Experiment : 7

7. Using ESP 32 Wifi Module and Temperature Sensor create Weather Station . Read data from sensor and send it to a requesting client using socket communication.

Note: The client and server should be connected to same local area network.

Aim : To Design and develop Weather Station Using ESP 32 Wifi Module and Temperature Sensor and read and record the sensor temperature and humidity in a web server.

Apparatus Required:

1. Wokwi Web Portal
2. ESP 32 Wi-Fi Module
3. Temperature Sensor
4. www.hivemq.com. (online Web server)

ESP -32

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area.

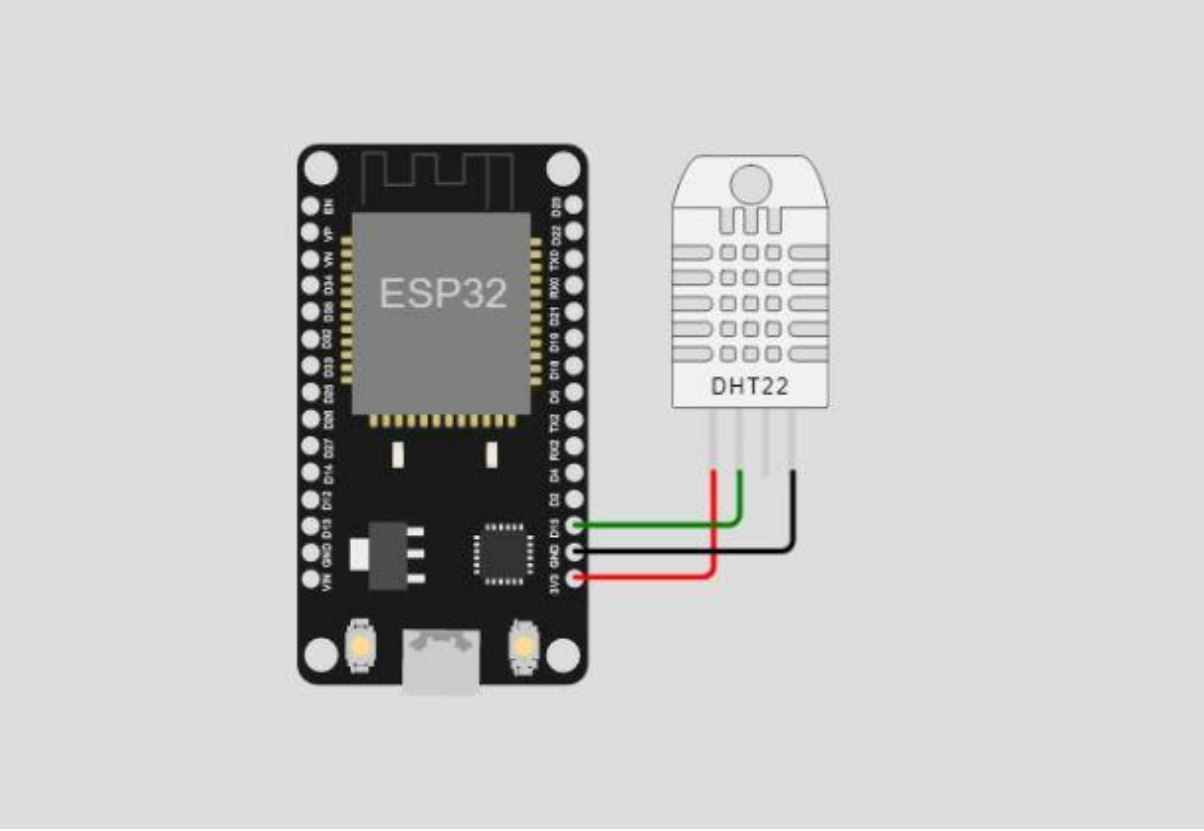
Temperature Sensor

A temperature sensor is an electronic device that measures the temperature of its environment and converts the input data into electronic data to record, monitor, or signal temperature changes. There are many different types of temperature sensors. Some temperature sensors require direct contact with the physical object that is being monitored (contact temperature sensors), while others indirectly measure the temperature of an object (non-contact temperature sensors). Non-contact temperature sensors are usually infrared (IR) sensors. They remotely detect the IR energy emitted by an object and send a signal to a calibrated electronic circuit that determines the object's temperature.

Among the contact temperature sensors are thermocouples and thermistors. A thermocouple is comprised of two conductors, each made of a different type of metal, that are joined at an end to form a junction. When the junction is exposed to heat, a voltage is generated that directly corresponds to the temperature input. This happens on account of the phenomena called the thermoelectric effect. Thermocouples are generally inexpensive, as their design and materials are simple. The other type of contact temperature sensor is called a thermistor. In thermistors, resistance decreases as temperature increases. There are two main types of thermistors: Negative Temperature Coefficient (NTC) and Positive Temperature Coefficient (PTC). Thermistors are more precise than thermocouples (capable of measuring within 0.05-1.5 degrees Celsius), and they are made of ceramics or polymers. Resistance Temperature Detectors (RTD) are essentially the metal counterpart of thermistors, and they are the most precise and expensive type of temperature sensors.

Temperature sensors are used in automobiles, medical devices, computers, cooking appliances, and other types of machinery.

Circuit Connections – (Wokwi)



MICRO python Code :

```
import network
import time
from machine import Pin
import dht
import ujson
from umqtt.simple import MQTTClient
# MQTT Server Parameters
MQTT_CLIENT_ID = "micropython-weather-demo"
MQTT_BROKER    = "broker.mqttdashboard.com"
MQTT_USER      = ""
MQTT_PASSWORD  = ""
MQTT_TOPIC     = "wokwi-weather"
sensor = dht.DHT22(Pin(15))
print("Connecting to WiFi", end="")
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect('Wokwi-GUEST', '')
while not sta_if.isconnected():
  print(".", end="")
  time.sleep(0.1)
print(" Connected!")
```

```
print("Connecting to MQTT server... ", end="")
client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER,
password=MQTT_PASSWORD)
client.connect()
print("Connected!")
prev_weather = ""
while True:
  print("Measuring weather conditions... ", end="")
  sensor.measure()
  message = ujson.dumps({
    "temp": sensor.temperature(),
    "humidity": sensor.humidity(),
  })
  if message != prev_weather:
    print("Updated!")
    print("Reporting to MQTT topic {}: {}".format(MQTT_TOPIC, message))
    client.publish(MQTT_TOPIC, message)
    prev_weather = message
  else:
    print("No change")
  time.sleep(1)
```

Execution Steps

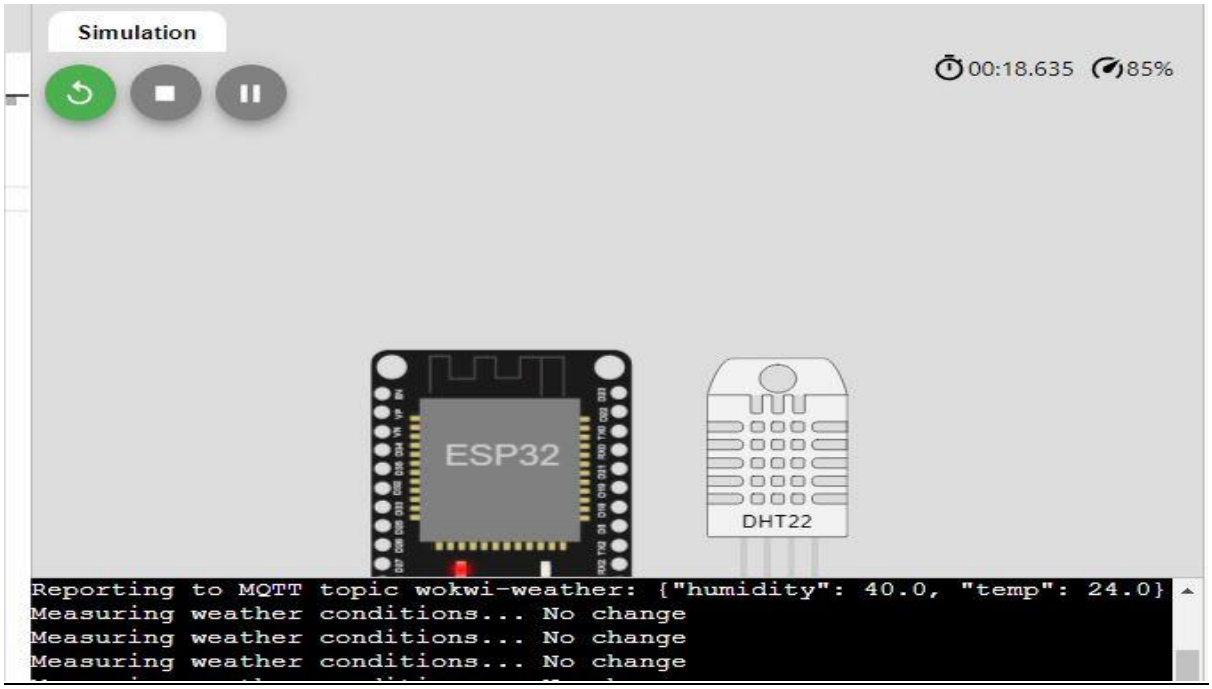
1. Connect the ESP 32 and Temperature Sensor.
2. Write the code

   Web server
3. 1. Go to http://www.hivemq.com/demos/websocket-client/
4. 2. Click "Connect"
5. 3. Under Subscriptions, click "Add New Topic Subscription"
6. 4. In the Topic field, type "wokwi-weather" then click "Subscribe"
7. Now click on the DHT22 sensor in the simulation,
8. change the temperature/humidity, and you should see
   the message appear on the MQTT Broker, in the "Messages" pane.

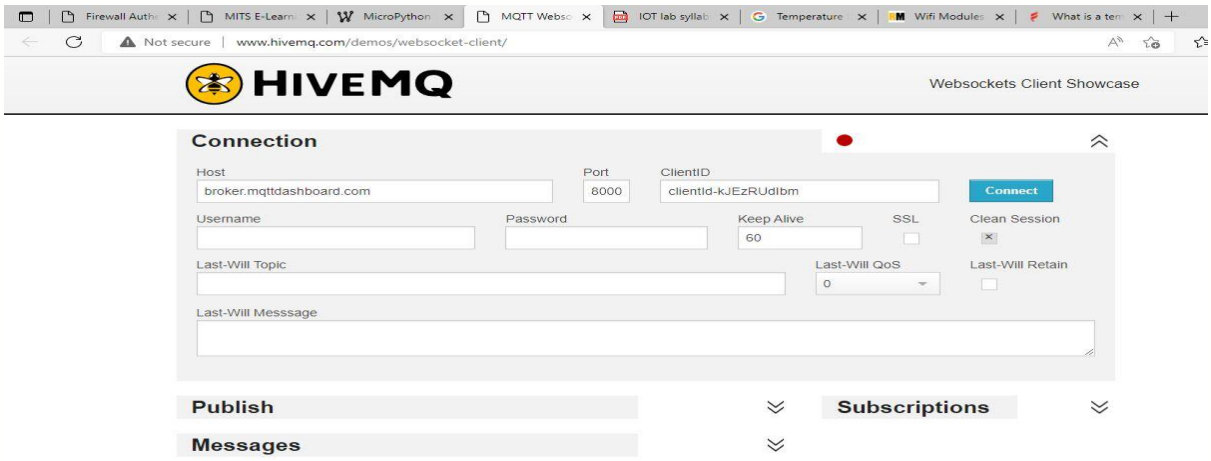
OutPut Screen.

WOkwi

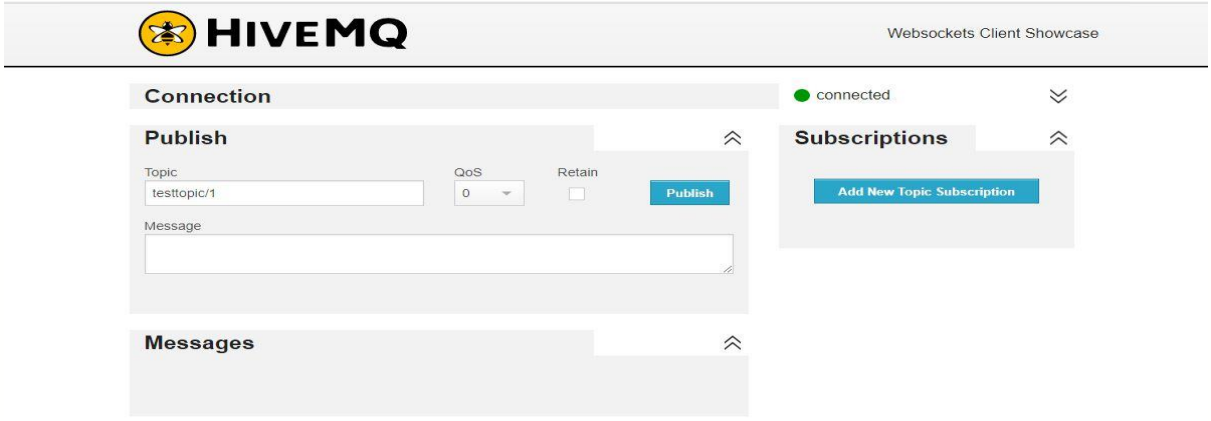
1. **Run the simulation and check the temperature sensor Working change temperature and humidity**



Go to http://www.hivemq.com/demos/websocket-client/

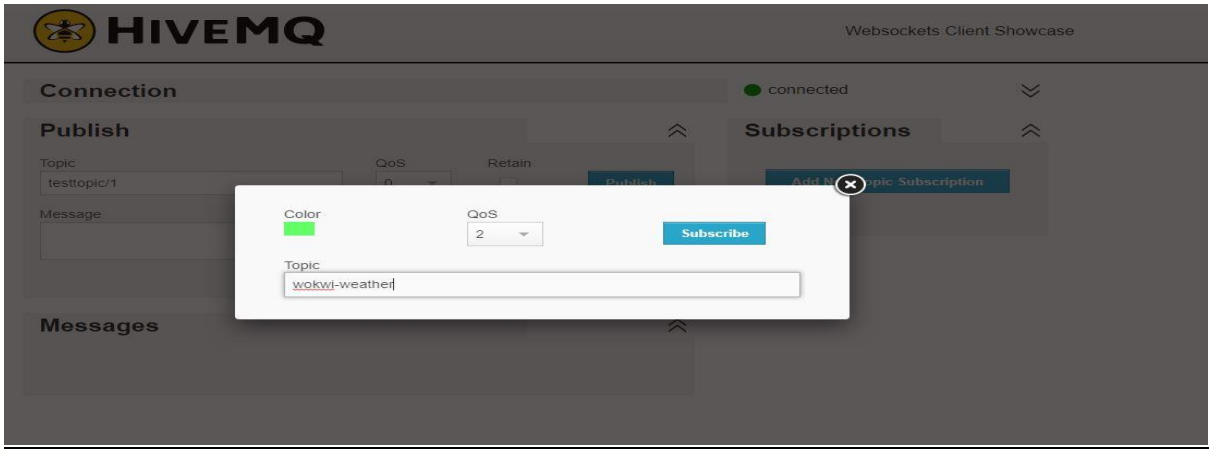


Click connect Button – ensure Connected Button turns GREEN

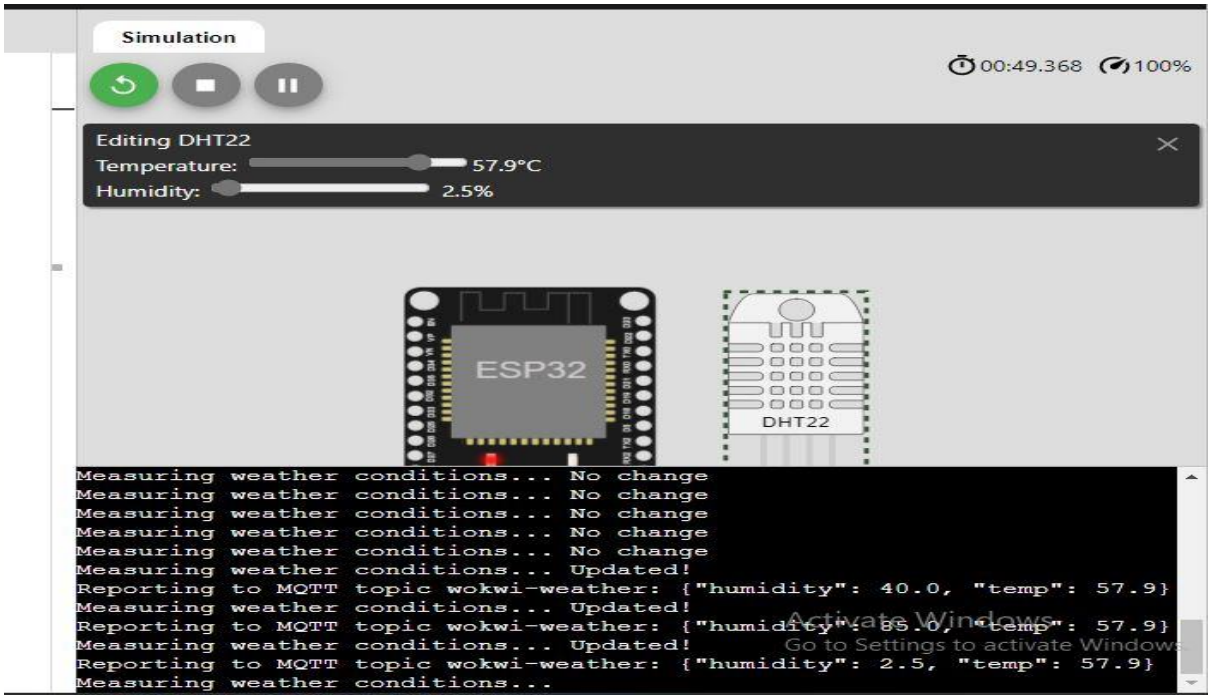


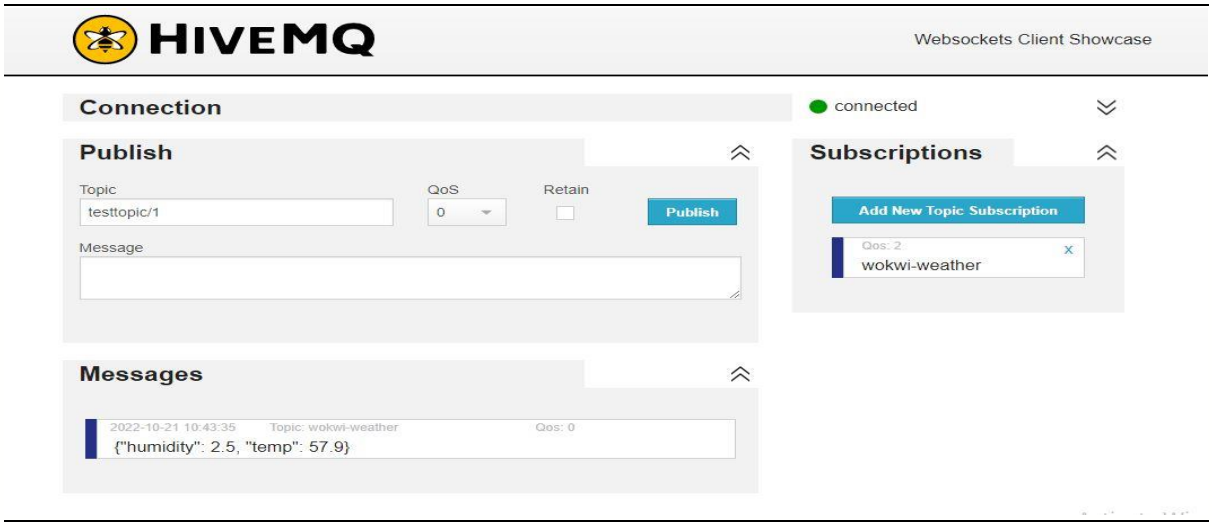
1. Click "Connect"
2. Under Subscriptions, click "Add New Topic Subscription"

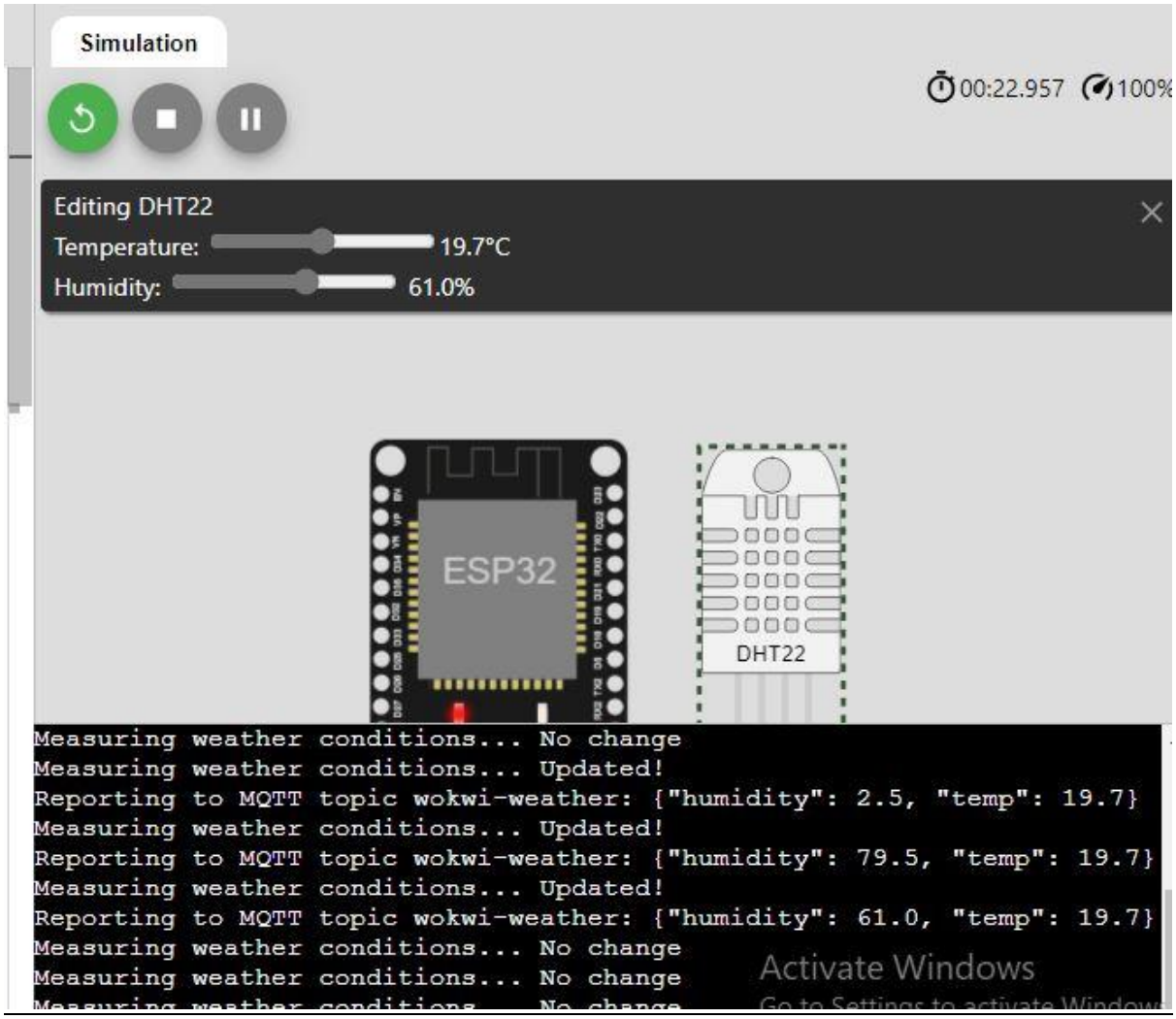
In the Topic field, type "**wokwi-weather**" then click "Subscribe

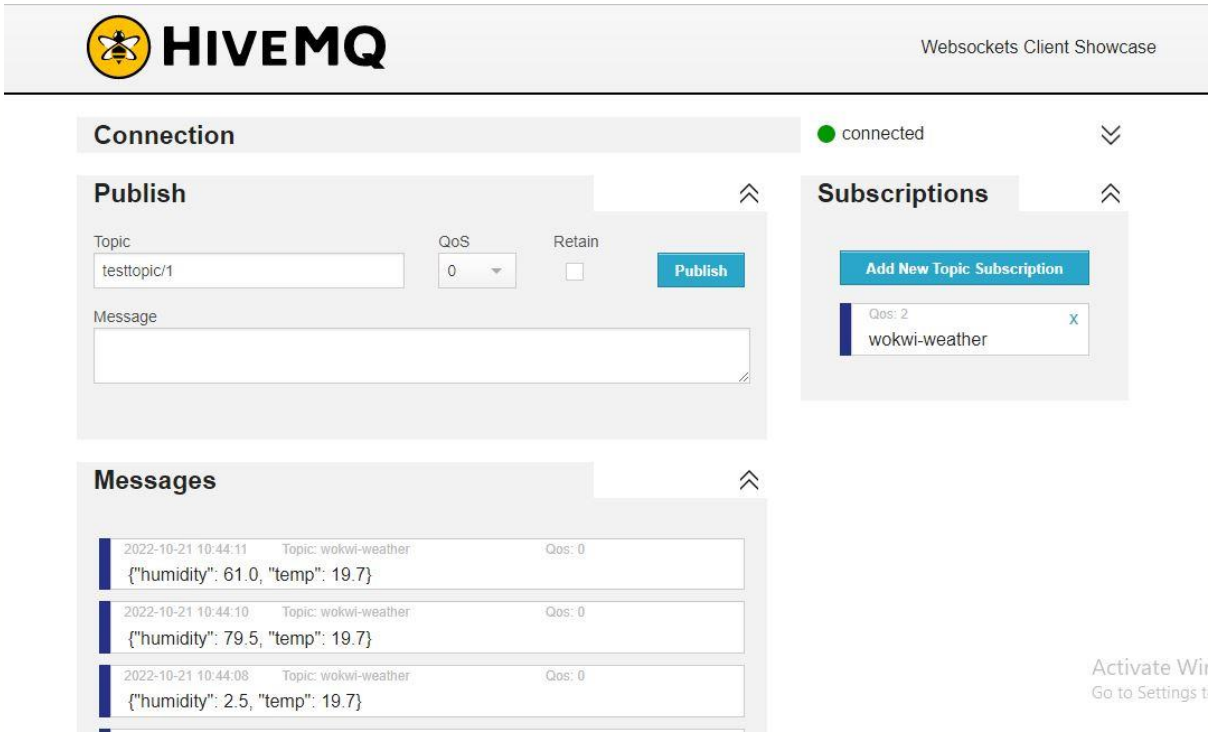


Run Simulator in Wokwi and you may find the changes of humidity & Temperature – by Moving slider and Same changes will be observed in the Web server also









Result : We successfully Designed and developed Weather Station Using ESP 32 Wifi Module and Temperature Sensor and read and record the sensor temperature and humidity in a web server.