**ENVIRONMENTAL MONITORING USING IOT**

**PHASE 3**

**SENSOR DESIGN SIMULATION AND ITS CODE**

**AIM:**

To design and simulate Temperature & Humidity sensors using ESP32 as an IoT Interface with python program**.**

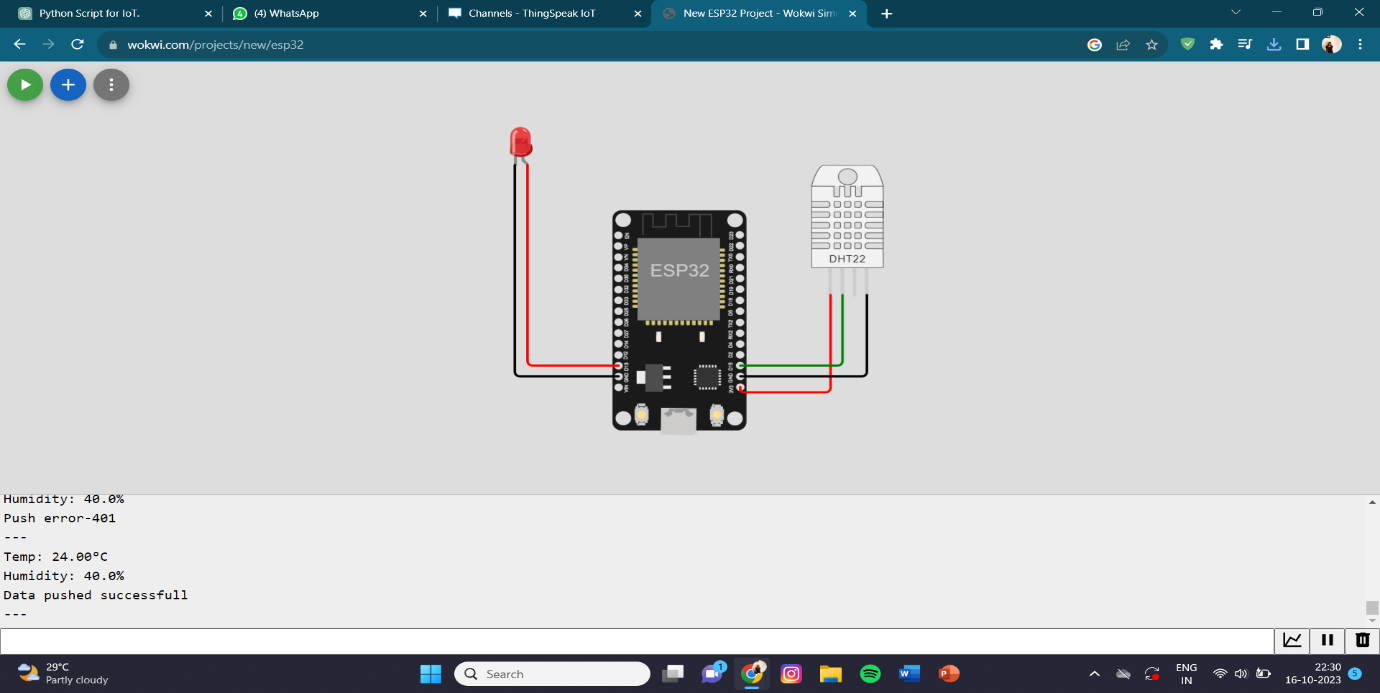
**COMPONENTS USED:**

SOFTWARE USED-WOKWI

IoT DEVICE -ESP32

SENSOR USED -DHT22

**CIRCUIT DESIGN:**

****

**PYTHON CODE:**

import machine

import dht

import network

import urequests

import time

DHT\_PIN = 15

LED\_PIN = 2  # Assuming you are using NodeMCU or similar with built-in LED

WIFI\_NAME = "Wokwi-GUEST"

WIFI\_PASSWORD = ""

myChannelNumber = 2306875

myApiKey = "LGD2VGLYYVTP3YV9"

server = "api.thingspeak.com"

dhtSensor = dht.DHT22(machine.Pin(DHT\_PIN))

led = machine.Pin(LED\_PIN, machine.Pin.OUT)

def connect\_wifi():

    wlan = network.WLAN(network.STA\_IF)

    if not wlan.isconnected():

        print("Connecting to WiFi...")

        wlan.active(True)

        wlan.connect(WIFI\_NAME, WIFI\_PASSWORD)

        while not wlan.isconnected():

            pass

    print("WiFi connected!")

    print("Local IP:", wlan.ifconfig()[0])

def push\_to\_thingspeak(data):

    url = "https://api.thingspeak.com/update?api\_key={0}&field1={1}&field2={2}".format(myApiKey, data["temperature"], data["humidity"])

    response = urequests.get(url)

    return response.status\_code

def read\_dht\_sensor():

    dhtSensor.measure()

    return {

        "temperature": dhtSensor.temperature(),

        "humidity": dhtSensor.humidity()

    }

connect\_wifi()

while True:

    sensor\_data = read\_dht\_sensor()

    led.value(1 if sensor\_data["temperature"] > 35 or sensor\_data["temperature"] < 12 or sensor\_data["humidity"] > 70 or sensor\_data["humidity"] < 40 else 0)

    response\_code = push\_to\_thingspeak(sensor\_data)

    print("Temp: {:.2f}°C".format(sensor\_data["temperature"]))

    print("Humidity: {:.1f}%".format(sensor\_data["humidity"]))

    if response\_code == 200:

        print("Data pushed successfully")

    else:

        print("Push error", response\_code)

    print("---")

    time.sleep(10)  # Sleep for 10 seconds

**OUTPUT WITH CODE:**

