.wokwi.com) in browser and select “Arduino Uno” microcontroller.

Step 2: In the Simulation part, select the above list by clicking “+” symbol, which are specified in above requirements.

Step 3: By using jumper wires, Connect Anode (A-pin) of LED to digital pin 3 of Arduino microcontroller through the resistor1 and Cathode (C-pin) of LED to ground.

Step 4: Give 5volt power supply to pushbutton 1:1L pin.

Step 5: Connect pushbutton 1:1R to both GND through resistor2& digital pin5.

Step 6: Write program in “sketch.ino”.

**Program:**

int flag=0;

voidsetup() {

  // put your setup code here, to run once:

pinMode(3, OUTPUT);

pinMode(5, INPUT);

}

voidloop() {

  // put your main code here, to run repeatedly:

int value=digitalRead(5);

if(value==1&& flag==0)

{

  digitalWrite(3,HIGH);

  delay(1000);

  flag=1;

}

elseif(value==1&& flag==1)

{

  digitalWrite(3,LOW);

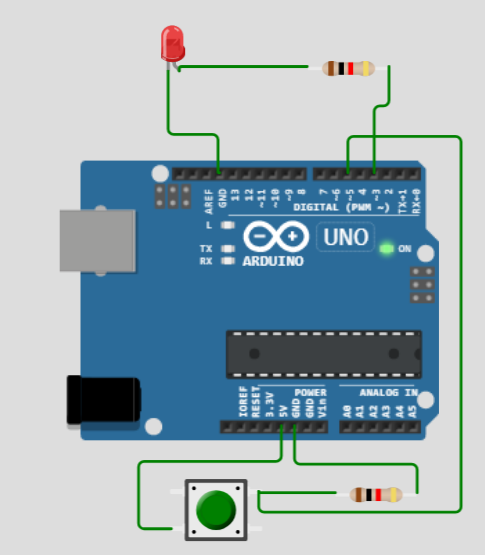
  delay(1000);

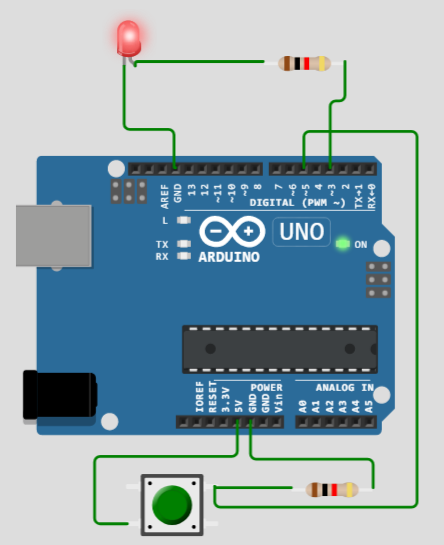
  flag=0;

}

}

**Output:**When push button is pressed, LED will be turned ON by interfacing Push button with Arduino.





**Experiment 4**

**Aim:To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings**

**Requirements:**

1. DHT22 sensor
2. Arduino Uno microcontroller board (ATmega328p)
3. Jumper wires

**Procedure:**

Step 1: Open [www.wokwi.com](http://www.wokwi.com) in browser and select “Arduino Uno” microcontroller.

Step 2: In the Simulation part, select the above list by clicking “+” symbol, which are specified in above requirements.

Step 3:

By using jumper wires,

1. Give 5 volts power to pin dht1:VCC of DHT22 sensor
2. Connect dht1:SDA pin of DHT22 sensor to digital pin3 of Arduino Uno microcontroller board (ATmega328p)
3. Connect dht1: GND pin of DHT22to ground of microcontroller board.

Step 4: Write program in “sketch.ino”.

**Program:**

#include<Adafruit\_Sensor.h>

#include<DHT.h>

#include<DHT\_U.h>

DHT Sensor(3,DHT22);

voidsetup() {

  // put your setup code here, to run once:

Sensor.begin();

**Serial**.begin(9600);

}

voidloop() {

  // put your main code here, to run repeatedly:

floattempr = Sensor.readTemperature();

**Serial**.print("Temperature = ");

**Serial**.print(tempr);

**Serial**.print(", ");

delay(1000);

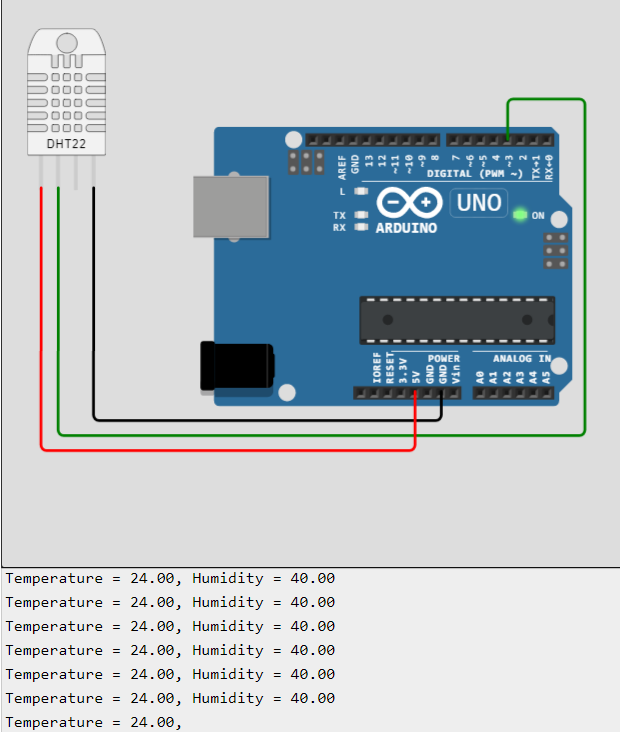
float humid = Sensor.readHumidity();

**Serial**.print("Humidity = ");

**Serial**.println(humid);

}

**Output:**Temperature and Humidity readings are display by interfacing DHT22 sensor with Arduino.



**Experiment 5**

**Aim: To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.**

**Requirements:**

1. DHT22 sensor
2. Arduino Uno microcontroller board (ATmega328p)
3. Jumper wires
4. SSD1306 OLED display

**Procedure:**

Step 1: Open [www.wokwi.com](http://www.wokwi.com) in browser and select “Arduino Uno” microcontroller.

Step 2: In the Simulation part, select the above list by clicking “+” symbol, which are specified in above requirements.

Step 3:

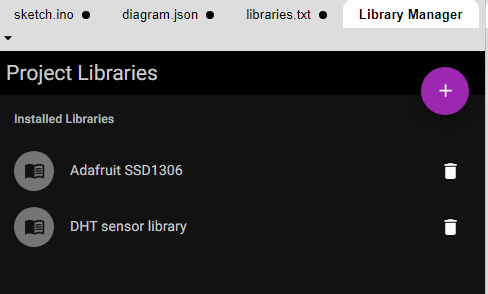
By using jumper wires,

1. Give 5 volts power to pin dht1:VCC
2. Connect dht1: SDA to digital pin3 of Arduino Uno microcontroller board (ATmega328p)
3. Connect dht1: GND to ground.
4. Make connections between

* Oled1:DATA toAnalog pin A4
* Oled1:CLK toAnalog pin A5
* Oled1:VIN to 5 volt power
* Oled1:GND to ground

Step 4: Write program in “sketch.ino”.

**Note:**Download libraries Adafruit SSD1306 and DHT sensor library.



**Program:**

#include<Adafruit\_Sensor.h>

#include<Adafruit\_GFX.h>

#include<Adafruit\_SSD1306.h>

#include<DHT.h>

#include<DHT\_U.h>

#include<Wire.h>

Adafruit\_SSD1306 display(128,64,&**Wire**);

DHT Sensor(3,DHT22);

voidsetup() {

  // put your setup code here, to run once:

Sensor.begin();

**Serial**.begin(9600);

display.begin(SSD1306\_SWITCHCAPVCC,0X3C);

display.setTextColor(WHITE);

}

voidloop() {

  // put your main code here, to run repeatedly:

floattempr= Sensor.readTemperature();

float humid= Sensor.readHumidity();

display.setCursor(0,10);

display.setTextSize(1);

display.print("Temperature= ");

display.println(tempr);

display.print("Humidity= ");

display.println(humid);

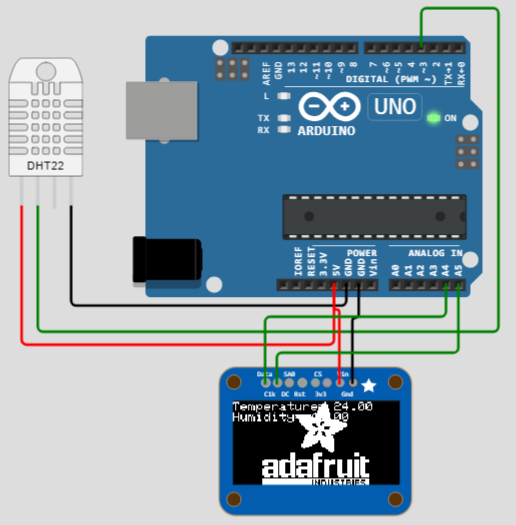
display.display();

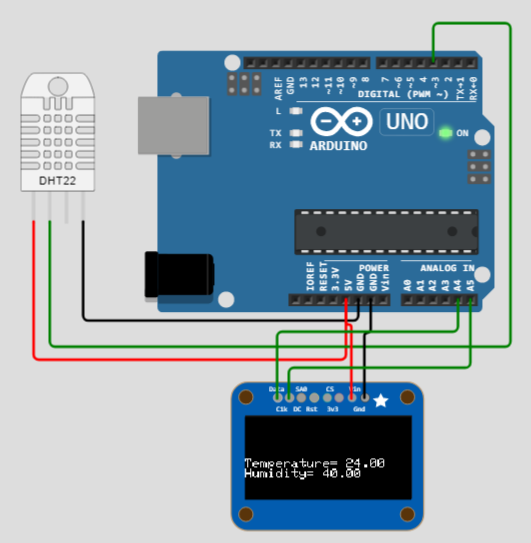
delay(500);

display.clearDisplay();

}

**Output**: Temperature and Humidity readings are display on OLED by interfacing DHT22 sensor with Arduino.

****

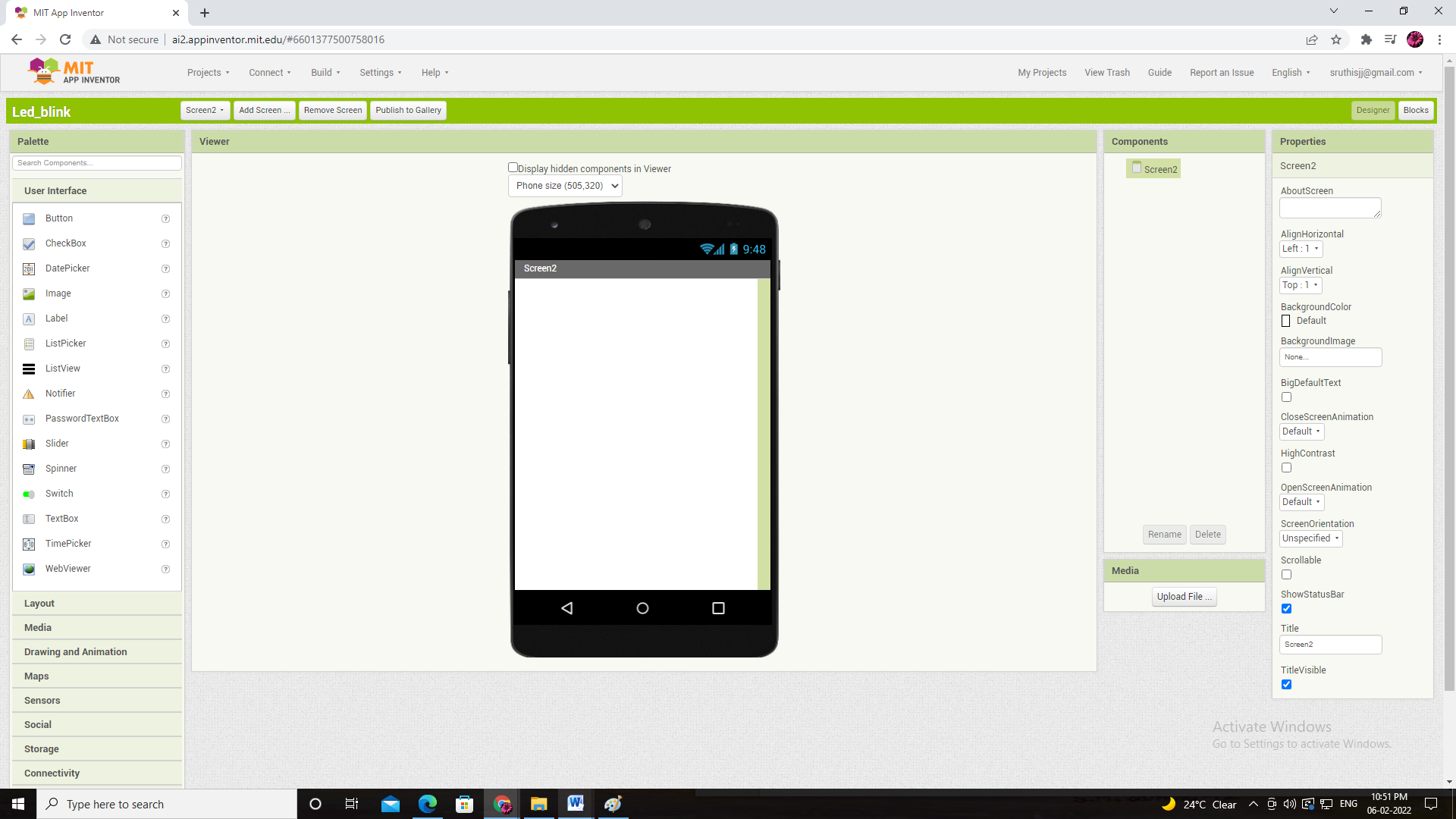
****

**Experiment 6**

**Aim: To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.**

Step 1: Open “MIT App Inventor” in Google Chrome

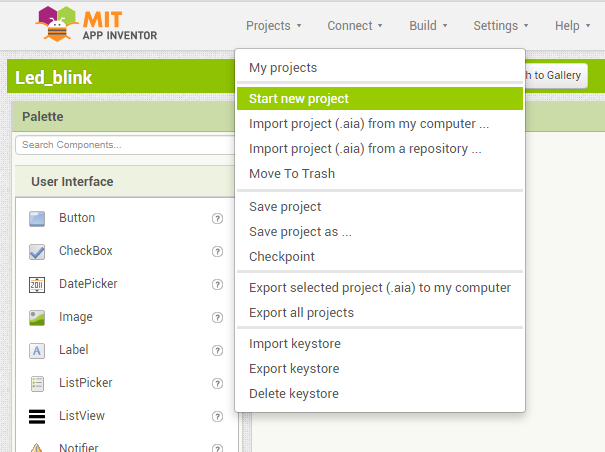
Step 2: Click on “Create Apps!” button

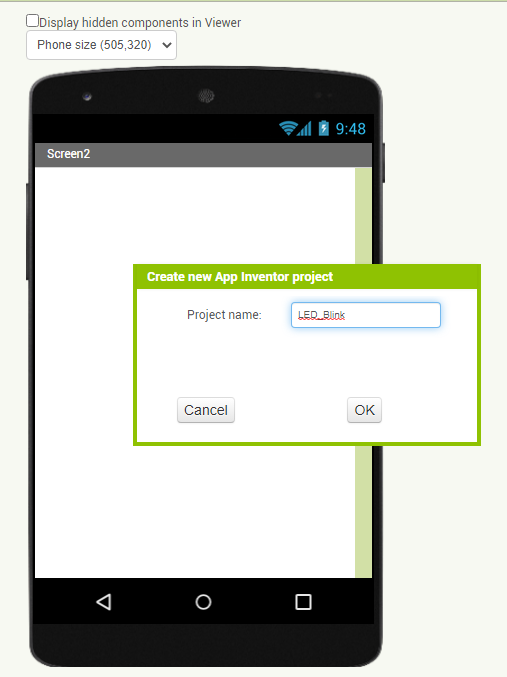


Step 3: Click on Projects.

Step 4: In drop down menu, select “Start new project”.

Step 5: Enter Project Name :“LED\_Blink” and press on OK button.





Step 6: Select “Label” option in “User Interface” menu and drag it on to themobile screen image.

1. Set Width is Fill Parent,
2. TextAlignment is Center:1,
3. Type “This App is made for Bluetooth base control” in Text box.

Step 7: Select “ListPicker” option in “User Interface” menuand drag it on to the mobile screen image.

* Type “Connect to Device” in Text box.

Step 8: Select “Button” option in “User Interface” menu and drag it on to the mobile screen image.

* Type “On” in Text box to change the text for Button1

Step 9: Select “Button” option again in “User Interface” menu and drag it on to the mobile screen image.

* Type “Off” in Text box to change the text for Button2.

Step 10: Select “Label” option in “User Interface” menu and drag it on to the mobile screen image.

* Type “Delete Everything” in Text box.

Step 11: Select “Clock” option in “Sensor” menu and drag it on to the mobile screen image

Step 12: Click on “Connectivity” option and select “BluetoothClient” option and drag it on to the mobile screen image.

1. Click on “Blocks” button.

a)

* click on “ListPicker1” option and select “when ListPicker1 .AfterPicking do”
* click on “Control option” and select “if then” option.
* click on “BluetoothClient1” and select “call BluetoothClient1 .Connect address” option.
* click on “ListPicker1” and select “ListPicker1 . Selection” option.
* click on “ListPicker1” and select “set ListPicker1 . Elements to” option.
* click on BluetoothClient1” and select “BluetoothClient1 . AddressesAndNames” option.

b)

* click on “ListPicker1” option and select “when ListPicker1 .BeforePicking do”
* click on “ListPicker1” and select “set ListPicker1 . Elements to” option.
* click on BluetoothClient1” and select “BluetoothClient1 . AddressesAndNames” option.

c)

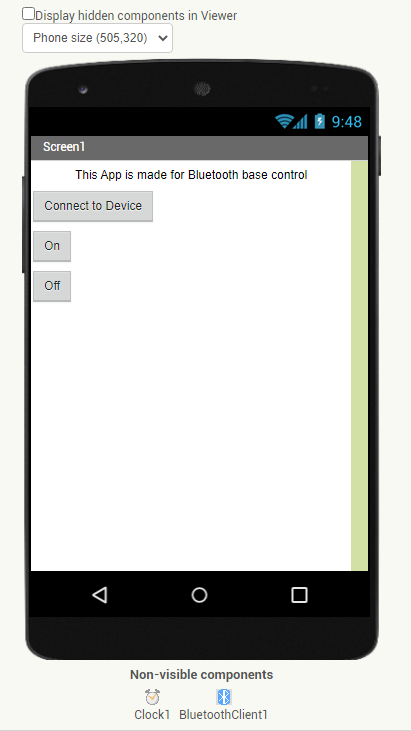
* click on “Button1” and select “when Button1 .Click do” option.
* click on “BluetoothClient1” and select “call BluetoothClient1 .SenText text” option.
* Click on “Text” option for select a “text String” and enter “1” into it.

d) Repeat above step and change it as “Button2” and set Text as 0.

e)

* Click on “Clock1” and select “when Clock1 . Timer do”
* Click on “Control” option for selecting “if then” option
* Click on “BluetoothClient1” for selecting “BluetoothClient1 .IsConnected”
* Click on “Label2” option for selecting “set Label2 . Text to”
* Click on “BluetoothClient1” for selecting “call BluetoothClient1 .ReceiveTextnumberOfBytes” and “call BluetoothClient1 .BytesAvailableToReceive” option.

**Output:**



**Experiment 7**

**Aim:To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when ‘1’/’0’ is received from smartphone using Bluetooth**

Step 1: Open “MIT App Inventor” in Google Chrome

Step 2: Click on “Create Apps!” button

Step 3:

Click on “Blocks” button.

a)

* click on “ListPicker1” option and select “when ListPicker1 .AfterPicking do”
* click on “Control option” and select “if then” option.
* click on “BluetoothClient1” and select “call BluetoothClient1 .Connect address” option.
* click on “ListPicker1” and select “ListPicker1 . Selection” option.
* click on “ListPicker1” and select “set ListPicker1 . Elements to” option.
* click on BluetoothClient1” and select “BluetoothClient1 . AddressesAndNames” option.

b)

* click on “ListPicker1” option and select “when ListPicker1 .BeforePicking do”
* click on “ListPicker1” and select “set ListPicker1 . Elements to” option.
* click on BluetoothClient1” and select “BluetoothClient1 . AddressesAndNames” option.

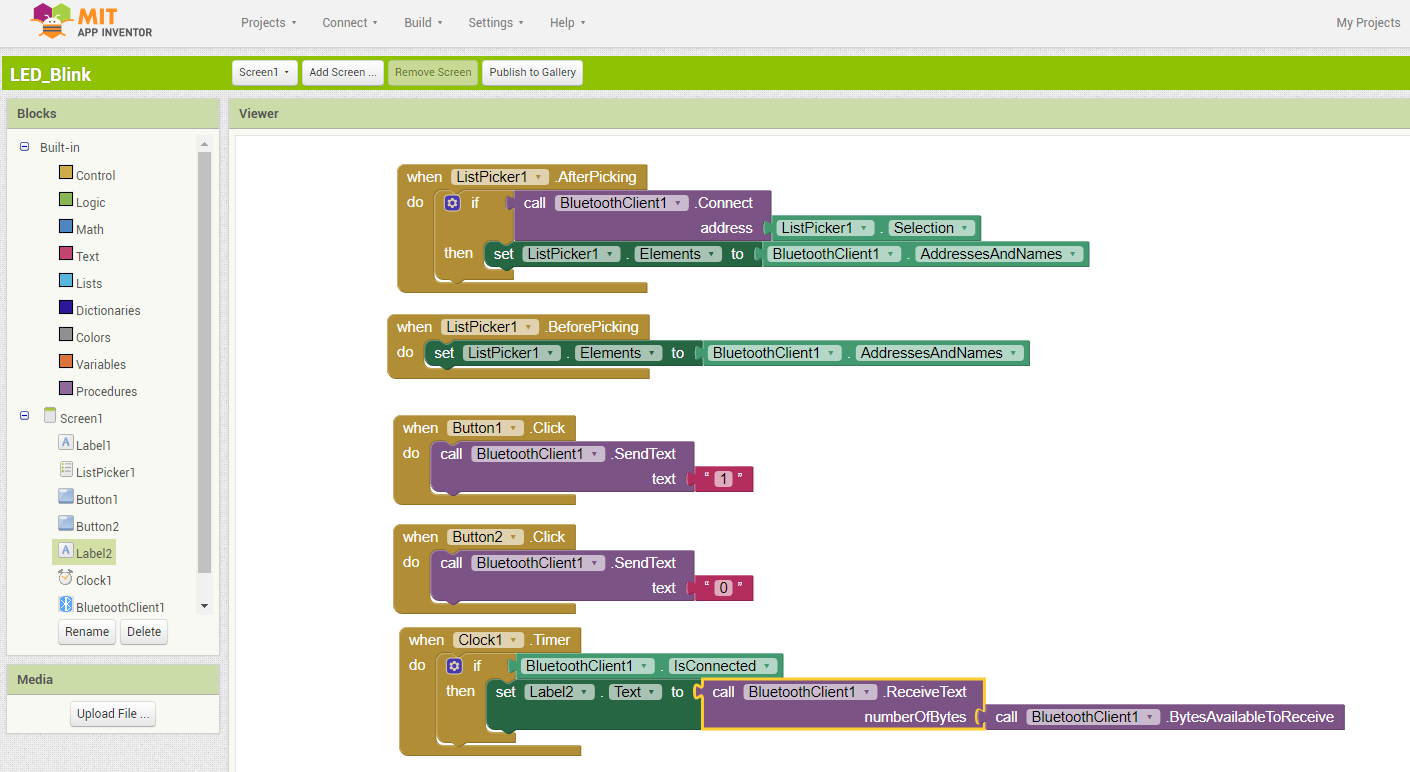
c)

* click on “Button1” and select “when Button1 .Click do” option.
* click on “BluetoothClient1” and select “call BluetoothClient1 .SenText text” option.
* Click on “Text” option for select a “text String” and enter “1” into it.

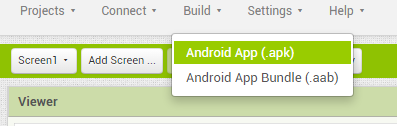
d)Repeat above step and change it as “Button2” and set Text as 0.

e)

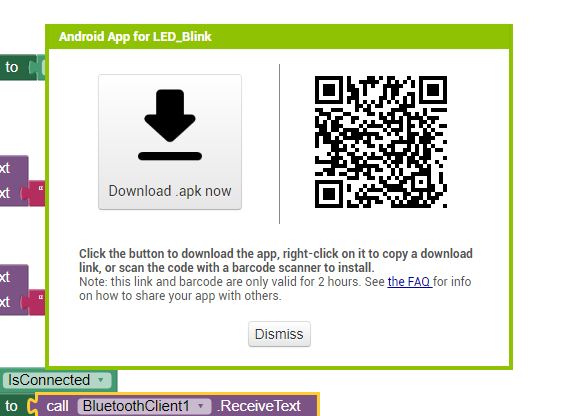
* Click on “Clock1” and select “when Clock1 . Timer do”
* Click on “Control” option for selecting “if then” option
* Click on “BluetoothClient1” for selecting “BluetoothClient1 .IsConnected”
* Click on “Label2” option for selecting “set Label2 . Text to”
* Click on “BluetoothClient1” for selecting “call BluetoothClient1 .ReceiveTextnumberOfBytes” and “call BluetoothClient1 .BytesAvailableToReceive” option.



Step 4: Click on “Build” and in drop down menu select “Andoid App (.apk)”



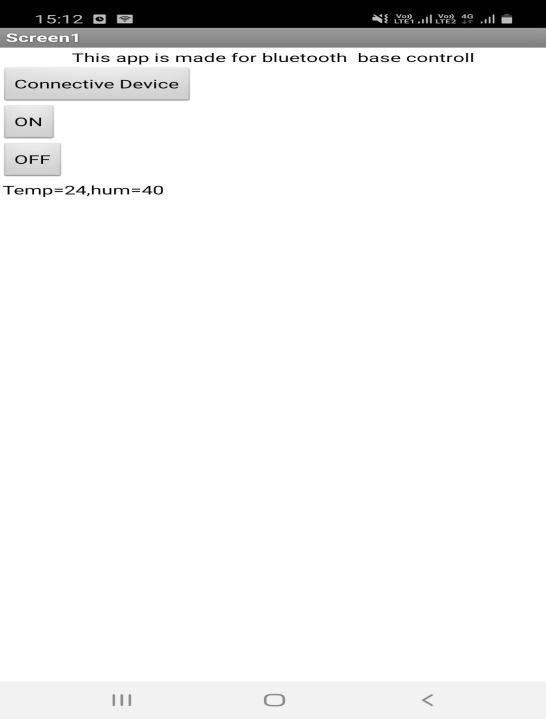
Step 5: Now QR Code will be generated.

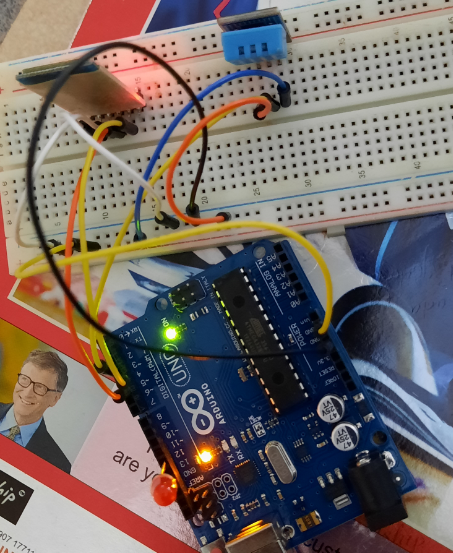


Step 6: Click on the link which is generated through scanning QR Code in our mobile and install “LED\_Blink”app.

Step 7: Now we can operate to turn LED ON/OFF when ‘1’/’0’ is received from smartphone using Bluetooth.

**Output:**





**Experiment 8**

**Aim: Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to thingspeak cloud**

**Procedure:**

Step 1: Open Arduino Interface and type program

**Program:**

#include <DHT.h>

#include <DHT\_U.h>

#include <ThingSpeak.h>

#include <ESP8266WiFi.h>

const char \*ssid="xxxxxxx"; // your network SSID (name)

const char \*pass="xxxxxxx"; // your network password

unsigned long chid=1648756;

const char \*key="WWB8GVQ941FIXU8W";

DHT Sensor(5,DHT11);

WiFiClientespclient;

void setup() {

// put your setup code here, to run once:

Sensor.begin();

ThingSpeak.begin(espclient);

Serial.begin(9600);

Serial.println("Connect with"+String(ssid));

WiFi.begin(ssid,pass);

while(WiFi.status()!=WL\_CONNECTED)

{Serial.println("....");

delay(150);

}

Serial.println("\nWifi Connected");

Serial.println(WiFi.localIP());

}

void loop() {

// put your main code here, to run repeatedly:

floattempr = Sensor.readTemperature();

float humid = Sensor.readHumidity();

ThingSpeak.setField(1,tempr);

ThingSpeak.setField(2,humid);

int x=ThingSpeak.writeFields(chid,key);

if(x == 200)

{Serial.println("Channel update successful.");

}

else{Serial.println("Problem updating channel HTTP error code " + String(x));

delay(15000);}

}

Step 2: Save the program

Step 3:

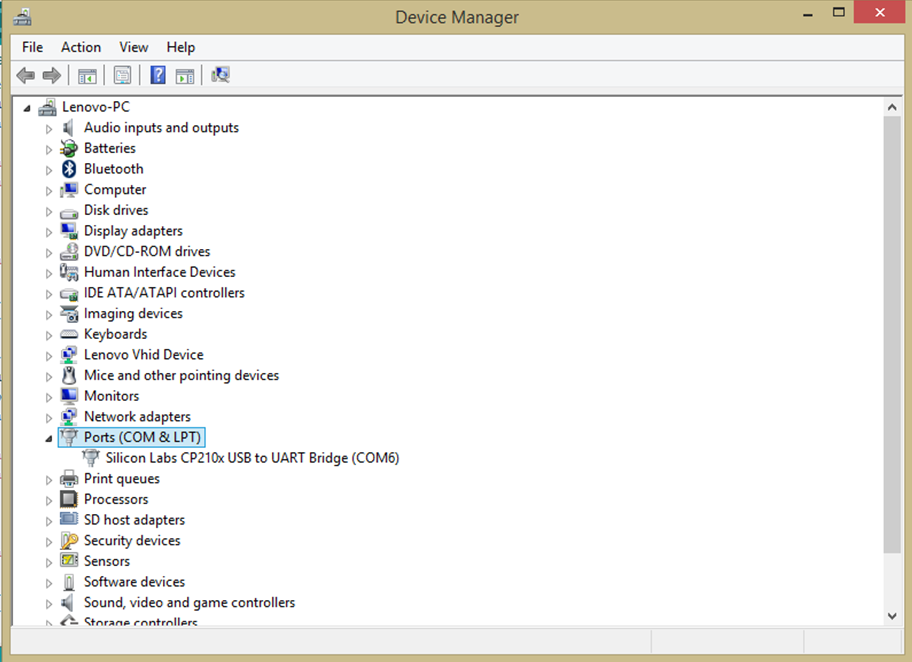
* Create “Thingspeak”account ( <https://thingspeak.com> )
* Login into “Thingspeak” account and click on “New Channel” button.
* Enter Experiment name in Channel Name field.
* Enter “Temperature” in Field1.
* Enter “Humidity” in Field2.
* Click on “Save Channel”

Step 4: Open program in Arduino Interface

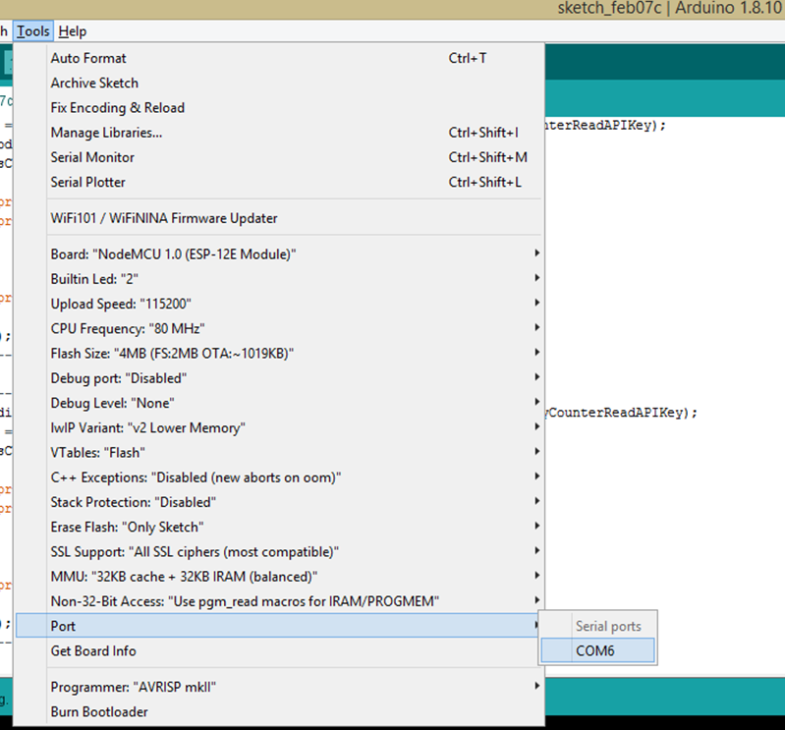
Step 5: click on compile  button.

Step 6: Using USB cable connect Arduino board to device.

Step 7: Open Device Manager in device and select ports.



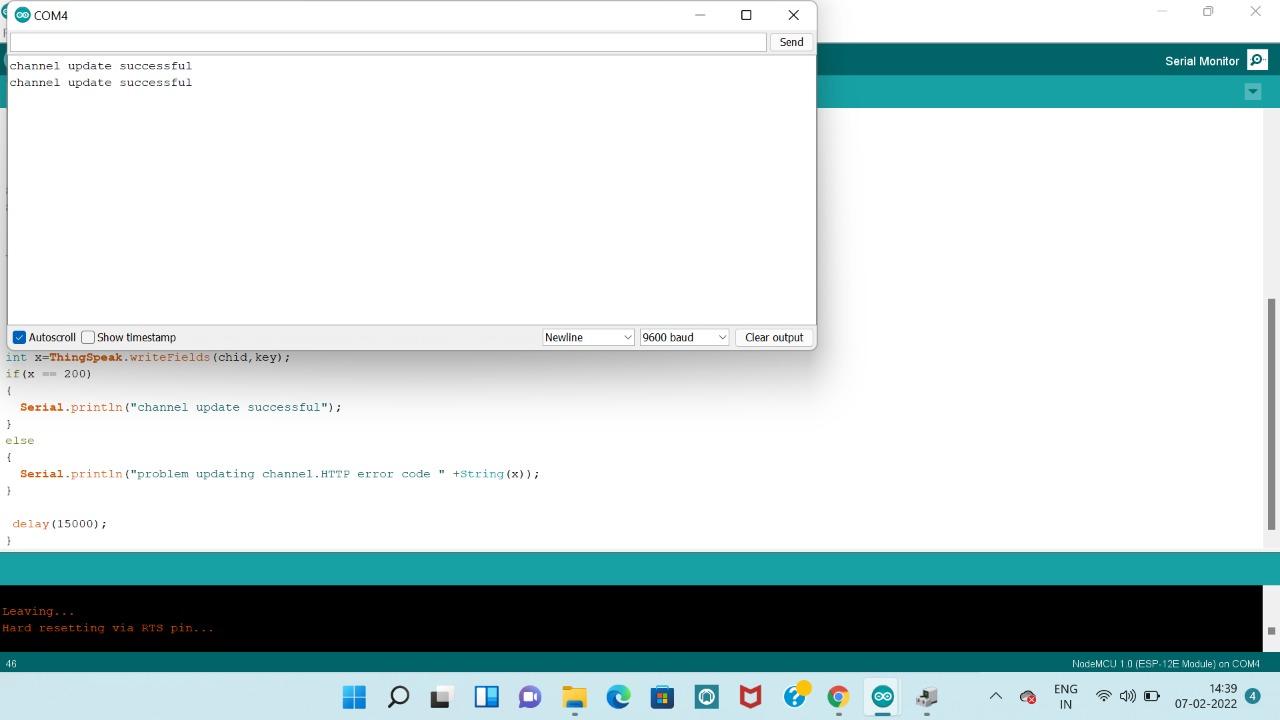
Step 8: Now click on Tools option in Arduino interface and select available port COM6/COM5/COM4/COM3



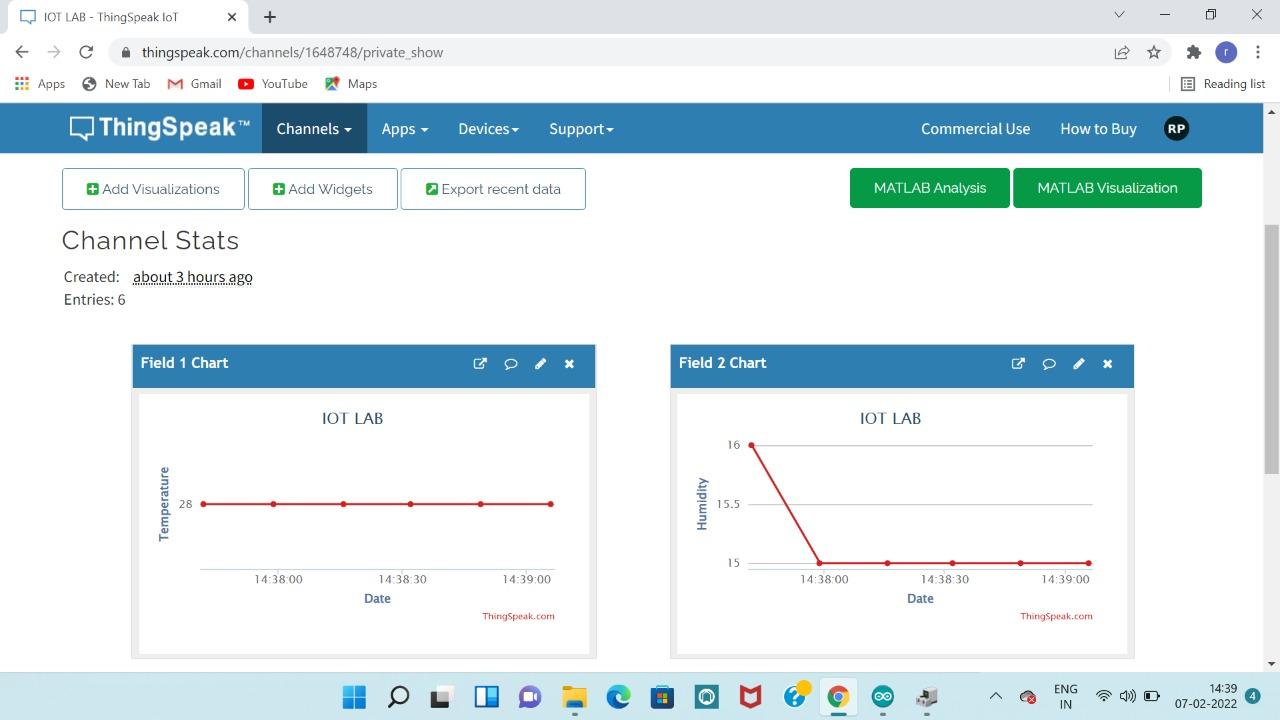
Step 9: Click upload  button

Step 10: Click on Serial Monitor 

**Output:**



Step 11: Click on Channels option and press Private View



**Experiment 9**

**Aim: Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thingspeak cloud.**

**Procedure:**

Step 1: Open Arduino Interface and type program

**Program:**

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

const char ssid[] = "xxxxxxx"; // your network SSID (name)

const char pass[] = "xxxxxxx"; // your network password

WiFiClient client;

//---------Channel Details---------//

unsigned long counterChannelNumber = 1648756; // Channel ID

const char \* myCounterReadAPIKey = "ODGH1JHLPGDUG39W"; // Read API Key

constint FieldNumber1 = 1; // The field you wish to read

constint FieldNumber2 = 2; // The field you wish to read

//-------------------------------//

void setup()

{

Serial.begin(9600);

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

}

void loop()

{

//----------------- Network -----------------//

if (WiFi.status() != WL\_CONNECTED)

{

Serial.print("Connecting to ");

Serial.print(ssid);

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

while (WiFi.status() != WL\_CONNECTED)

{

WiFi.begin(ssid, pass);

delay(5000);

}

Serial.println("Connected to Wi-Fi Succesfully.");

}

//--------- End of Network connection--------//

//---------------- Channel 1 ----------------//

long temp = ThingSpeak.readLongField(counterChannelNumber, FieldNumber1, myCounterReadAPIKey);

intstatusCode = ThingSpeak.getLastReadStatus();

if (statusCode == 200)

{

Serial.print("Temperature: ");

Serial.println(temp);

}

else

{

Serial.println("Unable to read channel / No internet connection");

}

delay(100);

//-------------- End of Channel 1 -------------//

//---------------- Channel 2 ----------------//

long humidity = ThingSpeak.readLongField(counterChannelNumber, FieldNumber2, myCounterReadAPIKey);

statusCode = ThingSpeak.getLastReadStatus();

if (statusCode == 200)

{

Serial.print("Humidity: ");

Serial.println(humidity);

}

else

{

Serial.println("Unable to read channel / No internet connection");

}

delay(100);

//-------------- End of Channel 2 -------------//

}

Step 2: Save the program

Step 3:

* Create “Thingspeak”account ( <https://thingspeak.com> )
* Login into “Thingspeak” account and click on “New Channel” button.
* Enter Experiment name in Channel Name field.
* Enter “Temperature” in Field1.
* Enter “Humidity” in Field2.
* Click on “Save Channel”

Step 4: Open program in Arduino Interface

Step 5: click on compile  button.

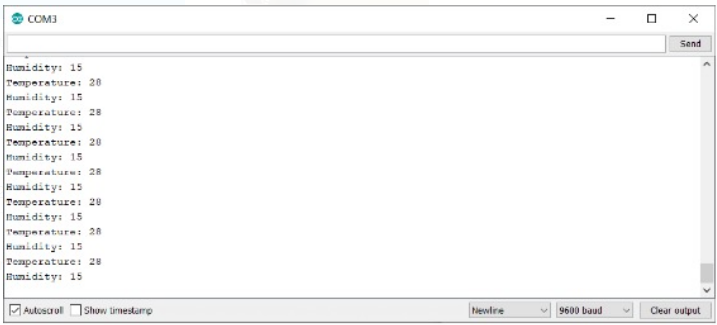
Step 6: Using USB cable connect Arduino board to device.

Step 7: Open Device Manager in device and select ports.

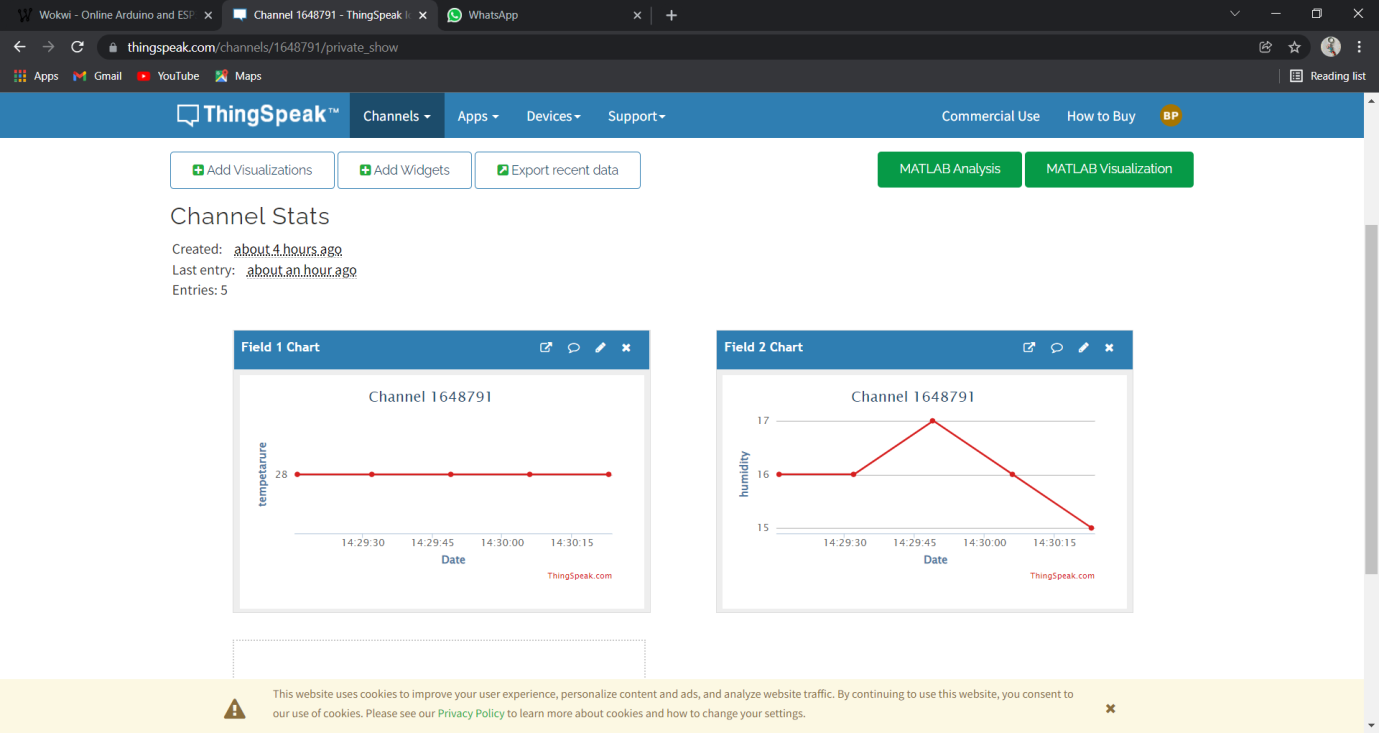
Step 8: Now click on Tools option in Arduino interface and select available port COM6/COM5/COM4/COM3

Step 9: Click upload  button

Step 10: Click on Serial Monitor 

**Output:**

Step 11: Click on Channels option and press Private View



**Experiment 10**

**Aim: To install MySql database on Raspberry Pi & perform basic SQL queries**

**Procedure: Install MySQL database on raspberry pi**

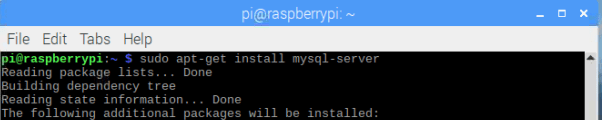
Follow below steps to know how to install MYSQL server on raspberry pi or to know how to install MariaDB server on raspberry pi.

1. Please open the raspberry pi terminal.
2. Execute the following command to update the existing packages.

**sudo apt-get update**

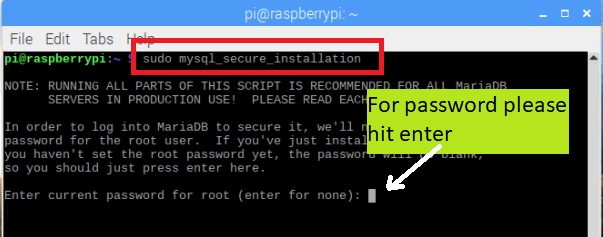
1. Now execute the following to install MySQL server which is shown below. While installing if it is asking do you want to continue then please enter y and hit enter

**sudo apt-get install mysql-server**

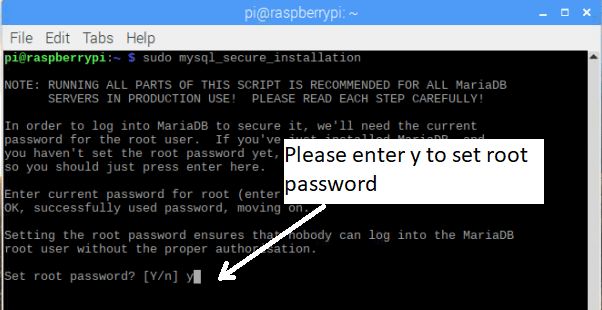


1. Now please execute the following command for secure installation which is shown below.

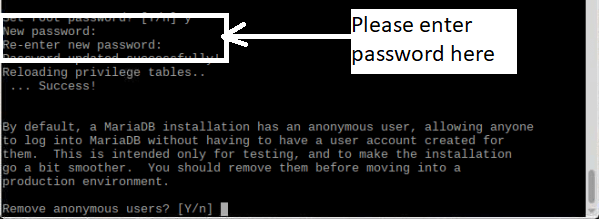
**sudomysql\_secure\_installation**



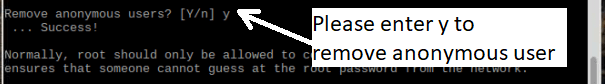
1. Please hit Enter for current root password.
2. Now please Enter y and hit Enter for setting a new password which is shown below



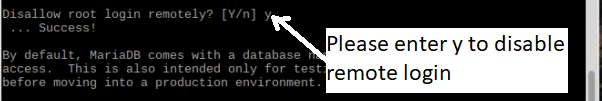
1. Now Please enter New password which is shown below.



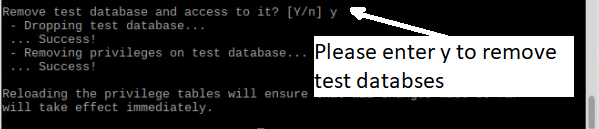
1. Now please enter y to remove anonymous user and hit Enter.



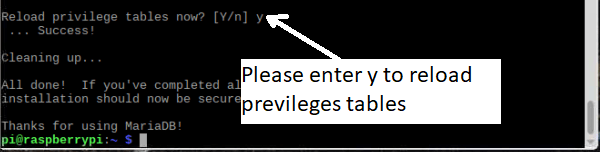
1. Now please enter y to disallow remote login which is shown below



1. Please enter y to remove test databases which is shown below

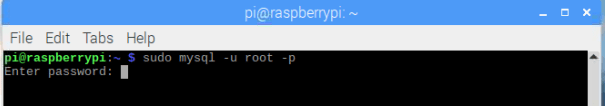


1. Please enter y to reload privileges tables which is shown below



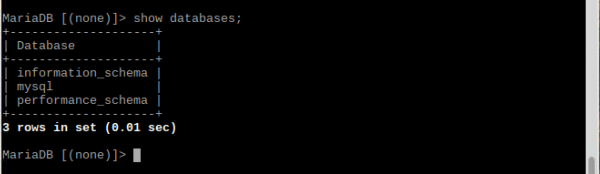
1. Now please execute the following command to login into the database and Enter the password which you have entered in step 7.

**sudomysql -u root –p**



1. Please execute the following command to see databases present in the mysql database.

**show databases;**



1. Execute the following to create Demo database in mysql server which is shown below.

**CREATE DATABASE Demo;**

Create database in mysql server

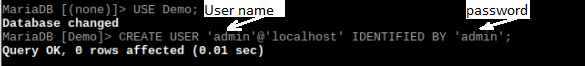
1. Now please execute the following to go in Demo database

**USE Demo;**

use database

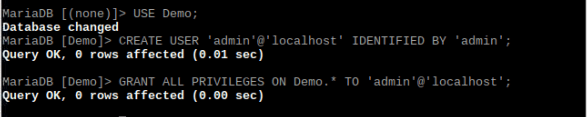
1. Please execute the following command to create database user

**CREATE USER ‘admin’@’localhost’ IDENTIFIED BY ‘admin’;**



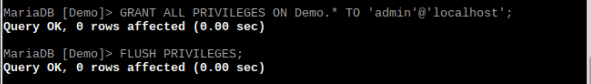
1. Execute the following command to grant all previleges

**GRANT ALL PRIVILEGES ON Demo.\* TO ‘admin’@’localhost’;**



1. Now execute the following command save all the changes

**FLUSH PRIVILEGES;**



1. Now please execute the following command to come out of database.

**quit**

Exit mysql server

1. Execute the following command to restart the MYSQL server

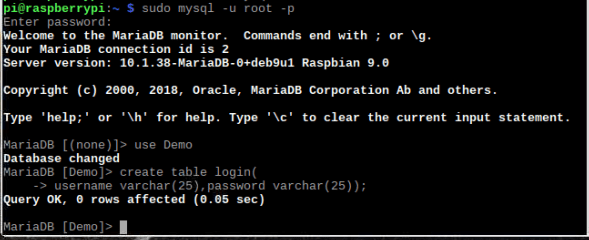
**sudo service mysql restart**

restart mysql server on raspberry pi

**Follow the following steps to insert and fetch from the MySQL database.**

1. Open the raspberry pi terminal.
2. Execute the following command to login to the database and enter the password which is shown below.

**sudomysql -u root –p**



1. Execute the following command to use Demo database which is shown above.

**USE Demo;**

**Output: MariaDB [(none)]> use Demo**

**Database changed**

1. Execute the following command to create login table which has two coloumsi.e is username and password which is shown above.

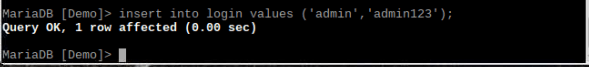
**create table login(username varchar(25), password varchar(25));**

**Output: MariaDB [Demo]> create table data(snoint(11), temp varchar(25), hum varchar(25));**

**Query OK, 0 rows affected (0.06 sec)**

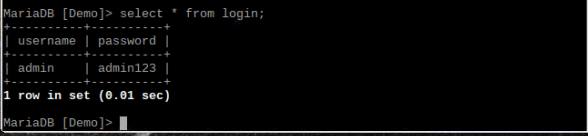
1. Execute the following command to insert data into login table which is shown below.

**insert into login values(‘admin’,’admin123′);**



1. To see the inserted values please execute the following command which is shown below

**select \* from login;**



1. To insert data values

**Output: MariaDB [Demo]>insert into login values(‘user’,’user123′);**

**Query OK, 1 row affected (0.01 sec)**

1. To delete data values

**Output: MariaDB [Demo]> delete from data where username=‘user’;**

**Query OK, 1 row affected (0.01 sec)**

1. To update data values

**Output: MariaDB [temphum]> update data set temp='26.4' where sno=1;**

**Query OK, 1 row affected (0.01 sec)**

**Rows matched: 1 Changed: 1 Warnings: 0**

1. To exit

**Output:MariaDB [temphum]> Ctrl-C -- exit!**

**Aborted**

**pi@raspberrypi:~ $**

**Experiment 11**

**Aim: Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker**

**Procedure:**

Step 1: Download & install Raspeberri Pi from https://www.raspberrypi.com/software/

Step 2: Open the raspberry pi terminal and write program

**Program:**

import sys

importAdafruit\_DHT

import time

frompaho.mqtt.client import \*

client=Client("Device1")

c=client.connect('broker.hivemq.com',1883)

if c==0:

print('client connected')

while True:

hum,temp=Adafruit\_DHT.read\_retry(11,4)

print('Temperature:',temp)

print('Humidity:', Hum)

message='temp='+str(temp)+",hum="+str(hum)

client.publish('iotdata',message)

time.sleep(2)

Step 3: Now save it

**Output:**

pi@raspberrypi:~ $ cd iot

pi@raspberrypi:~/iot $ python3 dhtpublish.py

client connected

Temperature: 28.0

Humidity: 95.0

Temperature: 29.0

Humidity: 94.0

Temperature: 28.0

Humidity: 94.0

Temperature: 29.0

Humidity: 95.0

Temperature: 28.0

Humidity: 94.0

Temperature: 29.0

Humidity: 94.0

Temperature: 28.0

Humidity: 93.0

**To see the result in Smartphone**

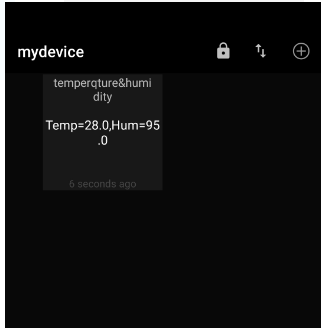
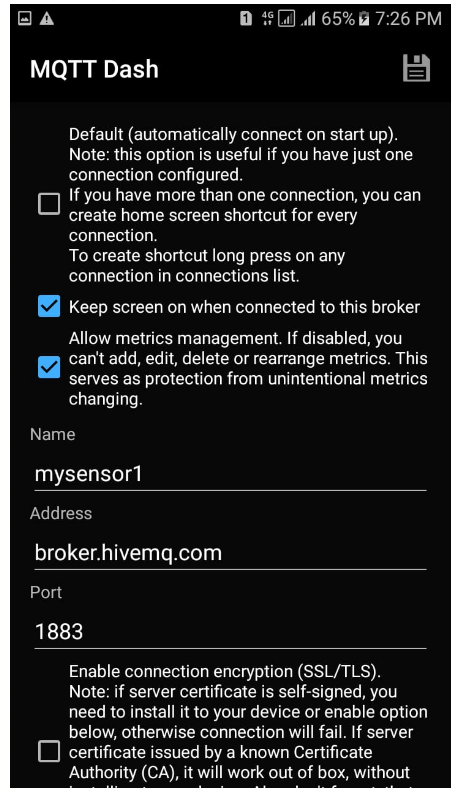
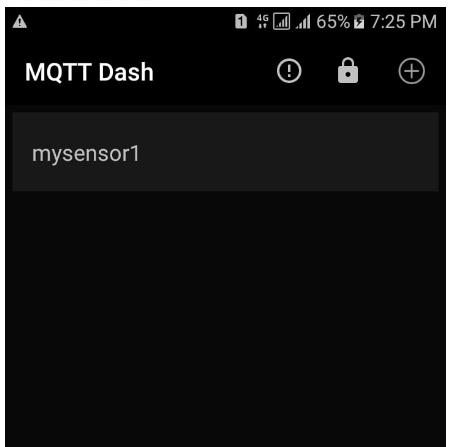
Step 1: Download MQTT Dash app from Play Store.

Step 2: Open MQTT Dash app in mobile and click on “+” icon.

Step 3: Enter Name like “mysensor1” and enter “broker.hivemq.com” in Address field.

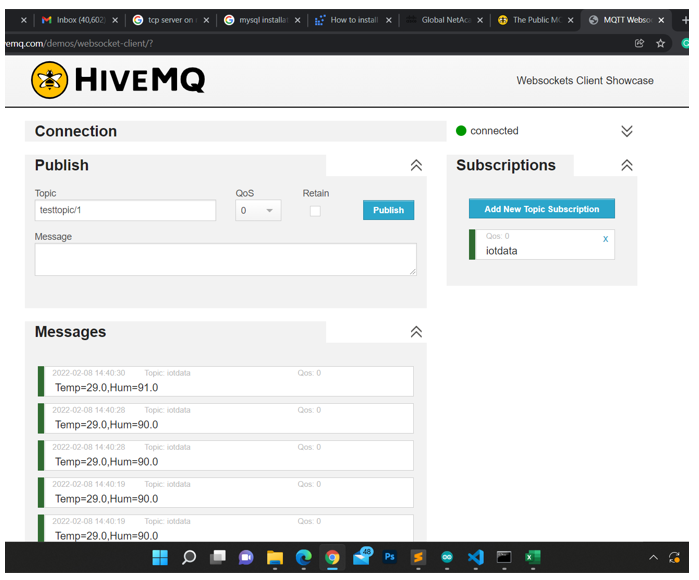
Step 4: Save it.

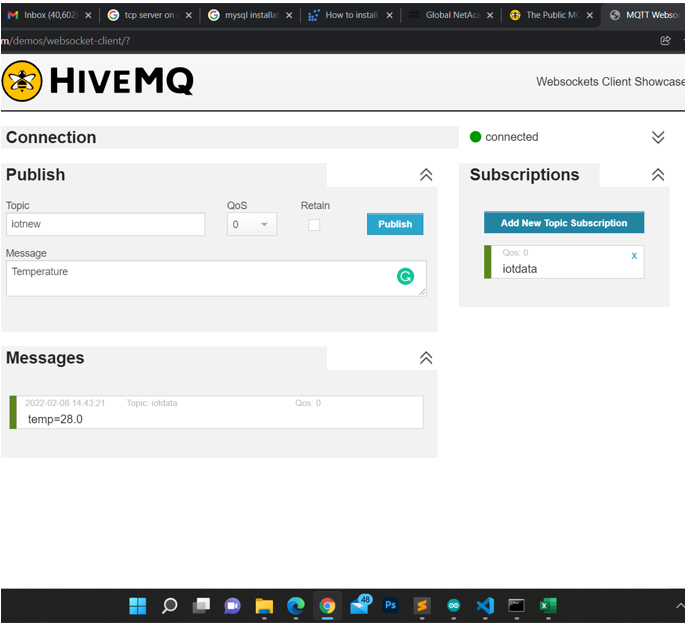
Step 5: Now click on “mysensor1”

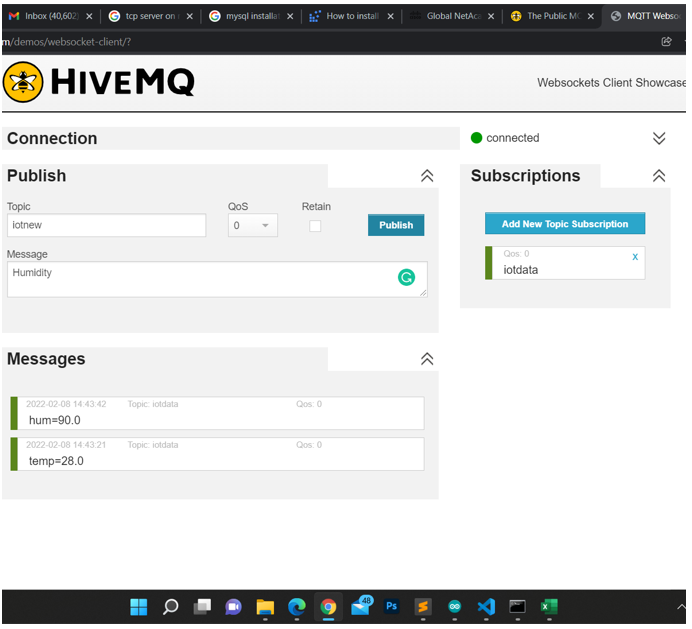


**To see the result in Website**

Open**“ hivemq.com/demos/websocket-client/ “** in google chrome.







**Experiment 12**

**Aim: Write a program on Arduino/Raspberry Pi to subscribe to MQTT broker for temperature data and print it.**

**Procedure:**

Step 1: Download & install Raspeberri Pi from https://www.raspberrypi.com/software/

Step 2: Open the raspberry pi terminal and write program

**Program:**

import sys

importAdafruit\_DHT

import time

frompaho.mqtt.client import \*

defon\_message(client,userdata,msg):

print('Channel:',msg.topic)

message=str(msg.payload.decode('utf-8'))

print('Message: ',message)

hum,temp=Adafruit\_DHT.read\_retry(11,4)

t='temp='+str(temp)

h='hum='+str(hum)

if message=='Temperature':

client.publish('iotdata',t)

if message=='Humidity':

client.publsih('iotdata',h)

client=Client("Device1")

c=client.connect('broker.hivemq.com',1883)

if c==0:

print('client connected')

client.subscribe('iot')

client.on\_message=on\_message

client.loop\_forever()

Step 3: Now save it

**Output:**

python3 dht.py

client connected

channel: iotnew

Message: Temperature

28.0

channel: iotnew

Message: Humidity

**To see the result in Smartphone**

Step 1: Download MQTT Dash app from Play Store.

Step 2: Open MQTT Dash app in mobile and click on “+” icon.

Step 3: Enter Name like “mysensor1” and enter “broker.hivemq.com” in Address field.

Step 4: Save it.

Step 5: Now click on “mysensor1”

